

mmWave Radar Interface Control Document

Revision 1.7



July 10, 2019

Contents

| | Page |
|-----------------------------------------------------------------------|-------------|
| Contents | xi |
| List of Figures | xi |
| List of Tables | xvii |
| Revision History | xix |
| 1 Introduction | 1 |
| 1.1 Scope | 1 |
| 1.2 Intended Audience | 1 |
| 2 TI mmWave Radar Sensor Communications Overview | 3 |
| 2.1 Communication Link Description | 3 |
| 2.2 Communication Link configuration | 3 |
| 2.2.1 SPI | 3 |
| 2.2.2 Mailbox | 3 |
| 2.3 Radar Message Structure | 5 |
| 2.3.1 SYNC | 6 |
| 2.3.2 MSGHDR | 6 |
| 2.3.3 MSGDATA | 12 |
| 2.3.4 CRC | 13 |
| 3 Message Processing | 14 |
| 3.1 Communication protocol | 14 |
| 3.2 Communication Sequence | 15 |
| 3.2.1 Command/Response Sequence (Host) | 15 |
| 3.2.2 Flow Diagram (Host) – Command/Response | 16 |
| 3.2.3 Flow Diagram (Host) – Bootup/ Asynchronous Event | 17 |
| 3.2.4 SPI Message Sequence – Command/Response | 18 |
| 4 Radar Interface Messages Descriptions | 19 |
| 4.1 Summary of all messages and their associated sub-blocks | 19 |
| 4.2 AWR_ACK_MSG | 24 |

| | | |
|----------|-------------------------------------------------------------|-----------|
| 4.3 | AWR_NACK_MSG | 25 |
| 4.4 | AWR_ERROR_MSG | 25 |
| 4.5 | AWR_RF_STATIC_CONF_SET_MSG | 26 |
| 4.6 | AWR_RF_STATIC_CONF_GET_MSG | 27 |
| 4.7 | AWR_RF_INIT_MSG | 27 |
| 4.8 | AWR_RF_DYNAMIC_CONF_SET_MSG | 28 |
| 4.9 | AWR_RF_DYNAMIC_CONF_GET_MSG | 29 |
| 4.10 | AWR_RF_FRAME_TRIG_MSG | 30 |
| 4.11 | AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG | 30 |
| 4.12 | AWR_RF_MONITORING_CONF_SET_MSG | 31 |
| 4.13 | AWR_RF_STATUS_GET_MSG | 33 |
| 4.14 | AWR_RF_MONITORING_REPORT_GET_MSG | 33 |
| 4.15 | AWR_RF_MISC_CONF_SET_MSG | 34 |
| 4.16 | AWR_RF_MISC_CONF_GET_MSG | 34 |
| 4.17 | AWR_RF_ASYNC_EVENT_MSG1 | 35 |
| 4.18 | AWR_RF_ASYNC_EVENT_MSG2 | 36 |
| 4.19 | AWR_DEV_RFPOWERUP_MSG | 37 |
| 4.20 | AWR_DEV_CONF_SET_MSG | 38 |
| 4.21 | AWR_DEV_CONF_GET_MSG | 39 |
| 4.22 | AWR_DEV_FILE_DOWNLOAD_MSG | 40 |
| 4.23 | AWR_DEV_FRAME_CONFIG_APPLY_MSG | 40 |
| 4.24 | AWR_DEV_STATUS_GET_MSG | 41 |
| 4.25 | AWR_DEV_ASYNC_EVENT_MSG | 42 |
| 5 | Radar Functional APIs | 43 |
| 5.1 | Sub block related to AWR_ERROR_MSG | 43 |
| 5.1.1 | Sub block 0x0000 – AWR_RESP_ERROR_SB | 43 |
| 5.2 | Sub blocks related to AWR_RF_STATIC_CONF_SET_MSG | 44 |
| 5.2.1 | Sub block 0x0080 – AWR_CHAN_CONF_SET_SB | 44 |
| 5.2.2 | Sub block 0x0082 – AWR_ADCOUT_CONF_SET_SB | 47 |
| 5.2.3 | Sub block 0x0083 – AWR_LOWPOWERMODE_CONF_SET_SB | 48 |
| 5.2.4 | Sub block 0x0084 – AWR_DYNAMICPOWERSAVE_CONF_SET_SB | 48 |
| 5.2.5 | Sub block 0x0085 – AWR_HIGHSPEEDINTFCLK_CONF_SET_SB | 49 |
| 5.2.6 | Sub block 0x0086 – AWR_RF_DEVICE_CFG_SB | 50 |
| 5.2.7 | Sub block 0x0087 – AWR_RF_RADAR_MISC_CTL_SB | 52 |
| 5.2.8 | Sub block 0x0088 – AWR_CAL_MON_FREQUENCY_LIMITS_SB | 53 |
| 5.2.9 | Sub block 0x0089 – AWR_RF_INIT_CALIBRATION_CONF_SB | 54 |
| 5.2.10 | Sub block 0x008A – AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB | 55 |
| 5.2.11 | Sub block 0x008B – AWR_CAL_DATA_RESTORE_SB | 58 |
| 5.2.12 | Sub block 0x008C – AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB | 59 |
| 5.3 | Sub blocks related to AWR_RF_STATIC_CONF_GET_MSG | 59 |
| 5.3.1 | Sub block 0x00A0 – 0x00AA – RESERVED | 59 |
| 5.3.2 | Sub block 0x00AB – AWR_CAL_DATA_SAVE_SB | 59 |

| | | |
|--------|---------------------------------------------------------------------------|-----|
| 5.3.3 | Sub block 0x00AC – AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB . . . | 62 |
| 5.4 | Sub blocks related to AWR_RF_INIT_MSG | 64 |
| 5.4.1 | Sub block 0x00C0 – AWR_RF_INIT_SB | 64 |
| 5.5 | Sub blocks related to AWR_RF_DYNAMIC_CONF_SET_MSG | 64 |
| 5.5.1 | Sub block 0x0100 – AWR_PROFILE_CONF_SET_SB | 64 |
| 5.5.2 | Sub block 0x0101 – AWR_CHIRP_CONF_SET_SB | 73 |
| 5.5.3 | Sub block 0x0102 – AWR_FRAME_CONF_SET_SB | 74 |
| 5.5.4 | Sub block 0x0103 – AWR_CONT_STREAMING_MODE_CONF_SET_SB | 77 |
| 5.5.5 | Sub block 0x0104 – AWR_CONT_STREAMING_MODE_EN_SB | 80 |
| 5.5.6 | Sub block 0x0105 – AWR_ADVANCED_FRAME_CONF_SB | 80 |
| 5.5.7 | Sub block 0x0106 – AWR_PERCHIRPPHASESHIFT_CONF_SB | 90 |
| 5.5.8 | Sub block 0x0107 – AWR_PROG_FILT_COEFF_RAM_SET_SB | 91 |
| 5.5.9 | Sub block 0x0108 – AWR_PROG_FILT_CONF_SET_SB | 92 |
| 5.5.10 | Sub block 0x0109 – AWR_CALIB_MON_TIME_UNIT_CONF_SB | 93 |
| 5.5.11 | Sub block 0x010A – AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB | 94 |
| 5.5.12 | Sub block 0x010B – AWR_INTER_RX_GAIN_PHASE_CONTROL_SB | 98 |
| 5.5.13 | Sub block 0x010C – AWR_RX_GAIN_TEMPLUT_SET_SB | 99 |
| 5.5.14 | Sub block 0x010D – AWR_TX_GAIN_TEMPLUT_SET_SB | 101 |
| 5.5.15 | Sub block 0x010E – AWR_LOOPBACK_BURST_CONF_SET_SB | 104 |
| 5.5.16 | Sub block 0x010F – AWR_DYN_CHIRP_CONF_SET_SB | 109 |
| 5.5.17 | Sub block 0x0110 – AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SET_SB | 112 |
| 5.5.18 | Sub block 0x0111 – AWR_DYN_CHIRP_ENABLE_SB | 114 |
| 5.5.19 | Sub block 0x0112 – AWR_INTERCHIRP_BLOCKCONTROLS_SB | 115 |
| 5.5.20 | Sub block 0x0113 – AWR_SUBFRAME_START_CONF_SB | 117 |
| 5.6 | Sub blocks related to AWR_RF_DYNAMIC_CONF_GET_SB | 119 |
| 5.6.1 | Sub block 0x0120 – AWR_PROFILE_CONF_GET_SB | 119 |
| 5.6.2 | Sub block 0x0121 – AWR_CHIRP_CONF_GET_SB | 119 |
| 5.6.3 | Sub block 0x0122 – AWR_FRAME_CONF_GET_SB | 119 |
| 5.6.4 | Sub block 0x0123 – RESERVED | 120 |
| 5.6.5 | Sub block 0x0124 – RESERVED | 120 |
| 5.6.6 | Sub block 0x0125 – AWR_ADV_FRAME_CONF_GET_SB | 120 |
| 5.6.7 | Sub block 0x0126 – RESERVED | 120 |
| 5.6.8 | Sub block 0x0127 – RESERVED | 120 |
| 5.6.9 | Sub block 0x0128 – RESERVED | 120 |
| 5.6.10 | Sub block 0x0129 – RESERVED | 120 |
| 5.6.11 | Sub block 0x012A – RESERVED | 120 |
| 5.6.12 | Sub block 0x012B – RESERVED | 120 |
| 5.6.13 | Sub block 0x012C – AWR_RX_GAIN_TEMPLUT_GET_SB | 120 |
| 5.6.14 | Sub block 0x012D – AWR_TX_GAIN_TEMPLUT_GET_SB | 121 |
| 5.7 | Sub blocks related to AWR_FRAME_TRIG_MSG | 121 |
| 5.7.1 | Sub block 0x0140 – AWR_FRAMESTARTSTOP_CONF_SB | 121 |
| 5.8 | Sub blocks related to AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG | 122 |

| | | |
|---------|--------------------------------------------------------|-----|
| 5.8.1 | Sub block 0x0180 – AWR_BPM_COMMON_CONF_SET_SB | 122 |
| 5.8.2 | Sub block 0x0181 – AWR_BPM_CHIRP_CONF_SET_SB | 123 |
| 5.9 | Sub blocks related to AWR_RF_STATUS_GET_MSG | 124 |
| 5.9.1 | Sub block 0x0220 – AWR_RF_VERSION_GET_SB | 124 |
| 5.9.2 | Sub block 0x0221 – AWR_RF_CPUFAULT_STATUS_GET_SB | 126 |
| 5.9.3 | Sub block 0x0222 – AWR_RF_ESMFAULT_STATUS_GET_SB | 128 |
| 5.9.4 | Sub block 0x0223 – AWR_RF_DIEID_GET_SB | 130 |
| 5.9.5 | Sub block 0x0224 – AWR_RF_BOOTUPBIST_STATUS_GET_SB | 131 |
| 5.10 | Sub blocks related to AWR_RF_MONITORING_REPORT_GET_MSG | 134 |
| 5.10.1 | Sub block 0x0260 – AWR_RF_DFE_STATISTICS_REPORT_GET_SB | 134 |
| 5.11 | Sub blocks related to AWR_RF_MISC_CONF_SET_MSG | 142 |
| 5.11.1 | Sub block 0x02C0 – RESERVED | 142 |
| 5.11.2 | Sub block 0x02C1 – RESERVED | 142 |
| 5.11.3 | Sub block 0x02C2 – AWR_RF_TEST_SOURCE_CONFIG_SET_SB | 142 |
| 5.11.4 | Sub block 0x02C3 – AWR_RF_TEST_SOURCE_ENABLE_SET_SB | 145 |
| 5.11.5 | Sub block 0x02C4 – 0x02CB RESERVED | 145 |
| 5.11.6 | Sub block 0x02CC – AWR_RF_LDO_BYPASS_SB | 145 |
| 5.11.7 | Sub block 0x02CD – AWR_RF_PALOOPBACK_CFG_SB | 147 |
| 5.11.8 | Sub block 0x02CE – AWR_RF_PSLOOPBACK_CFG_SB | 147 |
| 5.11.9 | Sub block 0x02CF – AWR_RF_IFLOOPBACK_CFG_SB | 149 |
| 5.11.10 | Sub block 0x02D0 – AWR_RF_GPADC_CFG_SET_SB | 150 |
| 5.11.11 | Sub block 0x02D1 – RESERVED | 153 |
| 5.11.12 | Sub block 0x02D2 – RESERVED | 153 |
| 5.11.13 | Sub block 0x02D3 – RESERVED | 153 |
| 5.12 | Sub blocks related to AWR_RF_MISC_CONF_GET_MSG | 153 |
| 5.12.1 | Sub block 0x02E0 to 0x2E9 – RESERVED | 153 |
| 5.12.2 | Sub block 0x02EA – AWR_RF_TEMPERATURE_GET_SB | 153 |
| 5.13 | Sub blocks related to AWR_RF_ASYNC_EVENT_MSG1 | 155 |
| 5.13.1 | Sub block 0x1000 – RESERVED | 155 |
| 5.13.2 | Sub block 0x1001 – RESERVED | 155 |
| 5.13.3 | Sub block 0x1002 – AWR_AE_RF_CPUFAULT_SB | 155 |
| 5.13.4 | Sub block 0x1003 – AWR_AE_RF_ESMFAULT_SB | 157 |
| 5.13.5 | Sub block 0x1004 – AWR_AE_RF_INITCALIBSTATUS_SB | 159 |
| 5.13.6 | Sub block 0x1005 – RESERVED | 161 |
| 5.13.7 | Sub block 0x1006 – RESERVED | 161 |
| 5.13.8 | Sub block 0x1007 – RESERVED | 161 |
| 5.13.9 | Sub block 0x1008 – RESERVED | 161 |
| 5.13.10 | Sub block 0x1009 – RESERVED | 161 |
| 5.13.11 | Sub block 0x100A – RESERVED | 161 |
| 5.13.12 | Sub block 0x100B – AWR_AE_RF_FRAME_TRIGGER_RDY_SB | 161 |
| 5.13.13 | Sub block 0x100C – AWR_AE_RF_GPADC_RESULT_DATA_SB | 161 |
| 5.13.14 | Sub block 0x100E – RESERVED | 163 |
| 5.13.15 | Sub block 0x100D – RESERVED | 163 |
| 5.13.16 | Sub block 0x100E – RESERVED | 163 |

| | | |
|---------|-----------------------------------------------------------------------------------|-----|
| 5.13.17 | Sub block 0x100F – AWR_FRAME_END_AE_SB | 163 |
| 5.13.18 | Sub block 0x1010 – AWR_ANALOGFAULT_AE_SB | 163 |
| 5.13.19 | Sub block 0x1011 – AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB | 164 |
| 5.13.20 | Sub block 0x1012 – AWR_RUN_TIME_CALIB_SUMMARY_REPORT_AE_SB | 165 |
| 5.13.21 | Sub block 0x1013 – AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB | 168 |
| 5.13.22 | Sub block 0x1014 – RESERVED | 170 |
| 5.13.23 | Sub block 0x1015 – AWR_MONITOR_REPORT_HEADER_AE_SB | 170 |
| 5.13.24 | Sub block 0x1016 – AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB | 170 |
| 5.13.25 | Sub block 0x1017 – AWR_MONITOR_TEMPERATURE_REPORT_AE_SB | 171 |
| 5.13.26 | Sub block 0x1018 – AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB | 172 |
| 5.13.27 | Sub block 0x1019 – AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB | 176 |
| 5.13.28 | Sub block 0x101A – AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB | 178 |
| 5.13.29 | Sub block 0x101B – AWR_MONITOR_TX0_POWER_REPORT_AE_SB | 180 |
| 5.13.30 | Sub block 0x101C – AWR_MONITOR_TX1_POWER_REPORT_AE_SB | 181 |
| 5.13.31 | Sub block 0x101D – AWR_MONITOR_TX2_POWER_REPORT_AE_SB | 182 |
| 5.13.32 | Sub block 0x101E – AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB | 183 |
| 5.13.33 | Sub block 0x101F – AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB | 184 |
| 5.14 | Sub blocks related to AWR_RF_ASYNC_EVENT_MSG2 | 185 |
| 5.14.1 | Sub block 0x1020 – AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB | 185 |
| 5.14.2 | Sub block 0x1021 – AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_REPORT_AE_SB | 186 |
| 5.14.3 | Sub block 0x1022 – AWR_MONITOR_TX0_BPM_REPORT_AE_SB | 188 |
| 5.14.4 | Sub block 0x1023 – AWR_MONITOR_TX1_BPM_REPORT_AE_SB | 189 |
| 5.14.5 | Sub block 0x1024 – AWR_MONITOR_TX2_BPM_REPORT_AE_SB | 191 |
| 5.14.6 | Sub block 0x1025 – AWR_MONITOR_SYNTHESIZER_FREQUENCY_REPORT_AE_SB | 192 |
| 5.14.7 | Sub block 0x1026 – AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 193 |
| 5.14.8 | Sub block 0x1027 – AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 194 |
| 5.14.9 | Sub block 0x1028 – AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 195 |
| 5.14.10 | Sub block 0x1029 – AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 196 |
| 5.14.11 | Sub block 0x102A – AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 197 |

| | | |
|---------|---------------------------------------------------------------------------------------|-----|
| 5.14.12 | Sub block 0x102B – AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 198 |
| 5.14.13 | Sub block 0x102C – AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB | 199 |
| 5.14.14 | Sub block 0x102D – AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB | 200 |
| 5.14.15 | Sub block 0x102E – AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB | 202 |
| 5.14.16 | Sub block 0x1031 – AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB | 203 |
| 5.15 | Sub blocks related to AWR_DEV_RFPOWERUP_MSG | 204 |
| 5.15.1 | Sub block 0x4000 – AWR_DEV_RFPOWERUP_SB | 204 |
| 5.16 | Sub blocks related to AWR_DEV_CONF_SET_MSG | 205 |
| 5.16.1 | Sub block 0x4040 – AWR_DEV_MCUCLOCK_CONF_SET_SB | 205 |
| 5.16.2 | Sub block 0x4041 – AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB | 206 |
| 5.16.3 | Sub block 0x4042 – AWR_DEV_RX_DATA_PATH_CONF_SET_SB | 207 |
| 5.16.4 | Sub block 0x4043 – AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB | 210 |
| 5.16.5 | Sub block 0x4044 – AWR_DEV_RX_DATA_PATH_CLK_SET_SB | 211 |
| 5.16.6 | Sub block 0x4045 – AWR_DEV_LVDS_CFG_SET_SB | 212 |
| 5.16.7 | Sub block 0x4046 – AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB | 214 |
| 5.16.8 | Sub block 0x4047 – AWR_DEV_CSI2_CFG_SET_SB | 215 |
| 5.16.9 | Sub block 0x4048 – AWR_DEV_PMICCLOCK_CONF_SET_SB | 217 |
| 5.16.10 | Sub block 0x4049 – AWR_MSS_PERIODICTESTS_CONF_SB | 221 |
| 5.16.11 | Sub block 0x404A – AWR_MSS_LATENTFAULT_TEST_CONF_SB | 222 |
| 5.16.12 | Sub block 0x404B – AWR_DEV_TESTPATTERN_GEN_SET_SB | 224 |
| 5.16.13 | Sub block 0x404C – AWR_DEV_CONFIGURATION_SET_SB | 227 |
| 5.17 | Sub blocks related to AWR_DEV_CONF_GET_MSG | 227 |
| 5.17.1 | Sub block 0x4060 – AWR_DEV_MCUCLOCK_GET_SB | 227 |
| 5.17.2 | Sub block 0x4061 – AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB | 228 |
| 5.17.3 | Sub block 0x4062 – AWR_DEV_RX_DATA_PATH_CONF_GET_SB | 228 |
| 5.17.4 | Sub block 0x4063 – AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB | 228 |
| 5.17.5 | Sub block 0x4064 – AWR_DEV_RX_DATA_PATH_CLK_GET_SB | 229 |
| 5.17.6 | Sub block 0x4065 – AWR_DEV_LVDS_CFG_GET_SB | 229 |
| 5.17.7 | Sub block 0x4066 – AWR_DEV_RX_CONTSTREAMING_MODE_CONF_GET_SB | 229 |
| 5.17.8 | Sub block 0x4067 – AWR_DEV_CSI2_CFG_GET_SB | 229 |
| 5.17.9 | Sub block 0x4068 – AWR_DEV_PMICCLOCK_CONF_GET_SB | 230 |
| 5.17.10 | Sub block 0x4069 – AWR_MSS_LATENTFAULT_TEST_CONF_GET_SB | 230 |
| 5.17.11 | Sub block 0x406A – AWR_MSS_PERIODICTESTS_CONF_GET_SB | 230 |
| 5.17.12 | Sub block 0x406B – AWR_DEV_TESTPATTERN_GEN_GET_SB | 231 |
| 5.18 | Sub blocks related to AWR_DEV_FILE_DOWNLOAD_MSG | 231 |
| 5.18.1 | Sub block 0x4080 – AWR_DEV_FILE_DOWNLOAD_SB | 231 |
| 5.19 | Sub blocks related to AWR_DEV_FRAME_CONFIG_APPLY_MSG | 232 |

| | | |
|----------|---------------------------------------------------------------------|------------|
| 5.19.1 | Sub block 0x40C0 – AWR_DEV_FRAME_CONFIG_APPLY_SB | 232 |
| 5.19.2 | Sub block 0x40C1 – AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB | 232 |
| 5.20 | Sub blocks related to AWR_DEV_STATUS_GET_MSG | 234 |
| 5.20.1 | Sub block 0x40E0 – AWR_MSSVERSION_GET_SB | 234 |
| 5.20.2 | Sub block 0x40E1 – AWR_MSSCPUFAULT_STATUS_GET_SB | 236 |
| 5.20.3 | Sub block 0x40E2 – AWR_MSSSEMFault_STATUS_GET_SB | 237 |
| 5.21 | Sub blocks related to AWR_DEV_ASYNC_EVENT_MSG | 242 |
| 5.21.1 | Sub block 0x5000 – AWR_AE_DEV_MSSPOWERUPDONE_SB | 242 |
| 5.21.2 | Sub block 0x5001 – AWR_AE_DEV_RFPOWERUPDONE_SB | 244 |
| 5.21.3 | Sub block 0x5002 – AWR_AE_MSS_CPUFAULT_SB | 246 |
| 5.21.4 | Sub block 0x5003 – AWR_AE_MSS_ESMFAULT_STATUS_SB | 247 |
| 5.21.5 | Sub block 0x5004 – RESERVED | 250 |
| 5.21.6 | Sub block 0x5005 – AWR_AE_MSS_BOOTERRORSTATUS_SB | 250 |
| 5.21.7 | Sub block 0x5006 – AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB | 252 |
| 5.21.8 | Sub block 0x5007 – AWR_AE_MSS_PERIODICTEST_STATUS_SB | 254 |
| 5.21.9 | Sub block 0x5008 – AWR_AE_MSS_RFERROR_STATUS_SB | 254 |
| 5.21.10 | Sub block 0x5009 – AWR_AE_MSS_VMON_ERRORSTATUS_SB | 255 |
| 5.21.11 | Sub block 0x500A – AWR_AE_MSS_ADC_DATA_SB | 256 |
| 5.21.12 | Sub block 0x500B – RESERVED | 256 |
| 5.22 | Brief notes on the order of issuing API SBs | 256 |
| 5.22.1 | Single device mode | 256 |
| 5.22.2 | Cascaded device mode | 259 |
| 5.22.3 | Continuous streaming mode (in single device case) | 260 |
| 5.22.4 | Continuous streaming (CW) mode (in cascaded device case) | 261 |
| 6 | API Error Codes | 264 |
| 6.1 | Error codes for boot on SPI | 273 |
| 7 | Radar Monitoring APIs | 275 |
| 7.1 | Common Configurations and Reports | 275 |
| 7.1.1 | Sub block 0x01C0 – AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB | 276 |
| 7.1.2 | Sub block 0x01C1 – AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB | 278 |
| 7.1.3 | Sub block 0x01C2 – AWR_MONITOR_ANALOG_ENABLES_CONF_SB | 279 |
| 7.2 | Temperature Monitor | 281 |
| 7.2.1 | Sub block 0x01C3 – AWR_MONITOR_TEMPERATURE_CONF_SB | 281 |
| 7.3 | RX Gain and Phase Monitor | 283 |
| 7.3.1 | Sub block 0x01C4 – AWR_MONITOR_RX_GAIN_PHASE_CONF_SB | 283 |
| 7.4 | RX Noise Monitor | 286 |
| 7.4.1 | Sub block 0x01C5 – AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB | 286 |
| 7.5 | RX IF Stage Monitor | 288 |
| 7.5.1 | Sub block 0x01C6 – AWR_MONITOR_RX_IFSTAGE_CONF_SB | 288 |
| 7.6 | TX Power Monitor | 289 |
| 7.6.1 | Sub block 0x01C7 – AWR_MONITOR_TX0_POWER_CONF_SB | 289 |
| 7.6.2 | Sub block 0x01C8 – AWR_MONITOR_TX1_POWER_CONF_SB | 291 |

| | | |
|--------|------------------------------------------------------------------------|-----|
| 7.6.3 | Sub block 0x01C9 – AWR_MONITOR_TX2_POWER_CONF_SB | 292 |
| 7.7 | TX Ball Break Monitor | 294 |
| 7.7.1 | Sub block 0x01CA – AWR_MONITOR_TX0_BALLBREAK_CONF_SB | 294 |
| 7.7.2 | Sub block 0x01CB – AWR_MONITOR_TX1_BALLBREAK_CONF_SB | 295 |
| 7.7.3 | Sub block 0x01CC – AWR_MONITOR_TX2_BALLBREAK_CONF_SB | 295 |
| 7.8 | TX Gain and Phase Mismatch Monitoring | 296 |
| 7.8.1 | Sub block 0x01CD – AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB | 297 |
| 7.9 | TX BPM Phase Monitor | 299 |
| 7.9.1 | Sub block 0x01CE – AWR_MONITOR_TX0_BPM_CONF_SB | 300 |
| 7.9.2 | Sub block 0x01CF – AWR_MONITOR_TX1_BPM_CONF_SB | 301 |
| 7.9.3 | Sub block 0x01D0 – AWR_MONITOR_TX2_BPM_CONF_SB | 303 |
| 7.10 | Synthesizer Frequency Monitoring | 305 |
| 7.10.1 | Sub block 0x01D1 – AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB | 305 |
| 7.11 | External Analog Signals Monitor | 306 |
| 7.11.1 | Sub block 0x01D2 – AWR_MONITORING_EXTERNAL_ANALOG_SIGNALS_CONF_SB | 307 |
| 7.12 | Internal Analog Signals Monitor | 310 |
| 7.12.1 | Sub block 0x01D3 – AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB | 311 |
| 7.12.2 | Sub block 0x01D4 – AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB | 312 |
| 7.12.3 | Sub block 0x01D5 – AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB | 313 |
| 7.12.4 | Sub block 0x01D6 – AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB | 313 |
| 7.12.5 | Sub block 0x01D7 – AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB | 314 |
| 7.12.6 | Sub block 0x01D8 – AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB | 315 |
| 7.13 | PLL Control Voltage Monitor | 316 |
| 7.13.1 | Sub block 0x01D9 – AWR_MONITOR_PLL_CONTROL_VOLTAGE_SIGNALS_CONF_SB | 316 |
| 7.14 | Dual Clock Comparator Based Clock Frequency Monitor | 317 |
| 7.14.1 | Sub block 0x01DA – AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB | 318 |
| 7.15 | RX Saturation Detection Monitor | 319 |
| 7.15.1 | Sub block 0x01DB – AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB | 320 |
| 7.15.2 | Sub block 0x01DC – AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB | 321 |
| 7.16 | RX mixer input power monitor | 323 |
| 7.16.1 | Sub block 0x01DD – AWR_MONITOR_RX_MIXER_IN_POWER_CONF_SB | 323 |
| 7.17 | Sub block 0x01DE – RESERVED | 324 |
| 7.18 | Analog Fault injection | 324 |

| | | |
|-----------|-------------------------------------------------------------|------------|
| 7.18.1 | Sub block 0x01DF – AWR_ANALOG_FAULT_INJECTION_CONF_SB . . . | 324 |
| 8 | Unsupported Features/APIs and Debug APIs | 331 |
| 8.1 | Unsupported Features/APIs and Debug APIs | 331 |
| 9 | Chirp Parameters (CP) and Chirp Quality (CQ) data | 332 |
| 9.1 | Chirp Parameters data | 332 |
| 9.2 | Chirp Quality data | 333 |
| 9.2.1 | CQ1 | 335 |
| 9.2.2 | CQ2 | 338 |
| 10 | Calibration and monitoring durations | 341 |
| 10.1 | Boot time calibration durations | 341 |
| 10.2 | Run time calibration durations | 341 |
| 10.3 | Monitoring duration | 342 |
| 10.4 | Software overheads | 344 |
| 10.4.1 | Note on idle time for clearing the watchdog (WDT) | 344 |
| 10.5 | Sample Application | 349 |

List of Figures

| | | |
|------|----------------------------------------------------------------------------|-----|
| 2.1 | AWR12xx, AWR22xx Software Architecture | 4 |
| 2.2 | xWR16xx, xWR18xx and IWR68xx Software Architecture | 5 |
| 2.3 | Radar Message Structure | 5 |
| 2.4 | Message Header Format | 6 |
| 2.5 | OPCODE Format | 7 |
| 2.6 | MSGLEN Format | 10 |
| 2.7 | FLAGS Format | 10 |
| 2.8 | NSBC Format | 11 |
| 2.9 | Message Sub block structure | 13 |
| 3.1 | Flow Diagram (API) | 16 |
| 3.2 | Flow Diagram (Asynchronous Events) | 17 |
| 3.3 | SPI Message Sequence | 18 |
| 5.1 | Frame trigger delay in case of external hardware trigger | 76 |
| 5.2 | Frame trigger delay in case of external hardware trigger | 89 |
| 5.3 | Dynamic chirp configuration use case timing diagram | 115 |
| 5.4 | Lane formats and the order of receiving the data from the lanes | 214 |
| 9.1 | Chirp parameter information fields | 332 |
| 9.2 | Chirp parameter information from DSS registers | 333 |
| 9.3 | CQ data start address configuration in single chirp use case | 334 |
| 9.4 | CQ data start address configuration in multi chirp use case | 335 |
| 9.5 | Time slices during RX signal and image band monitor and saturation monitor | 336 |
| 9.6 | CQ1 data format in memory in 16-bit mode | 337 |
| 9.7 | CQ1 data format in memory in 12-bit mode | 338 |
| 9.8 | CQ1 data format in memory in 14-bit mode | 338 |
| 9.9 | CQ2 data format in memory in 16-bit mode | 339 |
| 9.10 | CQ2 data format in memory in 12-bit mode | 340 |
| 9.11 | CQ2 data format in memory in 14-bit mode | 340 |
| 10.1 | Watchdog idle time calculation | 345 |

List of Tables

| | | |
|------|---------------------------------------------------------------|----|
| 1.1 | TI CMOS radar devices | 1 |
| 2.1 | Possible SYNC values and their usage | 6 |
| 2.3 | MSGLEN field descriptions | 10 |
| 2.4 | FLAGS field description | 10 |
| 2.5 | NSBC field description | 12 |
| 2.6 | Checksum computation example | 12 |
| 2.7 | CRC types and their polynomials | 13 |
| 4.1 | Summary of all Radar messages and their associated sub blocks | 19 |
| 5.1 | AWR_RESP_ERROR_SB contents | 43 |
| 5.2 | AWR_CHAN_CONF_SET_SB contents | 44 |
| 5.3 | AWR_ADCOUT_CONF_SB contents | 47 |
| 5.4 | AWR_LOWPOWERMODE_CONF_SET_SB contents | 48 |
| 5.5 | AWR_DYNAMICPOWERSAVE_CONF_SET_SB contents | 49 |
| 5.6 | AWR_HIGHSPEEDINTFCLK_CONF_SET_SB contents | 50 |
| 5.7 | AWR_RF_DEVICE_CFG_SB contents | 51 |
| 5.8 | AWR_RF_MISC_CTL_SB contents | 53 |
| 5.9 | AWR_CAL_MON_FREQUENCY_LIMITS_SB contents | 53 |
| 5.10 | AWR_RF_INIT_CALIBRATION_CONF_SB contents | 54 |
| 5.11 | AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB contents | 56 |
| 5.12 | AWR_CAL_DATA_RESTORE_SB contents | 58 |
| 5.13 | AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB contents | 59 |
| 5.14 | AWR_CAL_DATA_SAVE_SB contents | 60 |
| 5.15 | AWR_CAL_DATA_SAVE_SB response packet contents | 60 |
| 5.16 | AWR_CAL_DATA contents | 60 |
| 5.17 | AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB contents | 63 |
| 5.18 | AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB response packet contents | 63 |
| 5.19 | AWR_RF_INIT_SB contents | 64 |
| 5.20 | AWR_PROFILE_CONF_SB contents | 65 |
| 5.21 | Note on maximum sampling rate | 72 |
| 5.22 | AWR_CHIRP_CONF_SET_SB contents | 73 |
| 5.23 | AWR_FRAME_CONF_SET_SB contents | 74 |
| 5.24 | AWR_CONT_STREAMING_MODE_CONF_SET_SB contents | 77 |
| 5.25 | AWR_CONT_STREAMING_MODE_EN_SB contents | 80 |

| | | |
|------|-------------------------------------------------------|-----|
| 5.26 | AWR_ADVANCED_FRAME_CONF_SB contents | 81 |
| 5.27 | AWR_PERCHIRPPHASESHIFT_CONF_SB contents | 90 |
| 5.28 | AWR_PROG_FILT_COEFF_RAM_SET_SB contents | 92 |
| 5.29 | AWR_PROG_FILT_CONF_SET_SB contents | 92 |
| 5.30 | AWR_CALIB_MON_TIME_UNIT_CONF_SB contents | 93 |
| 5.31 | AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB contents | 95 |
| 5.32 | AWR_INTER_RX_GAIN_PHASE_CONTROL_SB contents | 99 |
| 5.33 | AWR_RX_GAIN_TEMPLUT_SET_SB contents | 99 |
| 5.34 | AWR_TX_GAIN_TEMPLUT_SET_SB contents | 101 |
| 5.35 | AWR_LOOPBACK_BURST_CONF_SET_SB contents | 105 |
| 5.36 | AWR_DYN_CHIRP_CONF_SET_SB contents | 109 |
| 5.37 | AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SB contents | 113 |
| 5.38 | AWR_DYN_CHIRP_ENABLE_SB contents | 114 |
| 5.39 | AWR_INTERCHIRP_BLOCKCONTROLS_SB contents | 115 |
| 5.40 | AWR_SUBFRAME_START_CONF_SB contents | 118 |
| 5.41 | AWR_PROFILE_CONF_GET_SB contents | 119 |
| 5.42 | AWR_CHIRP_CONF_GET_SB contents | 119 |
| 5.43 | AWR_FRAME_CONF_GET_SB contents | 120 |
| 5.44 | AWR_ADV_FRAME_CONF_GET_SB contents | 120 |
| 5.45 | AWR_RX_GAIN_TEMPLUT_GET_SB contents | 121 |
| 5.46 | AWR_TX_GAIN_TEMPLUT_GET_SB contents | 121 |
| 5.47 | AWR_FRAMESTARTSTOP_CONF_SB contents | 122 |
| 5.48 | AWR_BPM_COMMON_CONF_SET_SB contents | 122 |
| 5.49 | AWR_BPM_CHIRP_CONF_SET_SB contents | 123 |
| 5.50 | AWR_RF_VERSION_GET_SB contents | 124 |
| 5.51 | AWR_RF_VERSION_SB response contents | 125 |
| 5.52 | AWR_RF_CPUFAULT_STATUS_GET_SB response contents | 126 |
| 5.53 | AWR_RF_CPUFAULT_STATUS_GET_SB response contents | 126 |
| 5.54 | AWR_RF_ESMFAULT_STATUS_GET_SB response contents | 128 |
| 5.55 | AWR_RF_ESMFAULT_STATUS_SB response contents | 128 |
| 5.56 | AWR_RF_DIEID_GET_SB response contents | 131 |
| 5.57 | AWR_RF_DIEID_STATUS_SB response contents | 131 |
| 5.58 | AWR_RF_BOOTUPBIST_STATUS_GET_SB response contents | 131 |
| 5.59 | AWR_RF_BOOTUPBIST_STATUS_DATA_SB response contents | 132 |
| 5.60 | AWR_RF_DFE_STATISTICS_REPORT_GET_SB response contents | 134 |
| 5.61 | AWR_RF_DFE_STATISTICS_REPORT_SB response contents | 134 |
| 5.62 | AWR_RF_TEST_SOURCE_CONFIG_SET_SB contents | 143 |
| 5.63 | AWR_RF_TEST_SOURCE_ENABLE_SET_SB contents | 145 |
| 5.64 | AWR_RF_LDO_BYPASS_SB contents | 145 |
| 5.65 | AWR_RF_PALOOPBACK_CFG_SB contents | 147 |
| 5.66 | AWR_RF_PSLOOPBACK_CFG_SB contents | 148 |
| 5.67 | AWR_RF_IFLOOPBACK_CFG_SB contents | 150 |
| 5.68 | AWR_RF_GPADC_CFG_SET_SB contents | 151 |
| 5.69 | AWR_RF_TEMPERATURE_GET_SB contents | 154 |

| | | |
|-------|---------------------------------------------------------------------|-----|
| 5.70 | AWR_RF_TEMPERATURE_DATA_SB contents | 154 |
| 5.71 | AWR_AE_RF_CPUFAULT_SB response contents | 155 |
| 5.72 | AWR_AE_RF_ESMFAULT_STATUS_SB response contents | 157 |
| 5.73 | AWR_AE_RF_INITCALIBSTATUS_SB response contents | 159 |
| 5.74 | AWR_AE_RF_FRAME_TRIGGER_RDY_SB response contents | 161 |
| 5.75 | AWR_AE_RF_GPADC_RESULT_DATA_SB response contents | 161 |
| 5.76 | AWR_FRAME_END_AE_SB response contents | 163 |
| 5.77 | AWR_ANALOGFAULT_AE_SB response contents | 164 |
| 5.78 | AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB response contents | 164 |
| 5.79 | AWR_RUN_TIME_CALIB_SYMMARY_REPORT_AE_SB response contents | 165 |
| 5.80 | AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB response contents | 168 |
| 5.81 | AWR_MONITORING_REPORT_HEADER_AE_SB response contents | 170 |
| 5.82 | AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB contents | 170 |
| 5.83 | AWR_MONITORING_TEMPERATURE_REPORT_AE_SB contents | 171 |
| 5.84 | AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB contents | 172 |
| 5.85 | AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB contents | 176 |
| 5.86 | AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB contents | 178 |
| 5.87 | AWR_MONITOR_TX0_POWER_REPORT_AE_SB contents | 180 |
| 5.88 | AWR_MONITOR_TX1_POWER_REPORT_AE_SB contents | 182 |
| 5.89 | AWR_MONITOR_TX2_POWER_REPORT_AE_SB contents | 183 |
| 5.90 | AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB contents | 184 |
| 5.91 | AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB contents | 184 |
| 5.92 | AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB contents | 185 |
| 5.93 | AWR_MONITOR_TX_GAIN_PHASE_REPORT_AE_SB contents | 186 |
| 5.94 | AWR_MONITOR_TX0_BPM_REPORT_AE_SB contents | 188 |
| 5.95 | AWR_MONITOR_TX1_BPM_REPORT_AE_SB contents | 189 |
| 5.96 | AWR_MONITOR_TX2_BPM_REPORT_AE_SB contents | 191 |
| 5.97 | AWR_MONITOR_SYNTH_FREQUENCY_REPORT_AE_SB contents | 192 |
| 5.98 | AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 193 |
| 5.99 | AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 195 |
| 5.100 | AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 195 |
| 5.101 | AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 196 |
| 5.102 | AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 197 |
| 5.103 | AWR_MONITOR_PM_CLK_LO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 198 |
| 5.104 | AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents | 199 |
| 5.105 | AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB contents | 200 |
| 5.106 | AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB contents | 202 |
| 5.107 | AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB contents | 203 |
| 5.108 | AWR_DEV_POWERUP_SB contents | 205 |
| 5.109 | AWR_DEV_MCUCLOCK_CONF_SET_SB contents | 205 |
| 5.110 | AWR_DEV_RX_DATA_FORMAT_CONF_SB contents | 206 |
| 5.111 | AWR_DEV_RX_DATA_PATH_CONF_SB contents | 207 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.112AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB contents | 210 |
| 5.113AWR_DEV_RX_DATA_PATH_CLK_SET_SB contents | 211 |
| 5.114AWR_DEV_LVDS_CFG_SET_SB contents | 212 |
| 5.115AWR_DEV_RX_CONTSTREAMING_MODE_CFG_SET_SB contents | 215 |
| 5.116AWR_DEV_CSI2_CFG_SET_SB contents | 215 |
| 5.117AWR_DEV_PMICCLOCK_CONF_SET_SB contents | 217 |
| 5.118PMIC clock frequency across chirps in chirp-to-chirp staircase mode in an example when PMIC clock varies from 2 MHz to 2.5 MHz in 32 chirps | 220 |
| 5.119AWR_MSS_PERIODICTESTS_CONF_SB contents | 221 |
| 5.120AWR_MSS_LATENTFAULT_TEST_CONF_SB contents | 222 |
| 5.121AWR_DEV_TESTPATTERN_GEN_SET_SB contents | 224 |
| 5.122AWR_DEV_CONFIGURATION_SET_SB contents | 227 |
| 5.123AWR_DEV_MCUCLOCK_GET_SB contents | 228 |
| 5.124AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB contents | 228 |
| 5.125AWR_DEV_RX_DATA_PATH_CONF_GET_SB contents | 228 |
| 5.126AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB contents | 228 |
| 5.127AWR_DEV_RX_DATA_PATH_CLK_GET_SB contents | 229 |
| 5.128AWR_DEV_LVDS_CFG_GET_SB contents | 229 |
| 5.129AWR_DEV_RX_CONTSTREAMING_CONF_GET_SB contents | 229 |
| 5.130AWR_DEV_CSI2_CFG_GET_SB contents | 230 |
| 5.131AWR_DEV_PMICCLOCK_CONF_GET_SB contents | 230 |
| 5.132AWR_MSS_LATENTFAULT_CONF_GET_SB contents | 230 |
| 5.133AWR_MSS_PERIODICTESTS_CONF_GET_SB contents | 230 |
| 5.134AWR_DEV_TESTPATTERN_GEN_GET_SB contents | 231 |
| 5.135AWR_DEV_FILE_DOWNLOAD_SB contents | 231 |
| 5.136AWR_DEV_FRAME_CONFIG_APPLY_SB contents | 232 |
| 5.137AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB contents | 232 |
| 5.138AWR_MSSVERSION_GET_SB contents | 234 |
| 5.139AWR_MSSVERSION_SB contents | 235 |
| 5.140AWR_MSSVERSION_SB contents | 236 |
| 5.141AWR_MSSCPUFAULT_STATUS_SB contents | 236 |
| 5.142AWR_MSSSEMFFAULT_STATUS_GET_SB contents | 239 |
| 5.143AWR_MSSSEMFFAULT_STATUS_SB contents | 239 |
| 5.144AWR_AE_DEV_MSSPOWERUPDONE_SB contents | 242 |
| 5.145AWR_AE_DEV_RFPOWERUPDONE_SB contents | 244 |
| 5.146AWR_AE_MSS_CPUFAULT_STATUS_SB contents | 246 |
| 5.147AWR_AE_MSS_ESMFAULT_STATUS_SB contents | 247 |
| 5.148AWR_AE_MSS_BOOTERRORSTATUS_SB contents | 250 |
| 5.149AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB contents | 252 |
| 5.150AWR_AE_MSS_PERIODICTEST_STATUS_SB contents | 254 |
| 5.151AWR_AE_MSS_RFERROR_STATUS_SB contents | 254 |
| 5.152AWR_AE_MSS_VMON_ERRORSTATUS_SB contents | 255 |
| 5.153AWR_AE_MSS_ADC_DATA_SB contents | 256 |

| | | |
|-------|-------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.154 | Sequence of APIs to be issued to master and slave devices in cascaded mode configuration for FMCW mode measurements | 259 |
| 5.155 | Sequence of APIs to be issued to master and slave devices in cascaded mode for CW mode measurements | 261 |
| 6.1 | BSS API error codes | 264 |
| 6.2 | MSS API error codes (Applicable only in AWR1243) | 271 |
| 6.3 | Bit field describing the error status during boot on SPI | 274 |
| 7.1 | AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB contents | 276 |
| 7.2 | AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB contents | 278 |
| 7.3 | AWR_MONITOR_ANALOG_ENABLES_CONF_SB contents | 279 |
| 7.4 | AWR_MONITOR_TEMPERATURE_CONF_SB contents | 281 |
| 7.5 | AWR_MONITOR_RX_GAIN_PHASE_CONF_SB contents | 283 |
| 7.6 | AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB contents | 287 |
| 7.7 | AWR_MONITOR_RX_IFSTAGE_CONF_SB contents | 288 |
| 7.8 | AWR_MONITOR_TX0_POWER_CONF_SB contents | 289 |
| 7.9 | AWR_MONITOR_TX1_POWER_CONF_SB contents | 291 |
| 7.10 | AWR_MONITOR_TX2_POWER_CONF_SB contents | 292 |
| 7.11 | AWR_MONITOR_TX0_BALLBREAK_CONF_SB contents | 294 |
| 7.12 | AWR_MONITOR_TX1_BALLBREAK_CONF_SB contents | 295 |
| 7.13 | AWR_MONITOR_TX2_BALLBREAK_CONF_SB contents | 296 |
| 7.14 | AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB contents | 297 |
| 7.15 | AWR_MONITOR_TX0_BPM_CONF_SB contents | 300 |
| 7.16 | AWR_MONITOR_TX1_BPM_CONF_SB contents | 302 |
| 7.17 | AWR_MONITOR_TX2_BPM_CONF_SB contents | 303 |
| 7.18 | AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB contents | 305 |
| 7.19 | AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_CONF_SB contents | 307 |
| 7.20 | AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 311 |
| 7.21 | AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 312 |
| 7.22 | AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 313 |
| 7.23 | AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 313 |
| 7.24 | AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 314 |
| 7.25 | AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB contents | 315 |
| 7.26 | AWR_MONITOR_PLL_CONTROL_VOLTAGE_CONF_SB contents | 316 |
| 7.27 | DCC Clock monitor pairs | 318 |
| 7.28 | AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB contents | 318 |
| 7.29 | AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB contents | 320 |
| 7.30 | AWR_MONITOR_RX_SIG_IMG_MONITOR_CONF_SB contents | 321 |
| 7.31 | AWR_MONITOR_MIXER_IN_POWER_CONF_SB contents | 323 |
| 7.32 | AWR_ANALOG_FAULT_INJECTION_CONF_SB contents | 325 |
| 10.1 | Duration of boot time calibrations | 341 |
| 10.2 | Duration of run time calibrations | 342 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------|-----|
| 10.3 Duration of analog monitors | 343 |
| 10.4 Duration of digital monitors | 343 |
| 10.5 Software overheads every FTTI that should be accounted to program CALIB_ MON_TIME_UNIT and CALIBRATION_PERIODICITY | 344 |

Revision History

| Revision | Date | Description |
|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.97 | 13.09.2018 | <ol style="list-style-type: none">1. Added a new parameter CASCADING_PINOUT_CFG in AWR_CHAN_CONF_SET_SB in Section 5.2.1 on page 44.2. Added a new parameter PA LDO disable in AWR_RF_LDO_BYPASS_SB API in Section 5.11.6 on page 145.3. Added new APIs AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB in page 62 and AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB in page 59.4. Added a new parameter LDO_SC_MONITORING_EN in AWR_MONITOR_ANALOG_ENABLES_CONF_SB in page 279.5. Fixed the length for the SBLKID AWR_DEV_CONFIGURATION_SET_SB in page 227.6. Added a note in AWR_CHIRP_CONF_SET_SB in page 73 that dither parameters are only additive to the programmed parameters in AWR_PROFILE_CONF_SB.7. Updated the valid range of SFx_PERIOD parameter in AWR_ADVANCED_FRAME_CONF_SB in page 80 to 1.342 s.8. Added a note below AWR_SUBFRAME_START_CONF_SB on page 118 indicating that watchdog feature is not available when software based sub-frame trigger mode is used.9. Added new parameters in AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB in page 314 for 20 GHz sync signal monitoring. Also updated AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB in page 198 with 20 GHz monitoring report. |

10. Added parameters in `AWR_MONITOR_TXn_BPM_CONF_SB` in page numbers 188, 189 and 191 for phase shifter monitoring. Also updated `AWR_MONITOR_TXn_BPM_REPORT_AE_SB` in pages 188, 189 and 191 for phase shifter monitoring reports.
11. Added a note in page 93 indicating that programmable filter APIs should not be issued when frames are ongoing.
12. Updated the mapping of `PGA_GAIN_INDEX` numbers to gain values in `AWR_RF_PSLOOPBACK_CFG_SB` in page 148.
13. Added a note after `AWR_BPM_CHIRP_CONF_SET_SB` in page 123 and `AWR_PERCHIRPPHASESHIFT_CONF_SB` in page 90 indicating when the BPM and phase shifters are applied.
14. Added `TX_CAL_EN_CFG` parameter in `AWR_PROFILE_CONF_SET_SB` in page 64.
15. Added a note in `AWR_RF_INIT_CALIBRATION_CONF_SB` in page 54 indicating that if TX boot time calibration is disabled, no other backoff other than 0 dB is supported.
16. Added the monitoring duration of TX phase shifter in Table 10.3 in page 343.
17. Added a new API `AWR_RF_DIEID_GET_SB` in page 130 which reads the Die ID of the device.
18. Added a note in section `AWR_LOOPBACK_BURST_CONF_SET_SB` in page 104 indicating that when using loopback burst, the corresponding sub-frame in advanced frame configuration should use `SFx_NUM_UNIQUE_CHIRPS_PER_BURST` as 1.
19. Corrected the sequence of issuing APIs in page 256 - loopback burst config API should be issued after profile config API.
20. Updated error code 49 in Table 6.1 to include the maximum sampling rate based on device variant.
21. Added a note in page 344 explaining the watchdog clearing window calculation by the firmware.
22. Added a note in page 97 indicating if user has not enabled any one time calibrations, but if calibration report is enabled, then after issuing the `AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB` API, the firmware will immediately sent out a calibration report.
23. Added the `RFLDOBYPASS_EN` API after `AWR_ADCOUT_CONF_SET_SB` in Table 5.22.1, Table 5.22.3, Table 5.154 and Table 5.155.

24. Added a note after [AWR_FRAME_CONF_SET_SB](#) indicating the pulse width requirements of the SYNC.IN pulse in hardware triggered mode.
25. Added a new parameter CHIRP_ROW_SELECT in [AWR_DYN_CHIRP_CONF_SET_SB](#) API in page 109 to enable faster configuration of chirps dynamically.
26. Updated the definition of REPORTING_MODE in [AWR_MSS_LATENTFAULT_TEST_CONF_SB](#) and [AWR_MSS_PERIODICTESTS_CONF_SB](#) APIs in pages 221 and 222.
27. Added new error code 159 to indicate incorrect CHIRP_ROW_SELECT value and updated error code 135 to account for CHIRP_ROW_SELECT values.
28. Added new error codes 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048 and 1050 in Table 6.2 in page 271.
29. Added examples of PMIC clock configuration in Section 5.16.9 in page 217.
30. Added a note in [AWR_ANALOG_FAULT_INJECTION_CONF_SB](#) in page 324 under SUPPLY_LDO_FAULT indicating that this fault ineffective under LDO.BYPASS condition.
31. Added a note about when to stop the frames when using sub-frame trigger or hardware trigger mode in page 117.
32. Updated font type to Helvetica.
33. Added a note about usage of CW CZ mode in page 77.
34. Added a note about TX3 gain and phase imbalance offset wrt. TX1 and TX2 in page 297.
35. Added a note about inter-burst idle time requirement in page 48.
36. Added a note about Noise figure Values reporting in NF Monitoring report AE in page 176.
37. Added a note in MSS powerup done AE in page 242.
38. Added a note about Tx output Power backoff in profile config API in page 64.
39. Added a note about Tx output Power monitoring in page 180.
40. Added a note about Programmable filter tap start index selection in page 91.
41. Added a note in [AWR_AE_MSS_BOOTERRORSTATUS_SB](#) AE in page 250.
42. Updated runtime VCO calibration time in page 341.
43. Updated VCO control voltage monitor upper threshold in page 200.

| Revision | Date | Description |
|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0 | 24.09.2018 | <ol style="list-style-type: none"> Changed document name to "mmWave-Radar-Interface-Control.pdf" Updated supported device list and added two new IWR6843 60GHz and xWR1843 77GHz TI mmWave radar devices Added 60GHz RF frequency and slope programming details in following API sub-blocks AWR_CAL_MON_FREQUENCY_LIMITS_SB AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB AWR_PROFILE_CONF_SB AWR_CHIRP_CONF_SET_SB AWR_CONT_STREAMING_MODE_CONF_SET_SB AWR_LOOPBACK_BURST_CONF_SET_SB AWR_DYN_CHIRP_CONF_SET_SB Added IWR6843 ES1.0 limitations in following API sub-blocks AWR_CAL_MON_FREQUENCY_LIMITS_SB AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB AWR_RF_INIT_CALIBRATION_CONF_SB AWR_CAL_DATA_RESTORE_SB AWR_CAL_DATA_SAVE_SB AWR_CAL_DATA_SAVE_SB AWR_PROFILE_CONF_SB AWR_CONT_STREAMING_MODE_CONF_SET_SB AWR_PERCHIRPPHASESHIFT_CONF_SB AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB AWR_RX_GAIN_TEMPLUT_SET_SB AWR_TX_GAIN_TEMPLUT_SET_SB AWR_LOOPBACK_BURST_CONF_SET_SB AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SET_SB AWR_SUBFRAME_START_CONF_SB AWR_RX_GAIN_TEMPLUT_GET_SB AWR_TX_GAIN_TEMPLUT_GET_SB AWR_BPM_COMMON_CONF_SET_SB AWR_BPM_CHIRP_CONF_SET_SB AWR_RF_BOOTUPBIST_STATUS_DATA_SB AWR_RF_PALOOPBACK_CFG_SB AWR_RF_PSLOOPBACK_CFG_SB AWR_RF_IFLOOPBACK_CFG_SB AWR_RF_TEMPERATURE_GET_SB AWR_RF_DIEID_GET_SB AWR_DYN_CHIRP_CONF_SET_SB AWR_RF_TEST_SOURCE_CONFIG_SET_SB Radar Monitoring APIs Async event sub-blocks |

| Revision | Date | Description |
|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1 | 10.10.2018 | <ol style="list-style-type: none"> Updated AWR_DEV_LVDS_CFG_SET_SB subblock in page 212. Updated note on VCO selection in page 64 Added a note in Noise figure monitoring configuration SBC 286 |

| Revision | Date | Description |
|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.2 | 19.01.2019 | <ol style="list-style-type: none"> Updated 60GHz VCO2 frequency range and Tx output power backoff support for IWR6843 in page 64 Updated MONITOR_START_TIME range in AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB in page 305 Removed ESM periodic monitoring support in AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB in page 278 Added TX Phase Shifter Status flag in AWR_MONITOR_TX0/1/2_BPM_REPORT_AE_SB in page 188 Added Programmable filter Fatal error flags in BSS ESM G1 AE report AWR_AE_RF_ESMFAULT_STATUS_SB in page 157 Added FRC Lock step Fatal error flag in MSS ESM G1 AE report AWR_AE_MSS_CPUFAULT_STATUS_SB in page 247 Added a note on timing and handling of monitoring Async Events at the beginning of AWR_RF_ASYNC_EVENT_MSG1 section in page 155 Added a note on Frame Stop functionality and timing in AWR_FRAMESTARTSTOP_CONF_SB in page 121 Added a note on MSS max monitoring period in AWR_MSS_PERIODICTESTS_CONF_SB in page 221 Removed Rx Power detect monitoring and its status in AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB in page 197 Added a note on WDT timer enable conditions in AWR_RF_DEVICE_CFG_SB in page 50 Added a note on external GPADC signal measurement report in AWR_RF_GPADC_CFG_SET_SB in page 150 Updated Calibration periodicity valid range in AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB in page 94 Updated CAL_MON_TIME_UNIT period valid range in AWR_CALIB_MON_TIME_UNIT_CONF_SB in page 93 Updated note on Noise figure Values reported in NF Monitoring report AE in page 176. |

| Revision | Date | Description |
|-----------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3 | 07.03.2019 | <ol style="list-style-type: none">1. Added Rx Power detect monitoring and its status in AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT AE SB in page 1972. The PCR Test (b26) in AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB is not recommended to use (refer latest release note) in page 2763. Added a device API protocol limitation related to CRC in page 13 |

| Revision | Date | Description |
|-----------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.4 | 25.03.2019 | <ol style="list-style-type: none">1. Please refer Latest DFP release Note for all known issues and de-featured APIs.2. Added a note on destructive tests in AWR_MSS_LATENTFAULT_TEST_CONF_SB in page 2223. Updated PASS and FAIL bit definition/values in AWR_AE_DEV_MSSPOWERUPDONE_SB AND AWR_AE_MSS_BOOTERRORSTATUS_SB in page 242 and 250 respectively.4. Added new files type in AWR_DEV_FILE_DOWNLOAD_SB in page 231. |

| Revision | Date | Description |
|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.5 | 18.04.2019 | <ol style="list-style-type: none">1. Added new fields in AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB in page 172.2. Added reporting mode field in AWR_MONITOR_RX_GAIN_PHASE_CONF_SB in page 283.3. Added reporting mode field in AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB in page 297.4. Added new field NUM_OF_DUMMY_CHIRPS_AT_THE_END in AWR_FRAME_CONF_SET_SB in page ??.5. Added a note on minimum chirp duration in AWR_PROFILE_CONF_SET_SB in page 64.6. Added a new BSS fault type in AWR_AE_MSS_RFERROR_STATUS_SB in page 254.7. Added a note on RadarSS monitoring failure report redundancy for quiet mode in page 155.8. Updated AWR_MSSESMFAULT_STATUS_GET_SB to match with AWR_MSS_ESMFAULT_STATUS_SB in page 237.9. Updated description for error code 48 in page 26410. Updated API sequence description in page 25611. Added few notes in Rx saturation and signal image monitor configurations and sequence to follow in page 31912. De-featured PCR self test in AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB in page 276, refer latest release note for more info.13. De-featured PCR self test in AWR_MSS_LATENTFAULT_TEST_CONF_SB in page 222, refer latest release note for more info.14. De-featured LPF monitor in AWR_MONITOR_RX_IFSTAGE_CONF_SB in page 28815. De-featured Rampgen clock monitor in AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB in page 31716. De-featured Rampgen ECC self test monitor in AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB in page 27617. Added a note on synth frequency monitor configuration sequence in page 305 |

| Revision | Date | Description |
|-----------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.6 | 28.04.2019 | <ol style="list-style-type: none">1. Updated Rx gain computation formula in AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB in page 172.2. Added detailed description on calibration data and usage notes in AWR_CAL_DATA_RESTORE_SB in page 58 and in AWR_CAL_DATA_SAVE_SB in page 59.3. Added a note on Hw triggered mode and timing of Hw pulse in page 74.4. Added a note on Hw triggered mode and timing of Hw pulse in page 80.5. Added calibration restore and monitoring APIs sequence order in page 2566. Added a note on monitoring APIs configuration sequence in page 2767. Added a note on using TEST_MODE in RF digital monitoring API configuration in page 2768. Added a note on using analog fault injection API in page 3249. Disabled RX power detector internal signal monitor in AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB monitor and corresponding status bits made reserved in AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AE.10. Added PCR test status in AWR_AE_DEV_MSSPOWERUPDONE_SB AE in page 24211. Added new section about unsupported features and debug APIs in page 33112. Added notes about valid conditions to enable AWR_MSS_PERIODICTESTS_CONF_SB and AWR_MSS_LATENTFAULT_TEST_CONF_SB APIs in page 221 and 22213. Added a note about interference mitigation for boot calibrations in AWR_RF_INIT_SB API in page 64 |

| Revision | Date | Description |
|----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.7 | 10.07.2019 | <ol style="list-style-type: none">1. De-featured VIM interrupt self test (vim test) from RF digital boot and latent monitoring in 131, 244, 276.2. Updated test source BOUNDARY_MIN_VECx and BOUNDARY_MAX_VECx axes 142.3. De-featured VCO slope monitoring in AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB for 1st gen devices (please refer latest DFP 1.x release notes) 2004. De-featured VCO SYNTH_VCO_OPENLOOP fault injection in AWR_ANALOG_FAULT_INJECTION_CONF_SB for 1st gen devices (please refer latest DFP 1.x release notes) 3245. Updated Rx gain computation formula in AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB in page 172. |

NOTE1: Please refer latest mmWave device DFP release notes for all known issues and de-featured APIs

NOTE2: All reserved bytes/bits in configuration API sub blocks shall be programmed with value zero. The functionality of radar device is not guaranteed if reserved bytes are not zero.

NOTE3: All reserved bytes/bits in API message reports (ack or AE) sub blocks shall be masked off in application.

1 Introduction

1.1 Scope

The Scope of this document is to define interface control specifications for wide range of TI mmWave sensors. The wide range of highly integrated 77GHz and 60GHz CMOS TI mmwave sensors are tabulated in table 1.1. The devices integrate all of the RF and Analog functionality, including VCO, PLL, PA, LNA, Mixer and ADC for multiple TX/RX channels into a single chip. The AWR1243 is an RF transceiver device and it includes 4 receiver channels and 3 transmit channels in a single chip. The AWR1243 supports multi-chip cascading. The xWR1642, xWR1843 and IWR6843 are radar-on-a-chip devices, which includes 4 receive channels and 2 or 3 transmit channels and additionally an integrated DSP for radar signal processing.

All devices include a built-in BIST (Built-in Self-Test) processor, which is responsible to configure the RF/Analog and digital front-end in real-time, as well as to periodically schedule calibration and functional safety monitoring. This enables the mm-Wave front-end to be self-contained and capable of adapting itself to handle temperature and ageing effects, and to enable significant ease-of-use from an external host perspective.

This document contains the Interface Control Specification for communications on the serial interface (SPI) between the Radar device and the external host processor. The same protocol is used in all devices when the messages are sent to Radar Control subsystem (BIST subsystem) from the MCU subsystem (Master subsystem) and DSP subsystems.

Refer Link <http://www.ti.com/sensors/mmwave/overview.html> for more informations.

Table 1.1: TI CMOS radar devices

| Frequency Type | 60GHz RF Frequency (57GHz to 64GHz) | 77GHz RF Frequency (76GHz to 81GHz) |
|-----------------------------|------------------------------------------------|------------------------------------------------|
| TI Automotive Radar Devices | NA | AWR1243, AWR1642, AWR1443, AWR1843 |
| TI Industrial Radar Devices | IWR6843 | IWR1642, IWR1443, IWR1843 |

1.2 Intended Audience

The intended audience for this document is firmware, host software, and validation engineers needing to understand the format and contents of all communications between the Radar device

and the host processor.

2 TI mmWave Radar Sensor Communications Overview

2.1 Communication Link Description

The AWR1243 radar device communicates with the external host processor using the SPI interface. The radar device is configured and controlled from the external host processor by sending commands to AWR1243 device over SPI.

The xWR1642, xWR1843 and IWR6843 radar device is configured and controlled using the internal MCU (Master subsystem) and it communicates with an external ECU using the CAN interface. This document only talks about the communication protocol between radar device and external host processor using SPI in AWR1243. In xWR1642, xWR1843 and IWR6843 the same protocol is used to communicate between the BIST subsystem and Master subsystem.

2.2 Communication Link configuration

2.2.1 SPI

This interface is synchronous. The interface includes four signals (SPICCLK, SPICS, and Data In and Data Out) and supports clock rates up to 40 MHz. The AWR1243 radar device is always the SPI slave and the external host processor will be the SPI master.

2.2.2 Mailbox

This interface includes a SRAM and an interrupt line from Master subsystem to BIST subsystem. A reverse channel which includes a different SRAM and a different interrupt line from the BIST subsystem to Master subsystem is used for responses which originate from BIST subsystem.

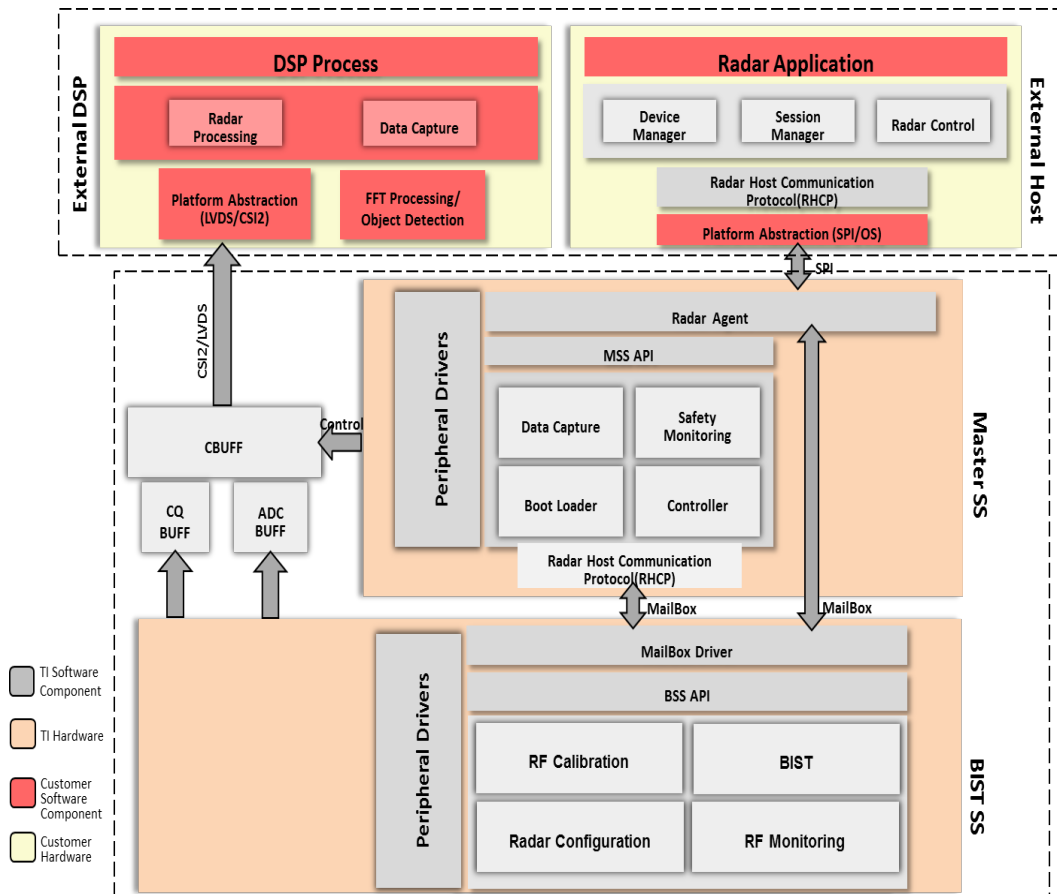


Figure 2.1: AWR12xx, AWR22xx Software Architecture

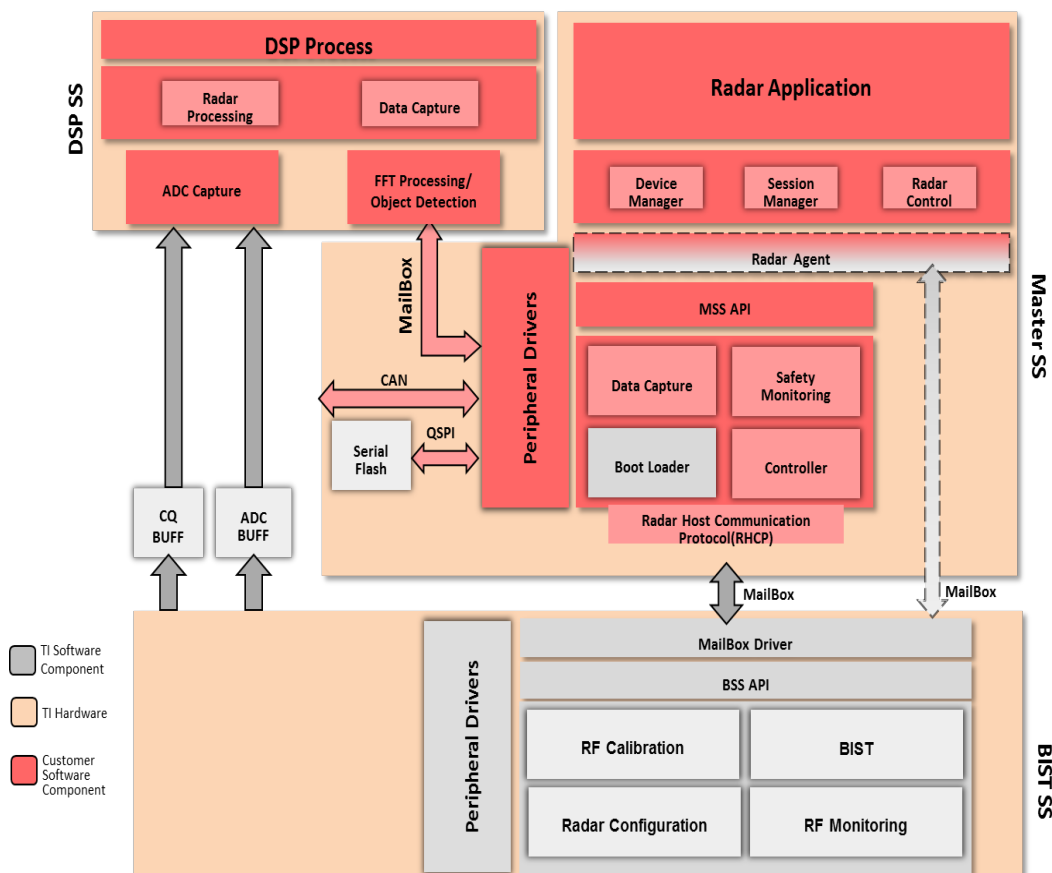


Figure 2.2: xWR16xx, xWR18xx and IWR68xx Software Architecture

2.3 Radar Message Structure

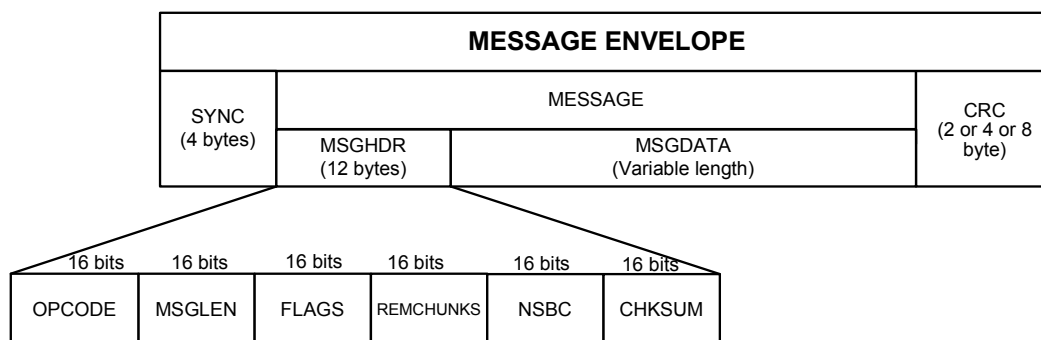


Figure 2.3: Radar Message Structure

Each message is sent in a message envelope, which starts with four special bytes called a sync pattern. Next, the message envelope contains the actual message and a CRC converted to a stream of bytes. Figure 2.3 defines the general form of radar messages. All communication messages between external host processor and the radar device will follow this message format. Each message consists of a 12-byte message header, variable length message data followed by a variable length CRC.

NOTE: The CRC and all the fields in the message headers and message data that are larger than one byte are sent in little-endian byte order i.e. the least significant byte is sent first.

A message envelope contains only one message.

2.3.1 SYNC

SYNC is a unique 4 byte pattern which marks the start of the message. It can take one of the following 3 values, in memory all the bytes are stored in little endian format (least significant byte first).

Table 2.1: Possible SYNC values and their usage

| SYNC word value | Description |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x43211234 | Messages from master to slave indicating a new command |
| 0x87655678 | Messages from external host to device indicating the host is now ready to receive a message from the device This pattern is defined as CNYS in this document. |
| 0xABCDDCBA | Messages from slave to master |

2.3.2 MSGHDR

Figure 2.4 defines the content of the message header. Each radar message must begin with this 12 byte message header in little endian format.

| OPCODE | LENGTH | FLAGS | REMCHUNKS | NSBC | CHKSUM |
|-----------|-----------|-----------|-----------|-----------|-----------|
| (16 bits) | (16 bits) | (16 bits) | (16 bits) | (16 bits) | (16 bits) |

Figure 2.4: Message Header Format

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|----|----|----|----|----|---|---|---|---|---------|---|-----------|---|---|---|
| MSGID | | | | | | | | | | MSGTYPE | | DIRECTION | | | |

Figure 2.5: OPCODE Format

OPCODE

The OPCODE is unique for a given message type. Figure 2.5 defines the OPCODE format.

| Bits | Field | Description |
|--------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [3:0] | DIRECTION | Direction of command 0000 Invalid 0001 Communication between Host to BSS 0010 Communication between BSS to Host 0011 Communication between Host to DSS 0100 Communication between DSS to Host 0101 Communication between Host to Master 0110 Communication between Master to Host 0111 Communication between BSS to Master 1000 Communication between Master to BSS 1001 Communication between BSS to DSS 1010 Communication between DSS to BSS 1011 Communication between Master to DSS 1100 Communication between DSS to Master 1101 RESERVED 1110 RESERVED 1111 RESERVED |
| [5:4] | MSGTYPE | Message type 00 COMMAND 01 RESPONSE (ACK or ERROR) 10 NACK 11 ASYNC |
| [15:6] | MSGID | Message ID 0x00 AWR_ERROR_MSG 0x01 RESERVED 0x02 RESERVED 0x03 RESERVED 0x04 AWR_RF_STATIC_CONF_SET_MSG 0x05 AWR_RF_STATIC_CONF_GET_MSG 0x06 AWR_RF_INIT_MSG 0x07 RESERVED 0x08 AWR_RF_DYNAMIC_CONF_SET_MSG 0x09 AWR_RF_DYNAMIC_CONF_GET_MSG 0x0A AWR_RF_FRAME_TRIG_MSG 0x0B RESERVED |

| | | | |
|--|--|-------|-------------------------------------------------------|
| | | 0x0C | AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG |
| | | 0x0D | RESERVED |
| | | 0x0E | AWR_RF_MONITORING_CONF_SET_MSG |
| | | 0x0F | RESERVED |
| | | 0x10 | RESERVED |
| | | 0x11 | AWR_RF_STATUS_GET_MSG |
| | | 0x12 | RESERVED |
| | | 0x13 | AWR_RF_MONITORING_REPORT_GET_MSG |
| | | 0x14 | RESERVED |
| | | 0x15 | RESERVED |
| | | 0x16 | AWR_RF_MISC_CONF_SET_MSG |
| | | 0x17 | AWR_RF_MISC_CONF_GET_MSG |
| | | 0x18 | RESERVED |
| | | 0x19 | RESERVED |
| | | 0x80 | AWR_RF_ASYNC_EVENT_MSG1 |
| | | 0x81 | AWR_RF_ASYNC_EVENT_MSG2 |
| | | 0x200 | AWR_DEV_RFPOWERUP_MSG |
| | | 0x201 | RESERVED |
| | | 0x202 | AWR_DEV_CONF_SET_MSG |
| | | 0x203 | AWR_DEV_CONF_GET_MSG |
| | | 0x204 | AWR_DEV_FILE_DOWNLOAD_MSG |
| | | 0x205 | RESERVED |
| | | 0x206 | AWR_DEV_FRAME_CONFIG_APPLY_MSG |
| | | 0x207 | AWR_DEV_STATUS_GET_MSG |
| | | 0x208 | RESERVED |
| | | 0x209 | RESERVED |
| | | 0x20A | RESERVED |
| | | 0x20B | RESERVED |
| | | 0x20C | RESERVED |
| | | 0x20D | RESERVED |
| | | 0x280 | AWR_DEV_ASYNC_EVENT_MSG |

LENGTH

The length field contains the length of the message in bytes including the message header, message data and CRC. Note that length field does not include the length of the sync field. The minimum length of the message is 12 bytes and maximum is 252 bytes. The message length minus CRC length must also be a multiple of 4 bytes.

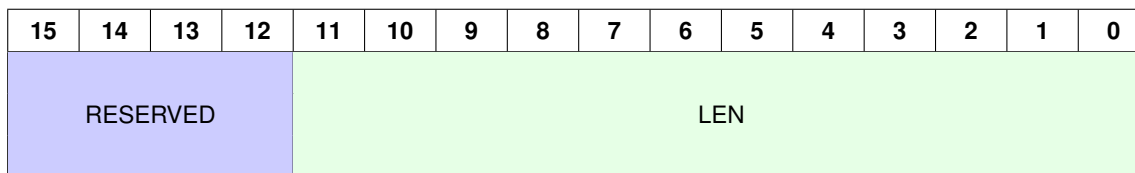


Figure 2.6: MSGLEN Format

Table 2.3: MSGLEN field descriptions

| Bits | Field | Description |
|---------|----------|----------------------------------------------------------------------------|
| [11:0] | LEN | Message length in bytes (It includes message header, message data and CRC) |
| [15:12] | RESERVED | Keep these bits as 0s |

FLAGS

The FLAGS is used to control the communication between the radar device and external host

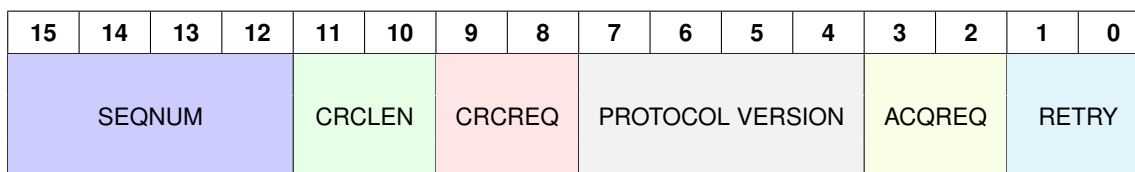


Figure 2.7: FLAGS Format

Table 2.4: FLAGS field description

| Bits | Field | Description |
|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [1:0] | RETRY | RETRY Value 00 New message 11 Retransmitted message 01 RESERVED 10 RESERVED |
| [3:2] | ACKREQ | Acknowledgement Request type 00 Acknowledgement is requested for the current message 11 Acknowledgement is not requested for the current message 01 RESERVED 10 RESERVED |

Continued on next page

Table 2.4 – continued from previous page

| | | |
|---------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [7:4] | PROTOCOL VERSION | Version number of the protocol that is used to communicate with the device (4 bits) |
| [9:8] | CRCREQ | CRC request type 00 CRC is appended to the message 11 CRC is not appended to the message 01 RESERVED 10 RESERVED |
| [11:10] | CRCLEN | Length of CRC appended to the message 00 16-bit CRC 01 32-bit CRC 10 64-bit CRC 11 RESERVED |
| [15:12] | SEQNUM | 4 bit sequence number of the message. Sequence number is reset to 0 after a device boot and each new message has the incremented sequence number. Whenever the same message is retransmitted, the sequence number is not incremented. |

NOTE: It is recommended to always append CRC to the message to prevent any message integrity issues

REMCHUNKS

If the message length is larger than 256 bytes, then it is split into multiple chunks of sizes less than 256 bytes. When this field is non-zero, this field indicates the number of remaining chunks that are to be expected.

NSBC

The message may contain several configuration sub blocks with structure as defined in Figure 2.3. The NSBC field indicates the total number sub blocks inside the message data.

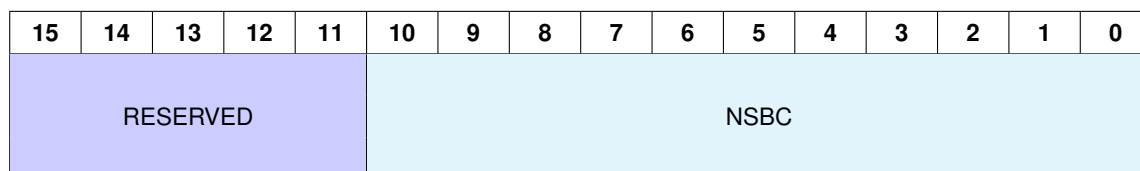

Figure 2.8: NSBC Format

Table 2.5: NSBC field description

| Bits | Field | Description |
|---------|----------|-------------------------------------|
| [10:0] | NSBC | Number of sub blocks in the message |
| [15:11] | RESERVED | Keep these bits as 0s |

CHKSUM

The message header is protected by a 16-bit checksum to enable the receiver to check the integrity of the message header. The checksum is computed on MSGHDR only (MSGID, MSGLEN, FLAGS, REMCHUNKS and NSBC fields). Note that SYNC field is not included in checksum calculation.

Checksum is 16-bit one's complement of the one's complement sum of all 16-bit words in the message header (Ref. <https://tools.ietf.org/html/rfc1071>).

For e.g., suppose the message header contents looks like this

Table 2.6: Checksum computation example

| Field | Value |
|-----------|--------|
| OPCODE | 0x0281 |
| MSGLEN | 0x0800 |
| FLAGS | 0x040C |
| REMCHUNKS | 0x0000 |
| NSBC | 0x0001 |
| CHKSUM | 0xF171 |

The receiver will compute the checksum as follows $0x0281 + 0x0800 = 0x0A81$.

Then, $0x0A81 + 0x040C = 0x0E8D$.

Then, $0x0E8D + 0x0000 = 0x0E8D$.

Then, $0x0E8D + 0x0001 = 0x0E8E$.

Ones complement of 0x0E8E is 0xF171 which matches with the received checksum.

2.3.3 MSGDATA

The message data contains the actual message specific data for the message. The message data contains sub blocks with structure as defined in Figure 2.9. More than one sub block can be appended in the MSGDATA to reduce the overall communication latency. The total number of sub blocks in MSGDATA is indicated in the NSBC field in the MSGHDR.

All data fields are aligned so that their offset in message is a multiple of the field size in bytes. For e.g. a 32 bit field in the message will be aligned to a 4 byte boundary and a 16 bit field will be aligned to a 2 byte boundary. This makes it possible to create a structure definition for the message for easy data access in most environments.

Any reserved (currently unused) fields in the messages should be always set as 0 when sent and ignored when received. This way those fields may be taken to use in later interface versions without modifying all old software.

All data structure in sub-blocks assumed to be in little endian format. For big endian Host system byte swap is required to match with defined protocol.

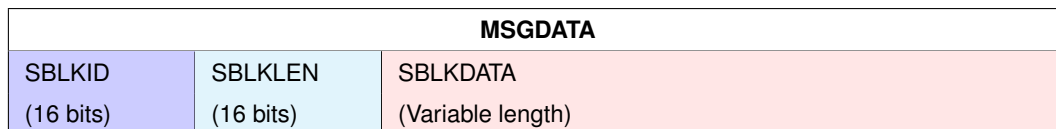


Figure 2.9: Message Sub block structure

SBLKID Unique ID of the sub block

SBLKLEN Length of the sub block in bytes

SBLKDATA Data corresponding to the sub block

2.3.4 CRC

This is a CRC which is appended to the message data to protect the integrity of the message. The CRC is computed on all the bytes in the MSGHDR and MSGDATA. Note that SYNC is not included in CRC calculation.

3 different types of CRCs can be used – 16 bit, 32 bit or 64 bit. The choice of the CRC type is indicated in the FLAGS field in the MSGHDR.

The polynomials used for each type of CRC calculation are

Table 2.7: CRC types and their polynomials

| CRC type | Polynomial | Remarks |
|----------|---------------------------------------------------------------------------------------------------------------|---------------------------|
| 16 bit | $x^{16} + x^{12} + x^5 + 1$ | 16-bit CRC-CCITT |
| 32 bit | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ | CRC-32 (used in Ethernet) |
| 64 bit | $x^{64} + x^4 + x^3 + x + 1$ | CRC-64-ISO (HDLC) |

NOTE: Device SPI protocol Limitation: The CRC length of the message or Async-event shall be multiple of 4 bytes to enable reliable retry recovery mechanism in case of any checksum failure in a message.

3 Message Processing

3.1 Communication protocol

When requested by the message transmitter, all correctly formatted radar messages are acknowledged by the receiver. This request for an acknowledgement is specified in FLAGS field of the MSGHDR (message header) field (see Section 2.3.2). A correctly formatted message is one that is formatted properly with a SYNC, MSGHDR, MSGDATA and CRC and that passes the CRC test when received. If an incorrectly formatted message is received, the radar device responds with a NACK message (MSGTYPE field in the MSGHDR set to NACK response). If a correctly formatted message is received, and after processing the message no errors are encountered, the radar device responds with an ACK response. In case of errors on a correctly formatted message, the radar device responds with an ERROR response.

The ACK response is a radar message which contains SYNC, MSGHDR, MSGDATA and CRC. In case the MSGTYPE was COMMAND_GET the MSGDATA for ACK response will contain the parameter values read by the radar device.

The NACK response is a radar message with only SYNC, MSGHDR and CRC. It does not contain MSGDATA.

For most commands the radar device prepares the acknowledgments and response packets immediately on reception. In certain cases, higher priority events in the system delay the execution of external communication function. The response time to command is a function of:

- Speed of the selected communication channel
- Although typical radar command/response occurs within a few hundreds of microseconds, it is recommended that host software wait up to 1 millisecond for response or acknowledgment before timing out on nonresponse.

The radar communication protocol is defined as follows

1. The host sends a message to the radar device requesting an acknowledgement. Host sets a timeout period of 1 ms for a response from the radar device.
2. The radar device checks the CHKSUM field for Message header validity and checks the MSGDATA field for correctness and also computes the CRC of the message and compares it with the received CRC.
 - If the computed CHKSUM does not match the received CHKSUM, the radar device does not send any response. The transmitter will timeout and eventually resend the command again with RETRY flag set

- If the CRC matches and all parameters are valid/correct, the radar device sends an ACK to the host
 - If the CRC matches, but any parameter in the message is invalid/incorrect, then the radar device sends an ERROR response to the host
 - If the CRC does not match, the radar device sends a NACK response to the host
3. On reception of the ACK, the host can send the next command to the radar device.
 4. If the host receives a NACK from the radar device within the timeout period, it sends the message again without the RETRY flag set.
 5. If the host does not receive any response from the radar device within the timeout period then it sends the same command with the RETRY flag set.

3.2 Communication Sequence

3.2.1 Command/Response Sequence (Host)

1. Host prepares the message as defined by protocol in Section 2.3
2. Host writes the message to the communication channel and starts Retry Timer (~1 ms)
3. Host then waits for HOST IRQ high Interrupt
 - a. If IRQ is received, go to Step 4
 - b. If Retry Time expires, Enable Retry Flag and go to Step 2
4. Host writes CNYS (SYNC word = 0x5678 0x8765) and Dummy bytes (0xFFFF 0xFFFF 0xFFFF 0xFFFF 0xFFFF 0xFFFF) on communication channel
5. Host waits for low on Host IRQ line
 - a. If Host IRQ line is low, go to Step 6
 - b. If Retry Time expires, Flag Error
6. Host reads the header from communication channel
7. Host checks the validity of header (verify checksum)
 - a. If header is valid, parse the header and go to Step 8
 - b. If header is invalid, ignore the header and reports to error to Application
8. Host reads the payload from communication channel
9. Host checks the validity of the message (verify CRC)
 - a. If message is valid, process the message
 - b. If message is invalid, go to Step 2 with new sequence number

3.2.2 Flow Diagram (Host) – Command/Response

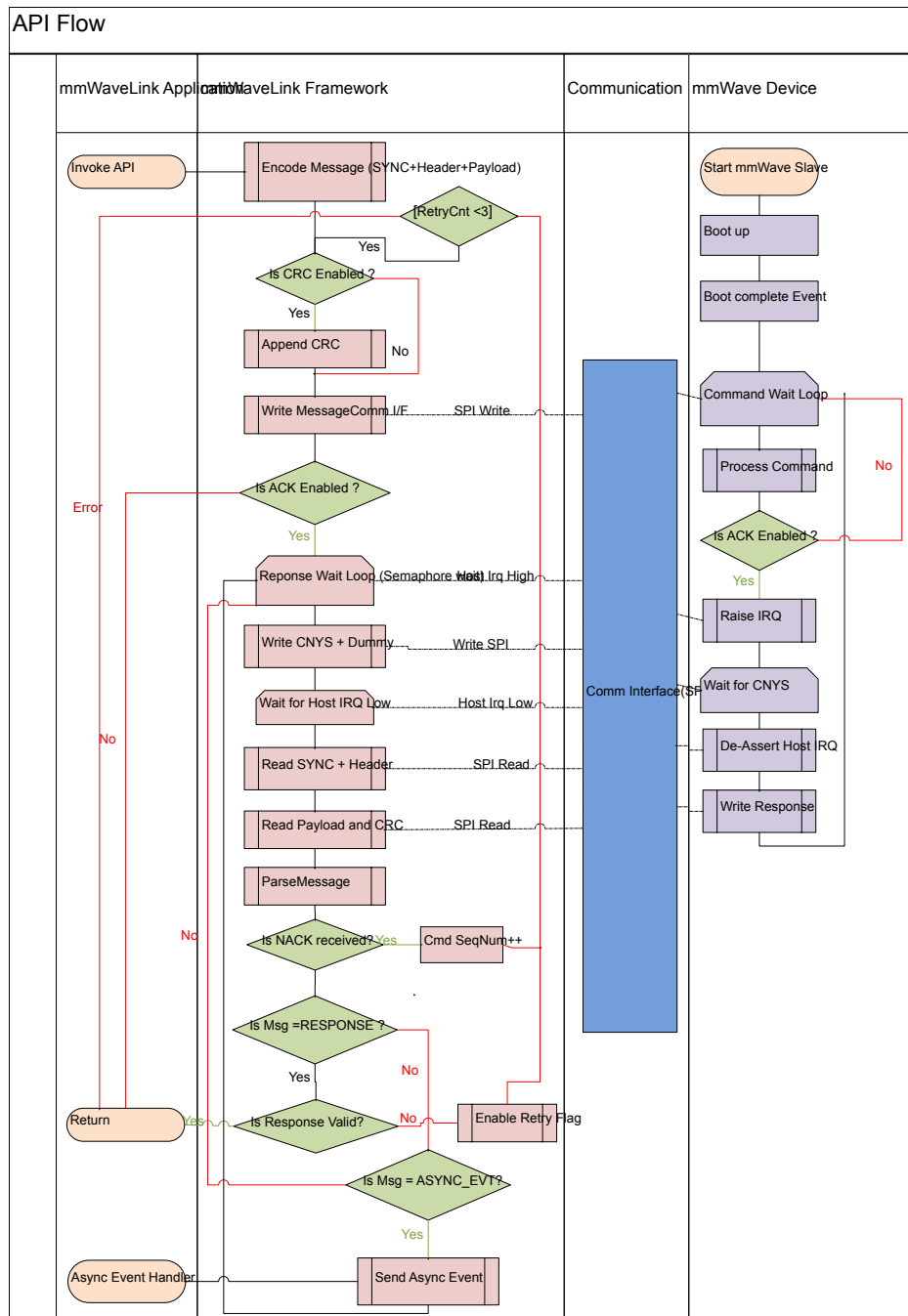


Figure 3.1: Flow Diagram (API)

3.2.3 Flow Diagram (Host) – Bootup/ Asynchronous Event

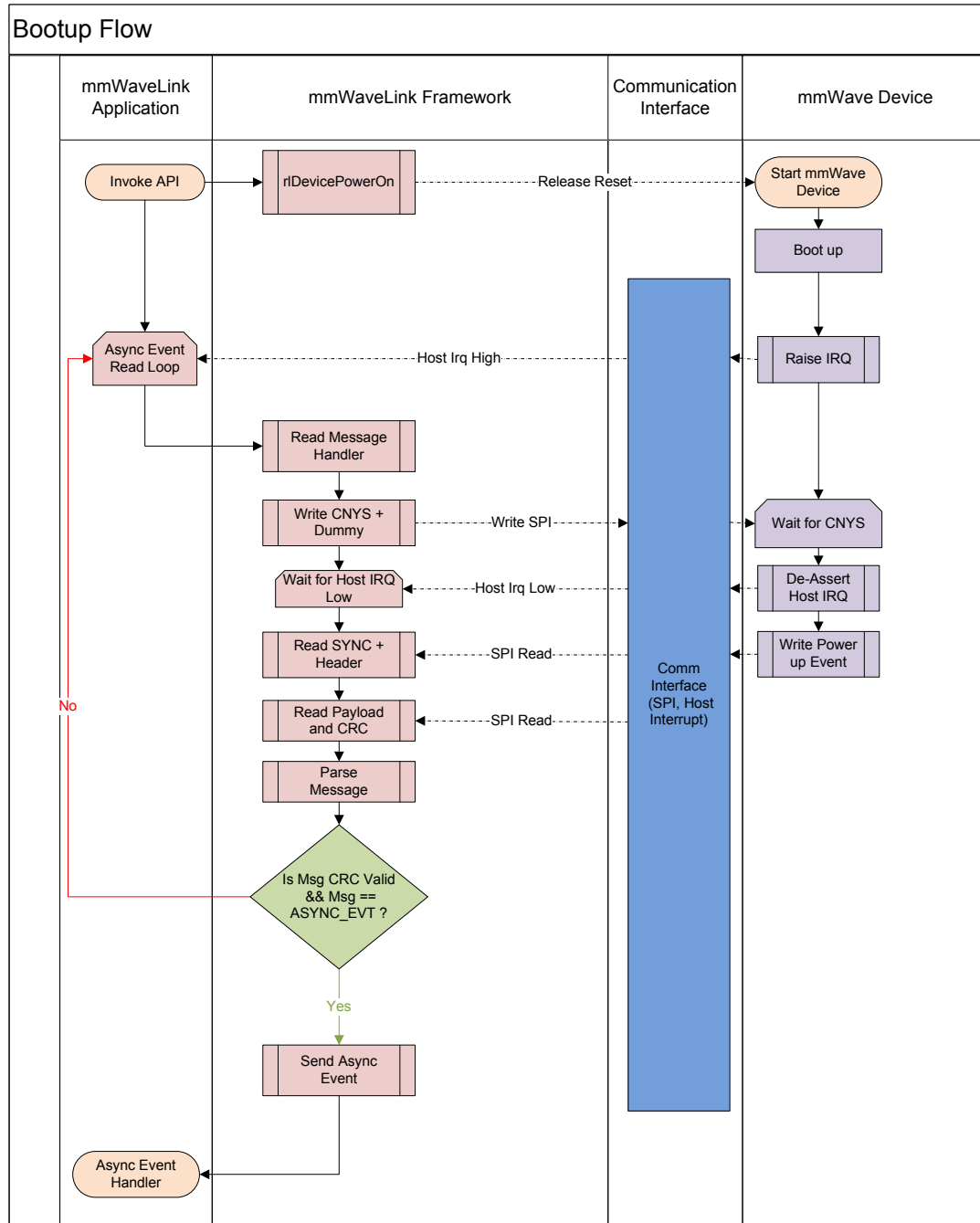


Figure 3.2: Flow Diagram (Asynchronous Events)

3.2.4 SPI Message Sequence – Command/Response

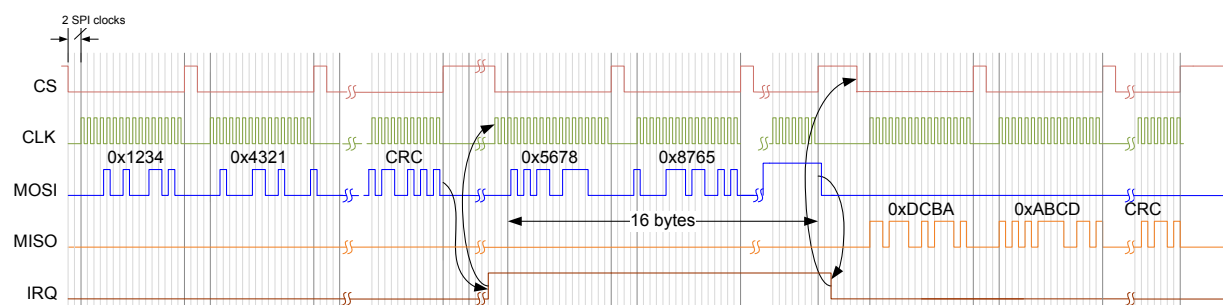


Figure 3.3: SPI Message Sequence

NOTE:

1. Host should ensure that there is a delay of at least 2 SPI clocks between CS going low and start of SPI clock
2. Host should ensure that CS is toggled for every 16 bits of transfer via SPI
3. There should be a delay of at least 2 SPI Clocks between consecutive CS
4. SPI needs to be operated at Mode 0 (Phase 0, Polarity 0)
5. SPI word length should be 16 bit (Half word)

4 Radar Interface Messages Descriptions

This section describes all the radar interface messages that are used in communication with the radar transceiver.

4.1 Summary of all messages and their associated sub-blocks

Table 4.1: Summary of all Radar messages and their associated sub blocks

| Radar Messages | Associated sub-blocks |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AWR_ACK_MSG | NA |
| AWR_NACK_MSG | NA |
| AWR_ERROR_MSG | AWR_RESP_ERROR_SB |
| AWR_RF_STATIC_CONF_SET_MSG | AWR_CHAN_CONF_SET_SB AWR_ADCOUT_CONF_SET_SB AWR_LOWPOWERMODE_CONF_SET_SB AWR_DYNAMICPOWERSAVE_CONF_SET_SB AWR_HIGHSPEEDINTFCLK_CONF_SET_SB AWR_RF_DEVICE_CFG_SB AWR_RF_RADAR_MISC_CTL_SB AWR_CAL_MON_FREQUENCY_LIMITS_SB AWR_RF_INIT_CALIBRATION_CONF_SB AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB AWR_CAL_DATA_RESTORE_SB AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB |
| AWR_RF_STATIC_CONF_GET_MSG | AWR_CAL_DATA_SAVE_SB AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB |
| AWR_RF_INIT_MSG | AWR_RF_INIT_SB |
| AWR_RF_DYNAMIC_CONF_SET_MSG | AWR_PROFILE_CONF_SET_SB AWR_CHIRP_CONF_SET_SB AWR_FRAME_CONF_SET_SB AWR_CONT_STREAMING_MODE_CONF_SET_SB AWR_CONT_STREAMING_MODE_EN_SB |

Continued on next page

Table 4.1 – continued from previous page

| Radar Messages | Associated sub-blocks |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | AWR_ADVANCED_FRAME_CONF_SB AWR_PERCHIRPPHASESHIFT_CONF_SB AWR_PROG_FILT_COEFF_RAM_SET_SB AWR_PROG_FILT_CONF_SET_SB AWR_CALIB_MON_TIME_UNIT_CONF_SB AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB AWR_INTER_RX_GAIN_PHASE_CONTROL_SB AWR_RX_GAIN_TEMPLUT_SET_SB AWR_TX_GAIN_TEMPLUT_SET_SB AWR_LOOPBACK_BURST_CONF_SET_SB AWR_DYN_CHIRP_CONF_SET_SB AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SB AWR_DYN_CHIRP_ENABLE_SB AWR_INTERCHIRP_BLOCKCONTROLS_SB AWR_SUBFRAME_START_CONF_SB |
| AWR_RF_DYNAMIC_CONF_GET_MSG | AWR_PROFILE_CONF_GET_SB AWR_CHIRP_CONF_GET_SB AWR_FRAME_CONF_GET_SB AWR_ADVANCED_FRAME_CONF_GET_SB AWR_RX_GAIN_TEMPLUT_GET_SB AWR_TX_GAIN_TEMPLUT_GET_SB |
| AWR_RF_FRAME_TRIG_MSG | AWR_FRAMESTARTSTOP_CONF_SB |
| AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG | AWR_BPM_COMMON_CONF_SET_SB AWR_BPM_CHIRP_CONF_SET_SB |
| AWR_RF_MONITORING_CONF_SET_MSG | AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB AWR_MONITOR_ANALOG_ENABLES_CONF_SB AWR_MONITOR_TEMPERATURE_CONF_SB AWR_MONITOR_RX_GAIN_PHASE_CONF_SB AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB AWR_MONITOR_RX_IFSTAGE_CONF_SB AWR_MONITOR_TX0_POWER_CONF_SB AWR_MONITOR_TX1_POWER_CONF_SB AWR_MONITOR_TX2_POWER_CONF_SB |

Continued on next page

Table 4.1 – continued from previous page

| Radar Messages | Associated sub-blocks |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | AWR_MONITOR_TX0_BALLBREAK_CONF_SB AWR_MONITOR_TX1_BALLBREAK_CONF_SB AWR_MONITOR_TX2_BALLBREAK_CONF_SB AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB AWR_MONITOR_TX0_BPM_CONF_SB AWR_MONITOR_TX1_BPM_CONF_SB AWR_MONITOR_TX2_BPM_CONF_SB AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_PLL_CONTROL_VOLTAGE_SIGNALS_CONF_SB AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB AWR_MONITOR_RX_MIXER_IN_POWER_CONF_SB AWR_ANALOG_FAULT_INJECTION_CONF_SB |
| AWR_RF_MONITORING_REPORT_GET_MSG | AWR_RF_DFE_STATISTICS_REPORT_GET_SB |
| AWR_RF_STATUS_GET_MSG | AWR_RF_VERSION_GET_SB AWR_RF_CPUFAULT_STATUS_GET_SB AWR_RF_ESMFAULT_STATUS_GET_SB AWR_RF_DIEID_GET_SB AWR_RF_BOOTUPBIST_STATUS_GET_SB |
| | AWR_RF_TEST_SOURCE_CONFIG_SET_SB AWR_RF_TEST_SOURCE_ENABLE_SET_SB |

Continued on next page

AWR_RF_MISC_CONF_SET_MSG

Table 4.1 – continued from previous page

| Radar Messages | Associated sub-blocks |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | AWR_RF_LDO_BYPASS_SB AWR_RF_PALOOPBACK_CFG_SB AWR_RF_PSLOOPBACK_CFG_SB AWR_RF_IFLOOPBACK_CFG_SB AWR_RF_GPADC_CFG_SET_SB |
| AWR_RF_MISC_CONF_GET_MSG | AWR_RF_TEMPERATURE_GET_SB |
| AWR_RF_ASYNC_EVENT_MSG1 | AWR_AE_RF_CPUFAULT_SB AWR_AE_RF_ESMFAULT_SB AWR_AE_RF_INITCALIBSTATUS_SB AWR_AE_RF_FRAME_TRIGGER_RDY_SB AWR_AE_RF_GPADC_RESULT_DATA_SB AWR_FRAME_END_AE_SB AWR_ANALOGFAULT_AE_SB AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB AWR_RUN_TIME_CALIBRATION_SUMMARY_REPORT_AE_SB AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB AWR_MONITOR_REPORT_HEADER_AE_SB AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB AWR_MONITOR_TEMPERATURE_REPORT_AE_SB AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB AWR_MONITOR_TX0_POWER_REPORT_AE_SB AWR_MONITOR_TX1_POWER_REPORT_AE_SB AWR_MONITOR_TX2_POWER_REPORT_AE_SB AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB |
| AWR_RF_ASYNC_EVENT_MSG2 | AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_REPORT_AE_SB AWR_MONITOR_TX0_BPM_REPORT_AE_SB AWR_MONITOR_TX1_BPM_REPORT_AE_SB AWR_MONITOR_TX2_BPM_REPORT_AE_SB AWR_MONITOR_SYNTHESIZER_FREQUENCY_REPORT_AE_SB |

Continued on next page

Table 4.1 – continued from previous page

| Radar Messages | Associated sub-blocks |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | AWR_MONITOR_EXTERNAL_ANALOG_SIGNALSREPORT_AE_SB AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB |
| AWR_DEV_RFPOWERUP_MSG | AWR_DEV_RFPOWERUP_SB |
| AWR_DEV_CONF_SET_MSG | AWR_DEV_MCUCLOCK_CONF_SET_SB AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB AWR_DEV_RX_DATA_PATH_CONF_SET_SB AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB AWR_DEV_RX_DATA_PATH_CLK_SET_SB AWR_DEV_LVDS_CFG_SET_SB AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB AWR_DEV_CSI2_CFG_SET_SB AWR_DEV_PMICCLOCK_CONF_SET_SB AWR_MSS_LATENTFAULT_TEST_CONF_SB AWR_MSS_PERIODICTESTS_CONF_SB AWR_DEV_TESTPATTERN_GEN_SET_SB AWR_DEV_CONFIGURATION_SET_SB |
| AWR_DEV_CONF_GET_MSG | AWR_DEV_MCUCLOCK_GET_SB AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB AWR_DEV_RX_DATA_PATH_CONF_GET_SB AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB AWR_DEV_RX_DATA_PATH_CLK_GET_SB AWR_DEV_LVDS_CFG_GET_SB |

Continued on next page

Table 4.1 – continued from previous page

| Radar Messages | Associated sub-blocks |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | AWR_DEV_RX_CONTSTREAMING_MODE_CONF_GET_SB AWR_DEV_CSI2_CFG_GET_SB AWR_DEV_PMICCLOCK_CONF_GET_SB AWR_MSS_LATENTFAULT_TEST_CONF_GET_SB AWR_MSS_PERIODICCONF_GET_SB AWR_DEV_TESTPATTERN_GEN_GET_SB |
| AWR_DEV_FILE_DOWNLOAD_MSG | AWR_DEV_FILE_DOWNLOAD_SB |
| AWR_DEV_FRAME_CONFIG_APPLY_MSG | AWR_DEV_FRAME_CONFIG_APPLY_SB AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB |
| AWR_DEV_STATUS_GET_MSG | AWR_MSSVERSION_GET_SB AWR_MSSCPUFAULT_STATUS_GET_SB AWR_MSSSEMFault_STATUS_GET_SB |
| AWR_DEV_ASYNC_EVENT_MSG | AWR_AE_DEV_MSSPOWERUPDONE_SB AWR_AE_DEV_RFPOWERUPDONE_SB AWR_AE_MSS_CPUFAULT_SB AWR_AE_MSS_ESMFAULT_SB AWR_AE_MSS_BOOTERRORSTATUS_SB AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB AWR_AE_MSS_PERIODICTEST_STATUS_SB AWR_AE_MSS_RFERROR_STATUS_SB AWR_AE_MSS_VMON_ERRORSTATUS_SB AWR_AE_MSS_ADC_DATA_SB |

4.2 AWR_ACK_MSG

The AWR_ACK_MSG is sent by the radar transceiver on a successful reception of a command after its CRC check.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------------------------------|
| SYNC | 4 | Value = 0xABCDDCBA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>01</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>Same as MSGID in the command</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 01 | b15:6 | MSGID | Same as MSGID in the command |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 01 | | | | | | | | | | | | |
| b15:6 | MSGID | Same as MSGID in the command | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLen field in FLAGS | | | | | | | | | | | | |

4.3 AWR_NACK_MSG

The AWR_NACK_MSG is sent by the radar transceiver if the CRC check of the command fails.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------------------------------|
| SYNC | 4 | Value = 0xABCDDCBA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>10</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>Same as MSGID in the command</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 10 | b15:6 | MSGID | Same as MSGID in the command |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 10 | | | | | | | | | | | | |
| b15:6 | MSGID | Same as MSGID in the command | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLen field in FLAGS | | | | | | | | | | | | |

4.4 AWR_ERROR_MSG

The AWR_RF_ERROR_MSG is sent by the radar transceiver on finding errors in the command send by host.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0xABCDDCBA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>01</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x00</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 01 | b15:6 | MSGID | 0x00 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 01 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x00 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_RESP_ERROR_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLEN field in FLAGS | | | | | | | | | | | | |

4.5 AWR_RF_STATIC_CONF_SET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x04</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x04 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x04 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_CHAN_CONF_SET_SB AWR_ADCOUT_CONF_SET_SB | | | | | | | | | | | | |

| | | |
|-----|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_LOWPOWERMODE_CONF_SET_SB AWR_DYNAMICPOWERSAVE_CONF_SET_SB AWR_HIGHSPEEDINTFCLK_CONF_SET_SB AWR_RF_DEVICE_CFG_SB AWR_RF_RADAR_MISC_CTL_SB AWR_CAL_MON_FREQUENCY_LIMITS_SB AWR_RF_INIT_CALIBRATION_CONF_SB AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB AWR_CAL_DATA_RESTORE_SB AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.6 AWR_RF_STATIC_CONF_GET_MSG

| Field Name | Number of bytes | Description |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| SYNC | 4 | Value = 0x43211234 |
| OPCODE | 2 | Bits Variable name Value b3:0 DIRECTION See Table 2.2 b5:4 MSGTYPE 00 b15:6 MSGID 0x05 |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) |
| FLAGS | 2 | See Section 2.3.2 |
| REMCHUNKS | 2 | Value = 0 |
| NSBC | 2 | Number of sub blocks contained in the message |
| CHKSUM | 2 | See Section 2.3.2 |
| MSGDATA | Variable | Supported sub blocks AWR_CAL_DATA_SAVE_SB AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.7 AWR_RF_INIT_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x06</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x06 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x06 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_RF_INIT_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLEN field in FLAGS | | | | | | | | | | | | |

4.8 AWR_RF_DYNAMIC_CONF_SET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x08</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x08 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x08 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_PROFILE_CONF_SET_SB AWR_CHIRP_CONF_SET_SB | | | | | | | | | | | | |

| | | |
|-----|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_FRAME_CONF_SET_SB AWR_CONT_STREAMING_MODE_CONF_SET_SB AWR_CONT_STREAMING_MODE_EN_SB AWR_ADVANCED_FRAME_CONF_SB AWR_PERCHIRPPHASESHIFT_CONF_SB AWR_PROG_FILTER_COEFF_RAM_SET_SB AWR_PROG_FILTER_CONF_SET_SB AWR_CALIB_MON_TIME_UNIT_CONF_SB AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB AWR_INTER_RX_GAIN_PHASE_CONTROL_SB AWR_RX_GAIN_TEMPLUT_SET_SB AWR_TX_GAIN_TEMPLUT_SET_SB AWR_LOOPBACK_BURST_CONF_SET_SB AWR_DYN_CHIRP_CONF_SET_SB AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SB AWR_DYN_CHIRP_ENABLE_SB AWR_INTERCHIRP_BLOCKCONTROLS_SB AWR_SUBFRAME_START_CONF_SB |
| CRC | Variable | Based on CRCLEN field in FLAGS |

4.9 AWR_RF_DYNAMIC_CONF_GET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x09</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x09 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x09 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |

| | | |
|---------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHKSUM | 2 | See Section 2.3.2 |
| MSGDATA | Variable | Supported sub blocks AWR_PROFILE_CONF_GET_SB AWR_CHIRP_CONF_GET_SB AWR_FRAME_CONF_GET_SB AWR_ADVANCED_FRAME_CONF_GET_SB AWR_RX_GAIN_TEMPLUT_GET_SB AWR_TX_GAIN_TEMPLUT_GET_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.10 AWR_RF_FRAME_TRIG_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|-------------------------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x0A</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x0A |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x0A | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_FRAMESTARTSTOP_CONF_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLLEN field in FLAGS | | | | | | | | | | | | |

4.11 AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x0C</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x0C |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x0C | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_BPM_COMMON_CONF_SET_SB AWR_BPM_CHIRP_CONF_SET_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRLEN field in FLAGS | | | | | | | | | | | | |

4.12 AWR_RF_MONITORING_CONF_SET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x0E</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x0E |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x0E | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB | | | | | | | | | | | | |

| | | |
|-----|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_MONITOR_ANALOG_ENABLES_CONF_SB AWR_MONITOR_TEMPERATURE_SONF_SB AWR_MONITOR_RX_GAIN_PHASE_CONF_SB AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB AWR_MONITOR_RX_IFSTAGE_CONF_SB AWR_MONITOR_TX0_POWER_CONF_SB AWR_MONITOR_TX1_POWER_CONF_SB AWR_MONITOR_TX2_POWER_CONF_SB AWR_MONITOR_TX0_BALLBREAK_CONF_SB AWR_MONITOR_TX1_BALLBREAK_CONF_SB AWR_MONITOR_TX2_BALLBREAK_CONF_SB AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB AWR_MONITOR_TX0_BPM_CONF_SB AWR_MONITOR_TX1_BPM_CONF_SB AWR_MONITOR_TX2_BPM_CONF_SB AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB AWR_MONITOR_PLL_CONTROL_VOLTAGE_SIGNALS_CONF_SB AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB AWR_MONITOR_RX_MIXER_IN_POWER_CONF_SB AWR_ANALOG_FAULT_INJECTION_CONF_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.13 AWR_RF_STATUS_GET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x11</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x11 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x11 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_RF_VERSION_GET_SB AWR_RF_CPUFAULT_STATUS_GET_SB AWR_RF_ESMFAULT_STATUS_GET_SB AWR_RF_DIEID_GET_SB AWR_RF_BOOTUPBIST_STATUS_GET_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRLEN field in FLAGS | | | | | | | | | | | | |

4.14 AWR_RF_MONITORING_REPORT_GET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x13</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x13 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x13 | | | | | | | | | | | | |

| | | |
|-----------|----------|-----------------------------------------------------------------------------|
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) |
| FLAGS | 2 | See Section 2.3.2 |
| REMCHUNKS | 2 | Value = 0 |
| NSBC | 2 | Number of sub blocks contained in the message |
| CHKSUM | 2 | See Section 2.3.2 |
| MSGDATA | Variable | Supported sub blocks AWR_RF_DFE_STATISTICS_REPORT_GET_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.15 AWR_RF_MISC_CONF_SET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|-------------------------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x16</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x16 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x16 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_RF_TEST_SOURCE_CONFIG_SET_SB AWR_RF_TEST_SOURCE_ENABLE_SET_SB AWR_RF_LDO_BYPASS_SB AWR_RF_PALLOOPBACK_CFG_SB AWR_RF_PSLOOPBACK_CFG_SB AWR_RF_IFLOOPBACK_CFG_SB AWR_RF_GPADC_CFG_SET_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLLEN field in FLAGS | | | | | | | | | | | | |

4.16 AWR_RF_MISC_CONF_GET_MSG

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="1"> <thead> <tr> <th>Bits</th> <th>Variable name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x17</td> </tr> </tbody> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x17 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x17 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_RF_TEMPERATURE_GET_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLEN field in FLAGS | | | | | | | | | | | | |

4.17 AWR_RF_ASYNC_EVENT_MSG1

The AWR_RF_ASYNC_EVENT_MSG1 is sent by the radar transceiver to the host. This message indicates that specific events have occurred within the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0xABCDDBCA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="1"> <thead> <tr> <th>Bits</th> <th>Variable name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>11</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x80</td> </tr> </tbody> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 11 | b15:6 | MSGID | 0x80 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 11 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x80 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks | | | | | | | | | | | | |

| | | |
|-----|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_AE_RF_CPUFAULT_SB AWR_AE_RF_ESMFAULT_SB AWR_AE_RF_INITCALIBSTATUS_SB AWR_AE_RF_FRAME_TRIGGER_RDY_SB AWR_AE_RF_GPADC_RESULT_DATA_SB AWR_FRAME_END_AE_SB AWR_ANALOGFAULT_AE_SB AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB AWR_RUN_TIME_CALIBRATION_SUMMARY_REPORT_AE_SB AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB AWR_MONITOR_REPORT_HEADER_AE_SB AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB AWR_MONITOR_TEMPERATURE_REPORT_AE_SB AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB AWR_MONITOR_TX0_POWER_REPORT_AE_SB AWR_MONITOR_TX1_POWER_REPORT_AE_SB AWR_MONITOR_TX2_POWER_REPORT_AE_SB AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB |
| CRC | Variable | Based on CRCLEN field in FLAGS |

4.18 AWR_RF_ASYNC_EVENT_MSG2

The AWR_RF_ASYNC_EVENT_MSG2 is sent by the radar transceiver to the host. This message indicates that specific events have occurred within the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|------|
| SYNC | 4 | Value = 0xABCDDBCA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>11</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x81</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 11 | b15:6 | MSGID | 0x81 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 11 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x81 | | | | | | | | | | | | |

| | | |
|-----------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) |
| FLAGS | 2 | See Section 2.3.2 |
| REMCHUNKS | 2 | Value = 0 |
| NSBC | 2 | Number of sub blocks contained in the message |
| CHKSUM | 2 | See Section 2.3.2 |
| MSGDATA | Variable | Supported sub blocks AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_REPORT_AE_SB AWR_MONITOR_TX0_BPM_REPORT_AE_SB AWR_MONITOR_TX1_BPM_REPORT_AE_SB AWR_MONITOR_TX2_BPM_REPORT_AE_SB AWR_MONITOR_SYNTHESIZER_FREQUENCY_REPORT_AE_SB AWR_MONITOR_EXTERNAL_ANALOG_SIGNALSREPORT_AE_SB AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB |
| CRC | Variable | Based on CRCLEN field in FLAGS |

4.19 AWR_DEV_RFPOWERUP_MSG

The AWR_DEV_RFPOWERUP_MSG is sent by the host to the MSS. This message indicates that BSS can now be powered up.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0xABCDDBCA | | | | | | | | | | | | |
| OPCODE | 2 | <table border="1"> <thead> <tr> <th>Bits</th> <th>Variable name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x200</td> </tr> </tbody> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x200 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x200 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_DEV_RFPOWERUP_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLEN field in FLAGS | | | | | | | | | | | | |

4.20 AWR_DEV_CONF_SET_MSG

The AWR_DEV_CONF_SET_MSG is sent by the host to the radar transceiver. This message indicates that specific events have occurred within the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="1"> <thead> <tr> <th>Bits</th> <th>Variable name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x202</td> </tr> </tbody> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x202 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x202 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_DEV_MCUCLOCK_CONF_SET_SB | | | | | | | | | | | | |

| | | |
|-----|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB AWR_DEV_RX_DATA_PATH_CONF_SET_SB AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB AWR_DEV_RX_DATA_PATH_CLK_SET_SB AWR_DEV_LVDS_CFG_SET_SB AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB AWR_DEV_CSI2_CFG_SET_SB AWR_DEV_PMICCLOCK_CONF_SET_SB AWR_MSS_LATENTFAULT_TEST_CONF_SB AWR_MSS_PERIODICTESTS_CONF_SB AWR_DEV_TESTPATTERN_GEN_SET_SB AWR_DEV_CONFIGURATION_SET_SB |
| CRC | Variable | Based on CRCLEN field in FLAGS |

4.21 AWR_DEV_CONF_GET_MSG

The AWR_DEV_CONF_GET_MSG is sent by the host to the radar transceiver to read back the configuration values.

| Field Name | Number of bytes | Description |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYNC | 4 | Value = 0x43211234 |
| OPCODE | 2 | Bits Variable name Value b3:0 DIRECTION See Table 2.2 b5:4 MSGTYPE 00 b15:6 MSGID 0x203 |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) |
| FLAGS | 2 | See Section 2.3.2 |
| REMCHUNKS | 2 | Value = 0 |
| NSBC | 2 | Number of sub blocks contained in the message |
| CHKSUM | 2 | See Section 2.3.2 |
| MSGDATA | Variable | Supported sub blocks AWR_DEV_MCUCLOCK_GET_SB AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB AWR_DEV_RX_DATA_PATH_CONF_GET_SB |

| | | |
|-----|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB AWR_DEV_RX_DATA_PATH_CLK_GET_SB AWR_DEV_LVDS_CFG_GET_SB AWR_DEV_RX_CONTSTREAMING_MODE_CONF_GET_SB AWR_DEV_CSI2_CFG_GET_SB AWR_DEV_PMICCLOCK_CONF_GET_SB AWR_MSS_LATENTFAULT_TEST_CONF_GET_SB AWR_MSS_PERIODICCONF_GET_SB AWR_DEV_TESTPATTERN_GEN_GET_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.22 AWR_DEV_FILE_DOWNLOAD_MSG

The AWR_DEV_FILE_DOWNLOAD_MSG is sent by the host to MSS. This message sends a file to be written into the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x204</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x204 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x204 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_DEV_FILE_DOWNLOAD_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLLEN field in FLAGS | | | | | | | | | | | | |

4.23 AWR_DEV_FRAME_CONFIG_APPLY_MSG

The AWR_DEV_FRAME_CONFIG_APPLY_MSG is sent by the host to MSS. This message indicates to MSS to apply all the regular framing mode configurations related to ADC buffer and CBUFF.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x206</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x206 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x206 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_DEV_FRAME_CONFIG_APPLY_SB AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLLEN field in FLAGS | | | | | | | | | | | | |

4.24 AWR_DEV_STATUS_GET_MSG

The AWR_DEV_STATUS_GET_MSG is sent by the host to MSS to get some status information from the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>00</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x207</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 00 | b15:6 | MSGID | 0x207 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 00 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x207 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks | | | | | | | | | | | | |

| | | |
|-----|----------|------------------------------------------------------------------------------------------------------------------------------------------|
| | | AWR_MSSVERSION.GET_SB AWR_MSSCPUFAULT_STATUS.GET_SB AWR_MSSSESMFAULT_STATUS.GET_SB |
| CRC | Variable | Based on CRCLLEN field in FLAGS |

4.25 AWR_DEV_ASYNC_EVENT_MSG

The AWR_DEV_ASYNC_EVENT_MSG is sent by the radar transceiver to the host. This message indicates that specific events have occurred within the device.

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------|-------|------|-----------|---------------|------|---------|----|-------|-------|-------|
| SYNC | 4 | Value = 0x43211234 | | | | | | | | | | | | |
| OPCODE | 2 | <table border="0"> <tr> <td>Bits</td> <td>Variable name</td> <td>Value</td> </tr> <tr> <td>b3:0</td> <td>DIRECTION</td> <td>See Table 2.2</td> </tr> <tr> <td>b5:4</td> <td>MSGTYPE</td> <td>11</td> </tr> <tr> <td>b15:6</td> <td>MSGID</td> <td>0x280</td> </tr> </table> | Bits | Variable name | Value | b3:0 | DIRECTION | See Table 2.2 | b5:4 | MSGTYPE | 11 | b15:6 | MSGID | 0x280 |
| Bits | Variable name | Value | | | | | | | | | | | | |
| b3:0 | DIRECTION | See Table 2.2 | | | | | | | | | | | | |
| b5:4 | MSGTYPE | 11 | | | | | | | | | | | | |
| b15:6 | MSGID | 0x280 | | | | | | | | | | | | |
| MSGLEN | 2 | Length of the message in bytes (do not include sync length) | | | | | | | | | | | | |
| FLAGS | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| REMCHUNKS | 2 | Value = 0 | | | | | | | | | | | | |
| NSBC | 2 | Number of sub blocks contained in the message | | | | | | | | | | | | |
| CHKSUM | 2 | See Section 2.3.2 | | | | | | | | | | | | |
| MSGDATA | Variable | Supported sub blocks AWR_AE_DEV_MSSPOWERUPDONE_SB AWR_AE_DEV_RFPOWERUPDONE_SB AWR_AE_MSS_CPUFAULT_SB AWR_AE_MSS_ESMFAULT_SB AWR_AE_MSS_BOOTERRORSTATUS_SB AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB AWR_AE_MSS_PERIODICTEST_STATUS_SB AWR_AE_MSS_RFERROR_STATUS_SB AWR_AE_MSS_VMON_ERRORSTATUS_SB AWR_AE_MSS_ADC_DATA_SB | | | | | | | | | | | | |
| CRC | Variable | Based on CRCLLEN field in FLAGS | | | | | | | | | | | | |

5 Radar Functional APIs

This section describes all the radar interface sub blocks that are used in messages for communicating with the radar transceiver. Some of the sub blocks are status responses from the radar device. All the API sub-blocks defined in this document are applicable to all mmWave Radar Sensors unless it is specified in the sub-block.

5.1 Sub block related to AWR_ERROR_MSG

5.1.1 Sub block 0x0000 – AWR_RESP_ERROR_SB

This sub block contains the error response for an API command. Table 5.1 describes the contents of this sub block.

Table 5.1: AWR_RESP_ERROR_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0000 |
| SBLKLEN | 2 | Value = 8 |
| API_RESP | 2 | 0x0001 ERROR_CMD: Incorrect MSGID 0x0002 ERROR_CMD: No Sub block found in the MSG 0x0003 ERROR_CMD: Incorrect Sub block ID 0x0004 ERROR_CMD: Incorrect Sub block Length 0x0005 ERROR_CMD: Incorrect Sub block data 0x0006 ERROR_PROC: Error in processing the command 0x0007 ERROR_FILECRCMISMATCH: File CRC mismatched 0x0008 ERROR_FILETYPEMISMATCH: File type mismatched w.r.t. magic number 0x0009 See Section 6 for details on error codes from each API - 0xFFFF |

| | | |
|----------------------------|---|-----------------------------------------------------------------------------------------|
| API_RESP_ER- ROR_SBC_ID | 2 | 0x0000 Sub-Block ID in which Error Occurred for sub - block related errors 0xFFFF |
|----------------------------|---|-----------------------------------------------------------------------------------------|

5.2 Sub blocks related to AWR_RF_STATIC_CONF_SET_MSG

5.2.1 Sub block 0x0080 – AWR_CHAN_CONF_SET_SB

This sub block contains static device configurations (applicable for the given power cycle) - how many RX and TX channels are needed for operation. It also defines static configurations related to whether the sensor uses a single mmWave or multiple chips to realize a larger antenna array (multiple is applicable only in AWR1243). Table 5.2 describes the contents of this sub block.

Table 5.2: AWR_CHAN_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0080 |
| SBLKLEN | 2 | Value = 12 |
| RX_CHAN_EN | 2 | This field specifies which RX channels are to be enabled Bit Description b0 RX_CHAN0_EN 0 Disable RX Channel 0 1 Enable RX Channel 0 b1 RX_CHAN1_EN 0 Disable RX Channel 1 1 Enable RX Channel 1 b2 RX_CHAN2_EN 0 Disable RX Channel 2 1 Enable RX Channel 2 b3 RX_CHAN3_EN 0 Disable RX Channel 3 1 Enable RX Channel 3 b15:4 RESERVED 0b000000000000 |

Continued on next page

Table 5.2 – continued from previous page

| TX_CHAN_EN | 2 | <p>This field specifies which TX channels are to be enabled</p> <table border="0"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>TX_CHAN0_EN</td> </tr> <tr> <td>0</td> <td>Disable TX Channel 0</td> </tr> <tr> <td>1</td> <td>Enable TX Channel 0</td> </tr> <tr> <td>b1</td> <td>TX_CHAN1_EN</td> </tr> <tr> <td>0</td> <td>Disable TX Channel 1</td> </tr> <tr> <td>1</td> <td>Enable TX Channel 1</td> </tr> <tr> <td>b2</td> <td>TX_CHAN2_EN (3rd Tx is supported only on some of the devices, Please refer device data sheet)</td> </tr> <tr> <td>0</td> <td>Disable TX Channel 2</td> </tr> <tr> <td>1</td> <td>Enable TX Channel 2</td> </tr> <tr> <td>b15:3</td> <td>RESERVED</td> </tr> <tr> <td></td> <td>0b0000000000000</td> </tr> </tbody> </table> | Bit | Description | b0 | TX_CHAN0_EN | 0 | Disable TX Channel 0 | 1 | Enable TX Channel 0 | b1 | TX_CHAN1_EN | 0 | Disable TX Channel 1 | 1 | Enable TX Channel 1 | b2 | TX_CHAN2_EN (3rd Tx is supported only on some of the devices, Please refer device data sheet) | 0 | Disable TX Channel 2 | 1 | Enable TX Channel 2 | b15:3 | RESERVED | | 0b0000000000000 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------|--------|----------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------|---|----------------------|---|---------------------|----|-----------------------------------------------------------------------------------------------|---|----------------------|---|---------------------|-------|----------|--|-----------------|
| Bit | Description | | | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | TX_CHAN0_EN | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Disable TX Channel 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Enable TX Channel 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| b1 | TX_CHAN1_EN | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Disable TX Channel 1 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Enable TX Channel 1 | | | | | | | | | | | | | | | | | | | | | | | | | |
| b2 | TX_CHAN2_EN (3rd Tx is supported only on some of the devices, Please refer device data sheet) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Disable TX Channel 2 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Enable TX Channel 2 | | | | | | | | | | | | | | | | | | | | | | | | | |
| b15:3 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0b0000000000000 | | | | | | | | | | | | | | | | | | | | | | | | | |
| CASCADING_CFG | 2 | <p>This field specifies the cascading configuration.</p> <table border="0"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x0000</td> <td>SINGLECHIP: Single mmWave sensor application</td> </tr> <tr> <td>0x0001</td> <td>MULTICHIP_MASTER: Multiple cascade sensor application. This device is a master chip and generates LO and conveys to other slave sensors. This is applicable only for device which supports cascading.</td> </tr> <tr> <td>0x0002</td> <td>MULTICHIP_SLAVE: Multiple cascade sensor application. This device is a slave chip and uses LO conveyed to it by the master sensor. This is applicable only for device which supports cascading.</td> </tr> </tbody> </table> <p>MULTICHIP_MASTER and MULTICHIP_SLAVE are in general referred to as MULTICHIP applications, where larger antenna array sizes are possible in comparison with SINGLECHIP cases.</p> <p>Please refer device data sheet for cascading capability and 20G SYNC pins.</p> | Value | Description | 0x0000 | SINGLECHIP: Single mmWave sensor application | 0x0001 | MULTICHIP_MASTER: Multiple cascade sensor application. This device is a master chip and generates LO and conveys to other slave sensors. This is applicable only for device which supports cascading. | 0x0002 | MULTICHIP_SLAVE: Multiple cascade sensor application. This device is a slave chip and uses LO conveyed to it by the master sensor. This is applicable only for device which supports cascading. | | | | | | | | | | | | | | | | |
| Value | Description | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0000 | SINGLECHIP: Single mmWave sensor application | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0001 | MULTICHIP_MASTER: Multiple cascade sensor application. This device is a master chip and generates LO and conveys to other slave sensors. This is applicable only for device which supports cascading. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0002 | MULTICHIP_SLAVE: Multiple cascade sensor application. This device is a slave chip and uses LO conveyed to it by the master sensor. This is applicable only for device which supports cascading. | | | | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.2 – continued from previous page

| | | Bit | Description |
|-------------------------|---|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CASCADING_ PINOUTCFG | 2 | b0 | FM_CW_CLKOUT_MASTER_DIS Applicable only in MUTICHIP_MASTER device. Default value is 0 0 Enable FM_CW_CLKOUT on master 1 Disable FM_CW_CLKOUT on master |
| | | b1 | FM_CW_SYNCOUT_MASTER_DIS Applicable only in MULTICHIP_MASTER device. Default value is 0 0 Enable FM_CW_SYNCOUT on master 1 Disable FM_CW_SYNCOUT on master |
| | | b2 | FM_CW_CLKOUT_SLAVE_EN Applicable only in MULTICHIP_SLAVE device. Default value is 0 0 Disable FM_CW_CLKOUT on slave 1 Enable FM_CW_CLKOUT on slave |
| | | b3 | FM_CW_SYNCOUT_SLAVE_EN Applicable only in MULTICHIP_SLAVE device. Default value is 0 0 Disable FM_CW_SYNCOUT on slave 1 Enable FM_CW_SYNCOUT on slave |
| | | b4 | INTLO_MASTER_EN Applicable only in MULTICHIP_MASTER device. Default value is 0 0 Use externally looped back LO 1 Use internal LO in master Note that the externally looped-back LO mode is useful when length-matching the 20 GHz path between master and slave device. |
| | | b5 | OSCCLKOUT_MASTER_DIS Applicable only in MULTICHIP_MASTER device. Default value is 0 0 Enable OSCCLKOUT in master 1 Disable OSCCLKOUT in master |
| | | b15:6 | RESERVED |

NOTE: The cascade features are not supported in DFP 1.x (1st generation devices). Please refer latest release note for more info.

5.2.2 Sub block 0x0082 – AWR_ADCOUT_CONF_SET_SB

This sub block contains static device configurations (applicable for the given power cycle) - regarding the data format of the ADC output (including the digital filtering).

Table 5.3 describes the contents of this sub block.

Table 5.3: AWR_ADCOUT_CONF_SB contents

| Field Name | Number of bytes | Description |
|-----------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0082 |
| SBLKLEN | 2 | Value = 12 |
| NUM_ADC_BITS | 1 | Bit Description b1:0 Value Definition 00 12 bits 01 14 bits 10 16 bits Other RESERVED b7:2 RESERVED 0b000000 |
| FULL_SCALE_REDUCTION_FACTOR | 1 | Number of bits to reduce ADC full scale by Valid range: 0 to (16 – Number of ADC bits) For e.g. for 12 bit ADC output, this field can take values 0, 1, 2 or 3 For 14 bit ADC output, this field can take values 0, 1 or 2 For 16 bit ADC output, this field can take only value 0 Example: If the user desires 12 bit ADC output, then the digital front end (DFE) chain drops 4 LSBs before placing the data in ADC buffer (DFE output is 16 bits wide). If the user sets FULL_SCALE_REDUCTION_FACTOR as 1, then the DFE will drop only 3 LSBs but still restricting the data in ADC buffer to be within $\pm 2^{12}$. This allows wider ADC swings in smaller signal conditions. |

Continued on next page

Table 5.3 – continued from previous page

| | | | |
|-------------|---|--------|--------------------------------------|
| ADC_OUT_FMT | 2 | Bits | Description |
| | | b1:0 | Value Definition |
| | | 00 | Real |
| | | 01 | Complex 1x (image band filtered out) |
| | | 10 | Complex 2x (image band visible) |
| | | 11 | Pseudo Real |
| | | b15:2 | RESERVED |
| | | | 0b00000000000000 |
| RESERVED | 2 | 0x0000 | |
| RESERVED | 2 | 0x0000 | |

5.2.3 Sub block 0x0083 – AWR_LOWPOWERMODE_CONF_SET_SB

This sub block contains static device configurations (applicable for this power cycle) - Sigma Delta ADC root sampling clock rate (reducing rate to half to save power in small IF bandwidth applications).

Table 5.4 describes the contents of this sub block.

Table 5.4: AWR_LOWPOWERMODE_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------|-----------------|-------------------------|
| SBLKID | 2 | Value = 0x0083 |
| SBLKLEN | 2 | Value = 8 |
| RESERVED | 2 | 0x0000 |
| LP_ADC.MODE | 2 | Value Definition |
| | | 0x00 Regular ADC mode |
| | | 0x01 Low power ADC mode |

NOTE: Low power ADC mode is mandatory on a 5 MHz part variant (for e.g. xWR1642), Normally if IF BW \leq 7.5MHz then low power mode setting is recommended.

5.2.4 Sub block 0x0084 – AWR_DYNAMICPOWERSAVE_CONF_SET_SB

This sub block defines static device configuration - whether to enable dynamic power saving during inter-chirp IDLE times by turning off various circuits e.g. TX, RX, LO Distribution blocks. If

Idle time + Tx start time < 10us or Idle time < 3.5us then inter-chirp dynamic power save option will be disabled, in that case, 15us of inter-burst idle time will be utilized to configure sequencer LO, TX and RX signal timings by firmware.

Table 5.4 describes the contents of this sub block.

Table 5.5: AWR_DYNAMICPOWERSAVE_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0084 |
| SBLKLEN | 2 | Value = 8 |
| BLOCK_CFG | 2 | Bits Definition b0 Enable power save by switching off TX during inter-chirp IDLE period 0 Disable 1 Enable Default value: 1 (power saving is enabled) b1 Enable power save by switching off RX during inter-chirp IDLE period 0 Disable 1 Enable Default value: 1 (power saving is enabled) b2 Enable power save by switching off LO Distribution blocks during inter-chirp IDLE period 0 Disable 1 Enable Default value: 1 (power saving is enabled) b15:3 RESERVED 0b00000000000000 |
| RESERVED | 2 | 0x0000 |

5.2.5 Sub block 0x0085 – AWR_HIGHSPEEDINTFCLK_CONF_SET_SB

This sub block contains static device configurations (applicable for the given power cycle) - regarding high speed interface clock rates which are related to sending the ADC data from AWR device to the host in either LVDS or CSI2 format.

Table 5.6 describes the contents of this sub block.

Table 5.6: AWR_HIGHSPEEDINTFCLK_CONF_SET_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|------|------|------|--|----|----|----|----|------------|----------|---------|---------|---------|------------|----------|---------|---------|---------|------------|----------|----------|---------|---------|------------|----------|----------|----------|----------|
| SBLKID | 2 | Value = 0x0085 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HSICLKRATE_CODE | 2 | <p>This field indicates the high speed interface input clock rate, needed by the LVDS or CSI2 module. It should be N times the final serial data rate, where $N = 2$ in DDR mode and $N = 1$ in SDR mode.</p> <p>Bit 15:5 = Reserved (all 0). Bit 3:0 are to be set based on desired rate as follows:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>b1:0</th> <th>b1:0</th> <th>b1:0</th> <th>b1:0</th> </tr> </thead> <tbody> <tr> <td></td> <td>00</td> <td>01</td> <td>10</td> <td>11</td> </tr> </tbody> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td>b3:2 00</td> <td>Reserved</td> <td>800 MHz</td> <td>400 MHz</td> <td>200 MHz</td> </tr> <tr> <td>b3:2 01</td> <td>Reserved</td> <td>900 MHz</td> <td>450 MHz</td> <td>225 MHz</td> </tr> <tr> <td>b3:2 10</td> <td>Reserved</td> <td>1200 MHz</td> <td>600 MHz</td> <td>300 MHz</td> </tr> <tr> <td>b3:2 11</td> <td>Reserved</td> <td>1800 MHz</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table> <p>For example, for 900 Mbps output rate with DDR, choose Bit3:0=0b1101, and for 450 Mbps output rate with SDR, choose Bit3:0=0b0110.</p> | | b1:0 | b1:0 | b1:0 | b1:0 | | 00 | 01 | 10 | 11 | b3:2 00 | Reserved | 800 MHz | 400 MHz | 200 MHz | b3:2 01 | Reserved | 900 MHz | 450 MHz | 225 MHz | b3:2 10 | Reserved | 1200 MHz | 600 MHz | 300 MHz | b3:2 11 | Reserved | 1800 MHz | Reserved | Reserved |
| | b1:0 | b1:0 | b1:0 | b1:0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 00 | 01 | 10 | 11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3:2 00 | Reserved | 800 MHz | 400 MHz | 200 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3:2 01 | Reserved | 900 MHz | 450 MHz | 225 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3:2 10 | Reserved | 1200 MHz | 600 MHz | 300 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3:2 11 | Reserved | 1800 MHz | Reserved | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

5.2.6 Sub block 0x0086 – AWR_RF_DEVICE_CFG_SB

This sub block configures the direction of async event from BSS. Typically async events are sent to MSS. With this API, the user can configure the destination of async event.

Table 5.7 describes the contents of this sub block.

Table 5.7: AWR_RF_DEVICE_CFG_SB contents

| Field Name | Number of bytes | Description |
|-----------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0086 |
| SBLKLEN | 2 | Value = 16 |
| RF_AE_DIRECTION | 4 | <p>Bits Definition</p> <p>b1:0 ASYNC_EVENT_DIR</p> <p> 00 BSS to MSS</p> <p> 01 BSS to HOST</p> <p> 10 BSS to DSS</p> <p> 11 RESERVED</p> <p>The ASYNC_EVENT_DIR controls the direction for following ASYNC_EVENTS</p> <p> 1. CPU_FAULT</p> <p> 2. ESM_FAULT</p> <p> 3. ANALOG_FAULT</p> <p>All other ASYNC_EVENTS are sent to the subsystem which issues the API</p> <p>Default value: 0b00</p> <p>b3:2 MONITORING_ASYNC_EVENT_DIR</p> <p> 00 BSS to MSS</p> <p> 01 BSS to HOST</p> <p> 10 BSS to DSS</p> <p> 11 RESERVED</p> <p>Default value: 0b00</p> <p>b31:4 RESERVED</p> <p> 0x0000000</p> |

Continued on next page

Table 5.7 – continued from previous page

| | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------|----|---------------------------------|---|---------------------------------|---|---------------------------------|------------------|----------|-----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|---|--------------------------------|------------------|--|------|----------|----------|--|
| AE_CONTROL | 1 | <table border="0"> <tr> <td>Bits</td> <td>Definition</td> </tr> <tr> <td>b0</td> <td>FRAME_START_ASYNC_EVENT_DIS</td> </tr> <tr> <td>0</td> <td>Frame Start async event enable</td> </tr> <tr> <td>1</td> <td>Frame Start async event disable</td> </tr> <tr> <td colspan="2">Default value: 0</td> </tr> <tr> <td>b1</td> <td>FRAME_STOP_ASYNC_EVENT_DIS</td> </tr> <tr> <td>0</td> <td>Frame Stop async event enable</td> </tr> <tr> <td>1</td> <td>Frame Stop async event disable</td> </tr> <tr> <td colspan="2">Default value: 0</td> </tr> <tr> <td>b7:2</td> <td>RESERVED</td> </tr> <tr> <td colspan="2">0b000000</td> </tr> </table> | Bits | Definition | b0 | FRAME_START_ASYNC_EVENT_DIS | 0 | Frame Start async event enable | 1 | Frame Start async event disable | Default value: 0 | | b1 | FRAME_STOP_ASYNC_EVENT_DIS | 0 | Frame Stop async event enable | 1 | Frame Stop async event disable | Default value: 0 | | b7:2 | RESERVED | 0b000000 | |
| Bits | Definition | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | FRAME_START_ASYNC_EVENT_DIS | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Frame Start async event enable | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Frame Start async event disable | | | | | | | | | | | | | | | | | | | | | | | |
| Default value: 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| b1 | FRAME_STOP_ASYNC_EVENT_DIS | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Frame Stop async event enable | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Frame Stop async event disable | | | | | | | | | | | | | | | | | | | | | | | |
| Default value: 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| b7:2 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | |
| 0b000000 | | | | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | | | | | | | |
| BSS_DIG_CTRL | 1 | <table border="0"> <tr> <td>Bits</td> <td>Definition</td> </tr> <tr> <td>b0</td> <td>WDT_DISABLE</td> </tr> <tr> <td>0</td> <td>Keep watchdog disabled</td> </tr> <tr> <td>1</td> <td>Enable watchdog</td> </tr> <tr> <td>b7:1</td> <td>RESERVED</td> </tr> <tr> <td colspan="2">0b0000000</td> </tr> <tr> <td colspan="2">NOTE: The Windowed WDT can be enabled only in Sw triggered framing Mode or in cascade mode where frames of all the devices synchronized with same clock source, if frames are triggered from Hw trigger pulse generated from unsynchronized clock then WDT can not be enabled. Refer section 9.4 for more details on WDT timing and programming window 344</td> </tr> </table> | Bits | Definition | b0 | WDT_DISABLE | 0 | Keep watchdog disabled | 1 | Enable watchdog | b7:1 | RESERVED | 0b0000000 | | NOTE: The Windowed WDT can be enabled only in Sw triggered framing Mode or in cascade mode where frames of all the devices synchronized with same clock source, if frames are triggered from Hw trigger pulse generated from unsynchronized clock then WDT can not be enabled. Refer section 9.4 for more details on WDT timing and programming window 344 | | | | | | | | | |
| Bits | Definition | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | WDT_DISABLE | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Keep watchdog disabled | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Enable watchdog | | | | | | | | | | | | | | | | | | | | | | | |
| b7:1 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | |
| 0b0000000 | | | | | | | | | | | | | | | | | | | | | | | | |
| NOTE: The Windowed WDT can be enabled only in Sw triggered framing Mode or in cascade mode where frames of all the devices synchronized with same clock source, if frames are triggered from Hw trigger pulse generated from unsynchronized clock then WDT can not be enabled. Refer section 9.4 for more details on WDT timing and programming window 344 | | | | | | | | | | | | | | | | | | | | | | | | |
| ASYNC_EVENT_CRC_CONFIG | 1 | <table border="0"> <tr> <td>Value</td> <td>Description</td> </tr> <tr> <td>0</td> <td>16 bit CRC for BSS async events</td> </tr> <tr> <td>1</td> <td>32 bit CRC for BSS async events</td> </tr> <tr> <td>2</td> <td>64 bit CRC for BSS async events</td> </tr> </table> | Value | Description | 0 | 16 bit CRC for BSS async events | 1 | 32 bit CRC for BSS async events | 2 | 64 bit CRC for BSS async events | | | | | | | | | | | | | | |
| Value | Description | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 16 bit CRC for BSS async events | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 32 bit CRC for BSS async events | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 64 bit CRC for BSS async events | | | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 3 | 0x000000 | | | | | | | | | | | | | | | | | | | | | | |

5.2.7 Sub block 0x0087 – AWR_RF_RADAR_MISC_CTL_SB

This sub block controls miscellaneous global RF controls for e.g. per-chirp phase shifter global control.

Table [5.8](#) describes the contents of this sub block.

Table 5.8: AWR_RF_MISC_CTL_SB contents

| Field Name | Number of bytes | Description |
|-------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0087 |
| SBLKLEN | 2 | Value = 12 |
| RF_MISC_CTL | 4 | Bits Definition b0 PERCHIRP_PHASESHIFTER_EN 0 Per chirp phase shifter is disabled 1 Per chirp phase shifter is enabled This control is applicable only in devices which support phase shifter (refer data sheet). For other devices, this is a RESERVED bit and should be set to 0. Default value: 0 b31:1 RESERVED 0b000_0000_0000_0000_0000_0000_0000 |
| RESERVED | 4 | 0x00000000 |

5.2.8 Sub block 0x0088 – AWR_CAL_MON_FREQUENCY_LIMITS_SB

This sub block sets the limits for RF frequency transmission.

Table 5.9 describes the contents of this sub block.

Table 5.9: AWR_CAL_MON_FREQUENCY_LIMITS_SB contents

| Field Name | Number of bytes | Description |
|-----------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0088 |
| SBLKLEN | 2 | Value = 16 |
| FREQ.LIMIT_LOW | 2 | The sensor's lower frequency limit for calibrations and monitoring is encoded in 2 bytes (16 bit unsigned number) 1 LSB = 100 MHz For 77GHz Devices(76GHz to 81Ghz): Valid range: 760 to 810 Default value: 760 (If this API is not issued) For 60GHz Devices(57GHz to 64Ghz): Valid range: 570 to 640 Default value: 570 (If this API is not issued) |
| FREQ.LIMIT_HIGH | 2 | The sensor's higher frequency limit for calibrations and monitoring is encoded in 2 bytes (16 bit unsigned number) |

Continued on next page

Table 5.9 – continued from previous page

| | | |
|----------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | 1 LSB = 100 MHz For 77GHz Devices (76GHz to 81Ghz): Valid range: 760 to 810 Default value: 810 (If this API is not issued) For 60GHz Devices(57GHz to 64Ghz): Valid range: 570 to 640 Default value: 640 (If this API is not issued) NOTE: <code>FREQ_LIMIT_HIGH</code> should be strictly greater than <code>FREQ_LIMIT_LOW</code> Examples: For an LRR device deployed in the US, one might typically configure <code>FREQ_LIMIT_LOW</code> to 760 and <code>FREQ_LIMIT_HIGH</code> to 770. |
| RESERVED | 8 | RESERVED 0x0000_0000_0000_0000 |

NOTE: This API is not supported in IWR6843 ES1.0

5.2.9 Sub block 0x0089 – `AWR_RF_INIT_CALIBRATION_CONF_SB`

This sub block configures device to perform boot time calibration.

Table 5.10 describes the contents of this sub block.

Table 5.10: `AWR_RF_INIT_CALIBRATION_CONF_SB` contents

| Field Name | Number of bytes | Description |
|---------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0089 |
| SBLKLEN | 2 | Value = 16 |
| RF_INIT_CALIB_ENABLE_MASK | 4 | Normally, upon receiving RF INIT message, the BSS performs all relevant initial calibrations. This step can be disabled by the host by setting the corresponding calibration bit in this field to 0x0. If disabled, the host needs to send the INJECT CALIB DATA message so that the BSS can operate using the calibration data thus injected. Internal/Debug use: Each of these calibrations can be selectively disabled by issuing this message before RF INIT message. |

Continued on next page

Table 5.10 – continued from previous page

| | | Bit | Definition |
|----------|---|--------|-------------------------------------------------------------------------------------------------|
| | | b0 | RESERVED |
| | | b1 | RESERVED |
| | | b2 | RESERVED |
| | | b3 | RESERVED |
| | | b4 | Enable LODIST calibration |
| | | b5 | Enable RX ADC DC offset calibration |
| | | b6 | Enable HPF cutoff calibration |
| | | b7 | Enable LPF cutoff calibration |
| | | b8 | Enable Peak detector calibration |
| | | b9 | Enable TX power calibration |
| | | b10 | Enable RX gain calibration |
| | | b11 | Enable TX Phase calibration (Device dependent feature, please refer data sheet) |
| | | b12 | Enable RX IQMM calibration |
| | | b31:13 | RESERVED |
| | | | 0b000_0000_0000_0000_0000 |
| | | | Default value: 0x1FF0 |
| | | | NOTE: If TX power calibration is disabled, then backoff other than 0 dB is not supported |
| RESERVED | 4 | | 0x00000000 |
| RESERVED | 4 | | 0x00000000 |

NOTE1: The APLL, SYNTH1 and SYNTH2 calibrations are always triggered by default on RF init command

NOTE2: In IWR6843 ES1.0, only APLL, SYNTH1, SYNTH2 and LODIST Calibrations are supported, rest of the Calibrations are not supported.

5.2.10 Sub block 0x008A – AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB

This sub block sets the limits for RF frequency transmission for each TX and also TX power limits.

Table 5.11: AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB contents

| Field Name | Number of bytes | Description |
|------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x008A |
| SBLKLEN | 2 | Value = 28 |
| FREQ.LIMIT_ LOW.TX0 | 2 | <p>The sensor's lower frequency limit for calibrations and monitoring for TX0 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices(76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 7600 (If this API is not issued)</p> <p>For 60GHz Devices(57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 5700 (If this API is not issued)</p> |
| FREQ.LIMIT_ LOW.TX1 | 2 | <p>The sensor's lower frequency limit for calibrations and monitoring for TX1 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices(76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 7600 (If this API is not issued)</p> <p>For 60GHz Devices(57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 5700 (If this API is not issued)</p> |
| FREQ.LIMIT_ LOW.TX2 | 2 | <p>The sensor's lower frequency limit for calibrations and monitoring for TX2 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices(76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 7600 (If this API is not issued)</p> <p>For 60GHz Devices(57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 5700 (If this API is not issued)</p> |

Continued on next page

Table 5.11 – continued from previous page

| | | |
|---------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FREQ_LIMIT_HIGH_TX0 | 2 | <p>The sensor's higher frequency limit for calibrations and monitoring for TX0 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices (76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 8100 (If this API is not issued)</p> <p>For 60GHz Devices (57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 6400 (If this API is not issued)</p> <p>NOTE: FREQ_LIMIT_HIGH_TXn should be strictly greater than FREQ_LIMIT_LOW_TXn</p> |
| FREQ_LIMIT_HIGH_TX1 | 2 | <p>The sensor's higher frequency limit for calibrations and monitoring for TX1 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices (76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 8100 (If this API is not issued)</p> <p>For 60GHz Devices (57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 6400 (If this API is not issued)</p> <p>NOTE: FREQ_LIMIT_HIGH_TXn should be strictly greater than FREQ_LIMIT_LOW_TXn</p> |
| FREQ_LIMIT_HIGH_TX2 | 2 | <p>The sensor's higher frequency limit for calibrations and monitoring for TX2 is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 10 MHz</p> <p>For 77GHz Devices (76GHz to 81Ghz): Valid range: 7600 to 8100 Default value: 8100 (If this API is not issued)</p> <p>For 60GHz Devices (57GHz to 64Ghz): Valid range: 5700 to 6400 Default value: 6400 (If this API is not issued)</p> <p>NOTE: FREQ_LIMIT_HIGH_TXn should be strictly greater than FREQ_LIMIT_LOW_TXn</p> |
| TX0_POWER_BACKOFF | 1 | <p>TX0 output power back off</p> <p>1 LSB = 1 dB</p> <p>Valid values: 0, 3, 6, 9 Default value: 0</p> |
| TX1_POWER_BACKOFF | 1 | <p>TX1 output power back off</p> <p>1 LSB = 1 dB</p> <p>Valid values: 0, 3, 6, 9 Default value: 0</p> |

Continued on next page

Table 5.11 – continued from previous page

| | | |
|-------------------|---|-------------------------------------------------------------------------------------------|
| TX2_POWER_BACKOFF | 1 | TX2 output power back off 1 LSB = 1 dB Valid values: 0, 3, 6, 9 Default value: 0 |
| RESERVED | 1 | 0x00 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |

NOTE: This API is not supported in IWR6843 ES1.0

5.2.11 Sub block 0x008B – AWR_CAL_DATA_RESTORE_SB

This sub block restores the calibration data which was stored previously using the [AWR_CAL_DATA_SAVE_SB](#) command. The async event [AWR_AE_RF_INITCALIBSTATUS_SB](#) will be issued after this API, this indicates success of the calibration data restore. The calibration data contents are defined in page 60 in AWR_CAL_DATA table.

Table 5.12: AWR_CAL_DATA_RESTORE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------------------------------------------------------|
| SBLKID | 2 | Value = 0x008B |
| SBLKLEN | 2 | Value = 232 |
| RESERVED | 2 | 0x0000 |
| CHUNK_ID | 2 | Index of the current chunk |
| CAL_DATA | 224 | Calibration data chunk which was stored in non-volatile memory |

NOTE1: All 3 chunks of 224 bytes each shall be sent to radar device to complete the restore process and to generate AWR_AE_RF_INITCALIBSTATUS_SB AE.

NOTE2: Refer recommended API sequence and order in page 256 for more details on sequence of issuing this API.

NOTE3: This API is not supported in IWR6843 ES1.0

5.2.12 Sub block 0x008C – AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB

This sub block restores the calibration data which was stored previously using the AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB command. This is device specific feature, please refer data sheet.

Table 5.13: AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | | | | |
|------------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------|---------------------------------------------------------|---|--------|------------------|---|--------|------------------|---|---------|------------------|---|---------|------------------|---|---|--|----|----------|----------------------|
| SBLKID | 2 | Value = 0x008C | | | | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 136 | | | | | | | | | | | | | | | | | | | | | |
| TX_INDXX | 1 | Index of the transmit channel for which the following data applies | | | | | | | | | | | | | | | | | | | | | |
| CAL_APPLY | 1 | Set this to 1 after applying calibration data from all transmitters. This bit will indicate to the firmware to start the correction process. | | | | | | | | | | | | | | | | | | | | | |
| OBS.PHSHIFT_DATA | 128 | <p>Observed phase shift corresponding to each desired phase shift. Index n corresponds to desired phase shift of $n \times 5.625^\circ$. For e.g.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">n</th> <th style="text-align: center;">Desired phase shift</th> <th style="text-align: center;">Observed phase shift is injected in the following bytes</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0.000°</td> <td style="text-align: center;">byte[1], byte[0]</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">5.625°</td> <td style="text-align: center;">byte[3], byte[2]</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">11.250°</td> <td style="text-align: center;">byte[5], byte[4]</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">16.875°</td> <td style="text-align: center;">byte[7], byte[6]</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td></td> </tr> <tr> <td style="text-align: center;">63</td> <td style="text-align: center;">354.375°</td> <td style="text-align: center;">byte[127], byte[126]</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ / 2^{10}$</p> | n | Desired phase shift | Observed phase shift is injected in the following bytes | 0 | 0.000° | byte[1], byte[0] | 1 | 5.625° | byte[3], byte[2] | 2 | 11.250° | byte[5], byte[4] | 3 | 16.875° | byte[7], byte[6] | ⋮ | ⋮ | | 63 | 354.375° | byte[127], byte[126] |
| n | Desired phase shift | Observed phase shift is injected in the following bytes | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0.000° | byte[1], byte[0] | | | | | | | | | | | | | | | | | | | | | |
| 1 | 5.625° | byte[3], byte[2] | | | | | | | | | | | | | | | | | | | | | |
| 2 | 11.250° | byte[5], byte[4] | | | | | | | | | | | | | | | | | | | | | |
| 3 | 16.875° | byte[7], byte[6] | | | | | | | | | | | | | | | | | | | | | |
| ⋮ | ⋮ | | | | | | | | | | | | | | | | | | | | | | |
| 63 | 354.375° | byte[127], byte[126] | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | | | | | | |

5.3 Sub blocks related to AWR_RF_STATIC_CONF_GET_MSG

5.3.1 Sub block 0x00A0 – 0x00AA – RESERVED

5.3.2 Sub block 0x00AB – AWR_CAL_DATA_SAVE_SB

This sub block reads the calibration data from the device which can be injected later using the AWR_CAL_DATA_RESTORE_SB command.

NOTE: The total size of the calibration data is 672 bytes, this has been split into 3 chunks (NUM_CHUNKS) of 224 bytes each due to SPI limitation. The Host should receive all these 3 chunks from radar device, later host can store only relevant data in non volatile memory.

Table 5.14: AWR_CAL_DATA_SAVE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x00AB |
| SBLKLEN | 2 | Value = 8 |
| RESERVED | 2 | 0x0000 |
| CHUNK.ID | 2 | Index of the requested chunk Valid values: 0 to NUM_CHUNKS - 1 |

Response to the above command will contain the calibration data which is formatted as shown below

Table 5.15: AWR_CAL_DATA_SAVE_SB response packet contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-------------------------------------------------|
| SBLKID | 2 | Value = 0x00AB |
| SBLKLEN | 2 | Value = 232 |
| NUM_CHUNKS | 2 | Total number of calibration data chunks |
| CHUNK.ID | 2 | Current chunk number |
| CAL_DATA | 224 | Calibration data, refer CAL_DATA contents below |

Table 5.16: AWR_CAL_DATA contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-------------|
|------------|-----------------|-------------|

| | | |
|--------------------------|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAL_VALIDITY_STATUS | 4 | <p>This field indicates the status of each calibration (0 – FAIL, 1 – PASS). If a particular calibration was not enabled, then its corresponding field should be ignored.</p> <p>Bit Definition (0 – FAIL, 1 – PASS)</p> <p>b0 RESERVED</p> <p>b1 APLL tuning (Ignore while store restore)</p> <p>b2 SYNTH VCO1 tuning (Ignore while store restore)</p> <p>b3 SYNTH VCO2 tuning (Ignore while store restore)</p> <p>b4 LODIST calibration (Ignore while store restore)</p> <p>b5 RX ADC DC offset calibration</p> <p>b6 HPF cutoff calibration</p> <p>b7 LPF cutoff calibration</p> <p>b8 Peak detector calibration (optional)</p> <p>b9 TX Power calibration (optional)</p> <p>b10 RX gain calibration</p> <p>b11 TX Phase calibration (Ignore while store restore)</p> <p>b12 RX IQMM calibration</p> <p>b31:13 RESERVED</p> <p>The recommended Validity status bits while restoring is 0x000014E0, assuming only RX_ADC_DC_CAL_DATA, HPF_CAL_DATA, LPF_CAL_DATA, RX_RF_GAIN_CAL_DATA and IQMM_CAL_DATA are stored and restored.</p> |
| CAL_VALIDITY_STATUS_COPY | 4 | Redundant CAL_VALIDITY_STATUS value, this value should match with CAL_VALIDITY_STATUS |
| IF_STAGE_CAL_STATUS | 4 | <p>This field indicates the status of IF stage calibration (0 – FAIL, 1 – PASS). If a particular calibration was not enabled, then its corresponding field should be ignored.</p> <p>Bit Definition (0 – FAIL, 1 – PASS)</p> <p>b0 HPF1</p> <p>b1 HPF2</p> <p>b2 LPF1</p> <p>b3 LPF2</p> <p>b31:4 RESERVED</p> <p>This value shall be set to 0xF if HPF and LPF calibration validity status are PASS</p> |
| RESERVED | 4 | RESERVED |
| CAL_TEMPERATURE | 2 | Temperature at which boot calibration is done |
| RESERVED | 14 | RESERVED |

| | | |
|---------------------|-----|----------------------------------|
| RX_ADC_DC_CAL_DATA | 16 | Rx chain ADC DC calibration data |
| HPF1_CAL_DATA | 1 | HPF1 calibration data |
| HPF2_CAL_DATA | 1 | HPF2 calibration data |
| RESERVED | 2 | RESERVED |
| LPF1_CAL_DATA | 24 | LPF1 calibration data |
| LPF2_CAL_DATA | 24 | LPF2 calibration data |
| RX_RF_GAIN_CAL_DATA | 12 | RX RF gain calibration data |
| IQMM_CAL_DATA | 72 | RX IQMM calibration data |
| TX_POWER_CAL_DATA | 84 | TX Power calibration data |
| POWER_DET_CAL_DATA | 348 | Power detector calibration data |
| RESERVED | 56 | RESERVED |

| | |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NOTE1: | Before storing the calibration data in non volatile memory, the host shall make sure validity status of all enabled calibrations are SET to value 1 including APLL, VCO1, VCO2 and LODIST calibration validity in RF_INIT of radar device. |
| NOTE2: | Host can store only relevant calibration data in non volatile memory and corresponding validity bits shall be set to 1 in AWR_CAL_DATA_RESTORE_SB and rest of the validity bits should be clear to 0 before restoring the data to radar device. |
| NOTE3: | Host shall ignore APLL, VCO1, VCO2 and LODIST calibration validity bits while restoring, these calibrations will be done in each device power-up. |

| | |
|---------------|--------------------------------------------|
| NOTE4: | This API is not supported in IWR6843 ES1.0 |
|---------------|--------------------------------------------|

5.3.3 Sub block 0x00AC – AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB

This sub block reads the phase shifter calibration data from the device which can be injected later using the AWR_PHASE_SHIFTER_CAL_DATA_RESTORE_SB command. This is device specific feature, please refer data sheet.

Table 5.17: AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-----------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x00AC |
| SBLKLEN | 2 | Value = 8 |
| TX_INDx | 1 | Index of the transmitter channel for which the phase shift is desired |
| RESERVED | 3 | 0x000000 |

Response to the above command will contain the phase shifter calibration data which is formatted as shown below

Table 5.18: AWR_PHASE_SHIFTER_CAL_DATA_SAVE_SB response packet contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | | | | |
|------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------|-----------------------------------------------------|---|--------|------------------|---|--------|------------------|---|---------|------------------|---|---------|------------------|---|---|--|----|----------|----------------------|
| SBLKID | 2 | Value = 0x00AC | | | | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 136 | | | | | | | | | | | | | | | | | | | | | |
| TX_INDx | 1 | Index of the transmitter channel for which the following phase shift values applies | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | | | | | |
| OBS_PHSHIFT_DATA | 128 | <p>Observed phase shift corresponding to each desired phase shift. Index n corresponds to desired phase shift of $n \times 5.625^\circ$. For e.g.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">n</th> <th style="text-align: left;">Desired phase shift</th> <th style="text-align: left;">Observed phase shift is read in the following bytes</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.000°</td> <td>byte[1], byte[0]</td> </tr> <tr> <td>1</td> <td>5.625°</td> <td>byte[3], byte[2]</td> </tr> <tr> <td>2</td> <td>11.250°</td> <td>byte[5], byte[4]</td> </tr> <tr> <td>3</td> <td>16.875°</td> <td>byte[7], byte[6]</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td></td> </tr> <tr> <td>63</td> <td>354.375°</td> <td>byte[127], byte[126]</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ / 2^{10}$</p> | n | Desired phase shift | Observed phase shift is read in the following bytes | 0 | 0.000° | byte[1], byte[0] | 1 | 5.625° | byte[3], byte[2] | 2 | 11.250° | byte[5], byte[4] | 3 | 16.875° | byte[7], byte[6] | ⋮ | ⋮ | | 63 | 354.375° | byte[127], byte[126] |
| n | Desired phase shift | Observed phase shift is read in the following bytes | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0.000° | byte[1], byte[0] | | | | | | | | | | | | | | | | | | | | | |
| 1 | 5.625° | byte[3], byte[2] | | | | | | | | | | | | | | | | | | | | | |
| 2 | 11.250° | byte[5], byte[4] | | | | | | | | | | | | | | | | | | | | | |
| 3 | 16.875° | byte[7], byte[6] | | | | | | | | | | | | | | | | | | | | | |
| ⋮ | ⋮ | | | | | | | | | | | | | | | | | | | | | | |
| 63 | 354.375° | byte[127], byte[126] | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | | | | | | |

5.4 Sub blocks related to AWR_RF_INIT_MSG

5.4.1 Sub block 0x00C0 – AWR_RF_INIT_SB

This sub block, needed to be initially issued, triggers one time calibrations such as those related to APLL and synthesizer. The BSS processor is woken up upon receiving this sub block, the RF analog and digital baseband sections are initialized.

Table 5.19 describes the content of this sub block.

Table 5.19: AWR_RF_INIT_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x00C0 |
| SBLKLEN | 2 | Value = 4 |

NOTE1: This sub block will be acknowledged immediately but an async event [AWR_AE_RF_INITCALIBSTATUS_SB](#) from BSS will indicate that the RF initialization is complete. No commands shall be sent to BSS till the async event is received.

NOTE2: The following boot-time calibrations are susceptible to corruption by interference. The calibrations may result in false configuration of the RF analog sections due to corruption by interference during the calibration measurements.

- a. RX gain calibration (susceptible to interference)
- b. RX IQMM calibration (susceptible to interference)
- c. TX Phase calibration (susceptible to interference)

It is recommended to perform factory calibration and store the calibration data in non volatile memory using [AWR_CAL_DATA_SAVE_SB](#) API. This data can be restored to radar device using [AWR_CAL_DATA_RESTORE_SB](#) API. More info related to save restore provided in page [59](#)

5.5 Sub blocks related to AWR_RF_DYNAMIC_CONF_SET_MSG

5.5.1 Sub block 0x0100 – AWR_PROFILE_CONF_SET_SB

This sub block contains FMCW radar chirp profiles or properties (FMCW slope, chirp duration, TX power etc.). Since the device supports multiple profiles, each profile is defined in this sub block. Internal RF and analog calibrations may be triggered upon receiving this sub block and [ASYNC_EVENT](#) response sent once completed.

NOTE: This API can be issued dynamically to change profile parameters. Few parameters which cannot be changed are

1. PF_NUM_ADC_SAMPLES
2. PF_DIGITAL_OUTPUT_SAMPLING_RATE
3. Programmable filter coefficients in
xWR1642/IWR6843/xWR1843

Table 5.20 describes the contents of this sub block.

Table 5.20: AWR_PROFILE_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-----------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0100 |
| SBLKLEN | 2 | Value = 48 |
| PF_INDX | 2 | The profile index for which the rest of the fields are applicable for |

Continued on next page

Table 5.20 – continued from previous page

| | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------|----|------------------------------------------------------------------|---|----------------------------|---|----------------------------------|----|------------------------------------------------------------|---|---------------------------------------------------------|---|-------------------------------------|-------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|------|----------------------|
| PF_VCO_SELECT | 1 | <table border="0"> <tr> <td>Bit</td> <td>Description</td> </tr> <tr> <td>b0</td> <td>FORCE_VCO_SEL (Not supported for production, debug purpose only)</td> </tr> <tr> <td>0</td> <td>Use internal VCO selection</td> </tr> <tr> <td>1</td> <td>Forced external VCO selection</td> </tr> <tr> <td>b1</td> <td>VCO_SEL (Not supported for production, debug purpose only)</td> </tr> <tr> <td>0</td> <td>VCO1 (77G: 76–78GHz, 60G: 57-61GHz)</td> </tr> <tr> <td>1</td> <td>VCO2 (77G: 77–81GHz, 60G: 60–64GHz)</td> </tr> <tr> <td colspan="2">NOTE:</td> </tr> <tr> <td colspan="2">1. xWR1xxx devices: There is an overlap region of 77-78 GHz in which any of the VCOs can be used, for other regions use only the VCO which can work in that region. For e.g. for 76-78 GHz use only VCO1 and for 77-81GHz use only VCO2, for 77-78 GHz, any VCO can be used. Also note that users can inter-mix chirps from different VCOs within the same frame. 2. IWR6843 device: There is an overlap region of 60-61 GHz in which any of the VCOs can be used.</td> </tr> <tr> <td>b7:2</td> <td>RESERVED 0b000000</td> </tr> </table> | Bit | Description | b0 | FORCE_VCO_SEL (Not supported for production, debug purpose only) | 0 | Use internal VCO selection | 1 | Forced external VCO selection | b1 | VCO_SEL (Not supported for production, debug purpose only) | 0 | VCO1 (77G: 76–78GHz, 60G: 57-61GHz) | 1 | VCO2 (77G: 77–81GHz, 60G: 60–64GHz) | NOTE: | | 1. xWR1xxx devices: There is an overlap region of 77-78 GHz in which any of the VCOs can be used, for other regions use only the VCO which can work in that region. For e.g. for 76-78 GHz use only VCO1 and for 77-81GHz use only VCO2, for 77-78 GHz, any VCO can be used. Also note that users can inter-mix chirps from different VCOs within the same frame. 2. IWR6843 device: There is an overlap region of 60-61 GHz in which any of the VCOs can be used. | | b7:2 | RESERVED 0b000000 |
| Bit | Description | | | | | | | | | | | | | | | | | | | | | |
| b0 | FORCE_VCO_SEL (Not supported for production, debug purpose only) | | | | | | | | | | | | | | | | | | | | | |
| 0 | Use internal VCO selection | | | | | | | | | | | | | | | | | | | | | |
| 1 | Forced external VCO selection | | | | | | | | | | | | | | | | | | | | | |
| b1 | VCO_SEL (Not supported for production, debug purpose only) | | | | | | | | | | | | | | | | | | | | | |
| 0 | VCO1 (77G: 76–78GHz, 60G: 57-61GHz) | | | | | | | | | | | | | | | | | | | | | |
| 1 | VCO2 (77G: 77–81GHz, 60G: 60–64GHz) | | | | | | | | | | | | | | | | | | | | | |
| NOTE: | | | | | | | | | | | | | | | | | | | | | | |
| 1. xWR1xxx devices: There is an overlap region of 77-78 GHz in which any of the VCOs can be used, for other regions use only the VCO which can work in that region. For e.g. for 76-78 GHz use only VCO1 and for 77-81GHz use only VCO2, for 77-78 GHz, any VCO can be used. Also note that users can inter-mix chirps from different VCOs within the same frame. 2. IWR6843 device: There is an overlap region of 60-61 GHz in which any of the VCOs can be used. | | | | | | | | | | | | | | | | | | | | | | |
| b7:2 | RESERVED 0b000000 | | | | | | | | | | | | | | | | | | | | | |
| PF_CALLUT_UPDATE | 1 | <table border="0"> <tr> <td>Bit</td> <td>Description</td> </tr> <tr> <td>b0</td> <td>RETAIN_TXCAL_LUT (Not supported in IWR6843 ES1.0)</td> </tr> <tr> <td>0</td> <td>Update TX calibration LUT</td> </tr> <tr> <td>1</td> <td>Do not update TX calibration LUT</td> </tr> <tr> <td>b1</td> <td>RETAIN_RXCAL_LUT (Not supported in IWR6843 ES1.0)</td> </tr> <tr> <td>0</td> <td>Update RX calibration LUT and update RX IQMM correction</td> </tr> <tr> <td>1</td> <td>Do not update RX calibration LUT</td> </tr> <tr> <td>b7:2</td> <td>RESERVED (set it to 0b000000)</td> </tr> <tr> <td colspan="2">If PF_TX_OUTPUT_POWER_BACKOFF is changed then set RETAIN_TXCAL_LUT to 0, else set it to 1 and if PF_RX_GAIN or if sweep bandwidth is changed, then set RETAIN_RXCAL_LUT to 0 else set them to 1</td> </tr> </table> | Bit | Description | b0 | RETAIN_TXCAL_LUT (Not supported in IWR6843 ES1.0) | 0 | Update TX calibration LUT | 1 | Do not update TX calibration LUT | b1 | RETAIN_RXCAL_LUT (Not supported in IWR6843 ES1.0) | 0 | Update RX calibration LUT and update RX IQMM correction | 1 | Do not update RX calibration LUT | b7:2 | RESERVED (set it to 0b000000) | If PF_TX_OUTPUT_POWER_BACKOFF is changed then set RETAIN_TXCAL_LUT to 0, else set it to 1 and if PF_RX_GAIN or if sweep bandwidth is changed, then set RETAIN_RXCAL_LUT to 0 else set them to 1 | | | |
| Bit | Description | | | | | | | | | | | | | | | | | | | | | |
| b0 | RETAIN_TXCAL_LUT (Not supported in IWR6843 ES1.0) | | | | | | | | | | | | | | | | | | | | | |
| 0 | Update TX calibration LUT | | | | | | | | | | | | | | | | | | | | | |
| 1 | Do not update TX calibration LUT | | | | | | | | | | | | | | | | | | | | | |
| b1 | RETAIN_RXCAL_LUT (Not supported in IWR6843 ES1.0) | | | | | | | | | | | | | | | | | | | | | |
| 0 | Update RX calibration LUT and update RX IQMM correction | | | | | | | | | | | | | | | | | | | | | |
| 1 | Do not update RX calibration LUT | | | | | | | | | | | | | | | | | | | | | |
| b7:2 | RESERVED (set it to 0b000000) | | | | | | | | | | | | | | | | | | | | | |
| If PF_TX_OUTPUT_POWER_BACKOFF is changed then set RETAIN_TXCAL_LUT to 0, else set it to 1 and if PF_RX_GAIN or if sweep bandwidth is changed, then set RETAIN_RXCAL_LUT to 0 else set them to 1 | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.20 – continued from previous page

| | | | | | | | | | | | | |
|----------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|------|---------------------------|-------|---------------------------|--------|---------------------------|--------|---------------------------|
| PF_FREQ_START_CONST | 4 | <p>Start frequency for this profile</p> <p>For 77GHz Devices (76GHz to 81Ghz): 1 LSB = $3.6e9/2^{26}$ Hz \approx 53.644 Hz Valid range: 0x5471C71B to 0x5A000000</p> <p>For 60GHz Devices (57GHz to 64Ghz): 1 LSB = $2.7e9/2^{26}$ Hz \approx 40.233 Hz Valid range: 0x5471C71C to 0x5ED097B4</p> | | | | | | | | | | |
| PF_IDLE_TIME_CONST | 4 | <p>Idle time for each profile</p> <p>1 LSB = 10 ns Valid range: 0 to 524287</p> | | | | | | | | | | |
| PF_ADC_START_TIME_CONST | 4 | <p>Time of starting of ADC capture relative to the knee of the ramp</p> <p>1 LSB = 10 ns Valid range: 0 to 4095</p> | | | | | | | | | | |
| PF_RAMP_END_TIME | 4 | <p>End of ramp time relative to the knee of the ramp</p> <p>1 LSB = 10 ns Valid range: 0 to 500000</p> <p>Ensure that the total frequency sweep is either within these ranges: 77G: 76-78 GHz or 77-81 GHz 60G: 57-61GHz or 60-64GHz</p> | | | | | | | | | | |
| PF_TX_OUTPUT_POWER_BACKOFF | 4 | <table border="0"> <tr> <td>Bits</td> <td>Description</td> </tr> <tr> <td>b7:0</td> <td>TX0 output power back off</td> </tr> <tr> <td>b15:8</td> <td>TX1 output power back off</td> </tr> <tr> <td>b23:16</td> <td>TX2 output power back off</td> </tr> <tr> <td>b31:24</td> <td>RESERVED (set it to 0x00)</td> </tr> </table> <p>This field defines how much the transmit power should be reduced from the maximum.</p> <p>1 LSB = 1 dB</p> <p>If Tx power boot time calibration is disabled then only 0dB back off is supported.</p> <p>0dB back-off corresponds to typically 12dBm power level. In IWR6843 (both ES1.0 and ES2.0), uncalibrated back-off in 3dB steps is supported in cases where the TX power calibration is disabled.</p> | Bits | Description | b7:0 | TX0 output power back off | b15:8 | TX1 output power back off | b23:16 | TX2 output power back off | b31:24 | RESERVED (set it to 0x00) |
| Bits | Description | | | | | | | | | | | |
| b7:0 | TX0 output power back off | | | | | | | | | | | |
| b15:8 | TX1 output power back off | | | | | | | | | | | |
| b23:16 | TX2 output power back off | | | | | | | | | | | |
| b31:24 | RESERVED (set it to 0x00) | | | | | | | | | | | |

Continued on next page

Table 5.20 – continued from previous page

| Field Name | Width | Bits | Description |
|---------------------|-------|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF_TX_PHASE_SHIFTER | 4 | b1:0 b7:2 b9:8 b15:10 b17:16 b23:18 b31:24 | RESERVED (set it to 0b00) TX0 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 \approx 5.625^\circ$ RESERVED (set it to 0b00) TX1 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 \approx 5.625^\circ$ RESERVED (set it to 0b00) TX2 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 \approx 5.625^\circ$ RESERVED 0x00 This field defines the additional phase shift to be introduced on each transmitter output. This option is supported only on selected device variants, Please refer data sheet. NOTE: Chirps corresponding to different profiles are not guaranteed to have phase coherency. In IWR6843 ES1.0, only 0 degree phase is supported. |
| PF_FREQ_SLOPE_CONST | 2 | | Frequency slope for each profile is encoded in 2 bytes (16 bit signed number) For 77GHz Devices (76GHz to 81Ghz): $1 \text{ LSB} = 3.6e9 \times 900 / 2^{26} \text{ Hz} \approx 48.279 \text{ kHz}/\mu\text{s}$ Valid range: xWR1xxx devices: -2072 to 2072 For 60GHz Devices (57GHz to 64Ghz): $1 \text{ LSB} = 2.7e9 \times 900 / 2^{26} \text{ Hz} \approx 36.21 \text{ kHz}/\mu\text{s}$ Valid range: -6905 to 6905 (250MHz/ μs) |
| PF_TX_START_TIME | 2 | | Time of start of transmitter relative to the knee of the ramp $1 \text{ LSB} = 10 \text{ ns}$ Valid range: -4096 to 4095 Positive numbers refer to start of TX after knee of the ramp and negative numbers refer to start of TX before the knee of the ramp |
| PF_NUM_ADC_SAMPLES | 2 | | Number of ADC samples to capture in a chirp for each RX |

Continued on next page

Table 5.20 – continued from previous page

| | | <p>Valid range: 2 to MAX_NUM_SAMPLES, where MAX_NUM_SAMPLES is such that all the enabled RX channels' data fits into 16 kB memory in AWR1243/xWR1443 or 32 kB memory in xWR1642/IWR6843/xWR1843, with each sample consuming 2 bytes for real ADC output case and 4 bytes for complex 1x and complex 2x ADC output cases. For example in AWR1243/ xWR1443 when the ADC buffer size is 16 kB</p> <table border="1"> <thead> <tr> <th>Number of RX chains</th> <th>ADC format</th> <th>MAX_NUM_SAMPLES</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>Complex</td> <td>1024</td> </tr> <tr> <td>4</td> <td>Real</td> <td>2048</td> </tr> <tr> <td>2</td> <td>Complex</td> <td>2048</td> </tr> <tr> <td>2</td> <td>Real</td> <td>4096</td> </tr> </tbody> </table> | Number of RX chains | ADC format | MAX_NUM_SAMPLES | 4 | Complex | 1024 | 4 | Real | 2048 | 2 | Complex | 2048 | 2 | Real | 4096 |
|---------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|------------|-----------------|---|---------|------|---|------|------|---|---------|------|---|------|------|
| Number of RX chains | ADC format | MAX_NUM_SAMPLES | | | | | | | | | | | | | | | |
| 4 | Complex | 1024 | | | | | | | | | | | | | | | |
| 4 | Real | 2048 | | | | | | | | | | | | | | | |
| 2 | Complex | 2048 | | | | | | | | | | | | | | | |
| 2 | Real | 4096 | | | | | | | | | | | | | | | |
| PF_DIGITAL_OUTPUT_SAMPLING_RATE | 2 | <p>ADC Sampling rate for this profile is encoded in 2 bytes (16 bit unsigned number) 1 LSB = 1 ksp/s Valid range: xWR1xxx and IWR6843 devices: 2000 to 37500 (Max 15MHz IF bandwidth)</p> | | | | | | | | | | | | | | | |
| PF_HPF1_CORNER_FREQ | 1 | <p>HPF1 corner frequency for each profile is encoded in 1 byte Value HPF1 corner frequency definition 0x00 175 kHz 0x01 235 kHz 0x02 350 kHz 0x03 700 kHz</p> | | | | | | | | | | | | | | | |
| PF_HPF2_CORNER_FREQ | 1 | <p>HPF2 corner frequency for each profile is encoded in 1 byte Value HPF2 corner frequency definition 0x00 350 kHz 0x01 700 kHz 0x02 1.4 MHz 0x03 2.8 MHz</p> | | | | | | | | | | | | | | | |

Continued on next page

Table 5.20 – continued from previous page

| | | |
|---------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX_CAL_EN_CFG | 2 | <p>Number of transmitters to turn on during TX power calibration. During actual operation, if more than 1 TXs are enabled during the chirp, then enabling the same TXs during calibration will have better TX output power accuracy</p> <p>Bit Definition</p> <p>b2:0 TX enabled during TX0 calibration b0 - TX0, b1 - TX1, b2 - TX2</p> <p>b5:3 TX enabled during TX1 calibration b3 - TX0, b4 - TX1, b5 - TX2</p> <p>b8:6 TX enabled during TX2 calibration b6 - TX0, b7 - TX1, b8 - TX2</p> <p>b14:9 RESERVED</p> <p>b15 Enable multi TX enable during TX power calibration If this bit is not set, only 1 TX is enabled during the TX power calibration. For e.g. during TX0 calibration, only TX0 will be enabled; during TX1 calibration, only TX1 will be enabled and so on</p> |
| PF_RX_GAIN | 2 | <p>Bit Definition</p> <p>b5:0 RX_GAIN This field defines RX gain for each profile. 1 LSB = 1 dB Valid values: all even values from 24 to 52</p> <p>b7:6 RF_GAIN_TARGET (NA for IWR6843 ES1.0, RF gain is fixed to 34 dB)</p> <p>Value RF gain target</p> <p>00 30 dB</p> <p>01 34 dB</p> <p>10 RESERVED</p> <p>11 26 dB</p> <p>b15:8 RESERVED (set it to 0x00)</p> |

Continued on next page

Table 5.20 – continued from previous page

| | | |
|----------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>The total RX gain is achieved as a sum of RF gain and IF amplifiers gain. The RF Gain target (30 dB, 34 dB and 26 dB) allows the user to control the RF gain independently from the total RX gain, thus giving flexibility to the user to trade-off linearity vs. noise figure. Out of multiple gain settings for the RF stages, the firmware calibration algorithm uses the one that makes the RF gain as close as possible to the user programmed RF Gain Target.</p> <p>In IWR6843 ES1.0 device, Rx Gain is not accurate and not calibrated. The RF gain is 34dB and total Rx gain can be varied from 28dB to 56dB</p> |
| RESERVED | 2 | 0x0000 |

NOTE:

In 1st generation devices (AWr1243, xWR1443, xWR1642, xWR1843 and IWR6843), the minimum duration (idle time + ramp end time) of a chirp can not be less than 15us (normal mode of operation) + 10us (if RadarSS WDT is enabled) + 10us (if per chirp phase-shifter API is enabled). The best case minimum chirp cycle time is 15us and worst case minimum chirp cycle time is 35us.

Table 5.21: Note on maximum sampling rate

NOTE:

The maximum sampling rate supported is limited based on the information in the table below

When device supports 15 MHz IF bandwidth (refer device data sheet)

| | Real/Pseudo Real | Complex1x | Complex2x |
|--------------------|---------------------|------------|------------|
| Regular ADC mode | 37.5 Msps | 18.75 Msps | 37.5 Msps |
| Low power ADC mode | 18.75 Msps | 9.375 Msps | 18.75 Msps |

When device supports 10 MHz IF bandwidth (refer device data sheet)

| | Real/Pseudo Real | Complex1x | Complex2x |
|--------------------|---------------------|------------|------------|
| Regular ADC mode | 25 Msps | 12.5 Msps | 25 Msps |
| Low power ADC mode | 18.75 Msps | 9.375 Msps | 18.75 Msps |

When device supports 5 MHz IF bandwidth (refer device data sheet)

| | Real/Pseudo Real | Complex1x | Complex2x |
|--------------------|---------------------|-----------|-----------|
| Regular ADC mode | 12.5 Msps | 6.25 Msps | 12.5 Msps |
| Low power ADC mode | 12.5 Msps | 6.25 Msps | 12.5 Msps |

- The IF bandwidth here refers to the IF frequency of the farthest reflection desired to be detected
- Typically, the IF frequency range preserved well in the receiver baseband is $0.9 \times$ Sampling Rate in Complex 1x and $0.45 \times$ Sampling Rate in Complex 2x and Real/Pseudo Real.
- The maximum sampling rates are also subject to restrictions from LVDS/CSI2 interface rate and ADC bits configurations. Typically in Complex2x mode, the maximum sampling rate would be 25 Msps

5.5.2 Sub block 0x0101 – AWR_CHIRP_CONF_SET_SB

This sub block contains chirp to chirp variations on top of the chirp profiles defined in the AWR_PROFILE_CONF_SET_SB. E.g. which profile is to be used for each chirp in a frame, and small dithers in FMCW start frequency and idle time for each chirp are possible to be defined here. The dithers used in this configuration sub block are only additive on top of programmed parameters in AWR_PROFILE_CONF_SET_SB.

Table 5.22 describes the contents of this sub block.

Table 5.22: AWR_CHIRP_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|--------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0101 |
| SBLKLEN | 2 | Value = 24 |
| CHIRP_START_IND_X | 2 | Valid range 0 to 511 |
| CHIRP_END_IND_X | 2 | Valid range CHIRP_START_IND_X to 511 |
| PROFILE_IND_X | 2 | Valid range 0 to 3 |
| RESERVED | 2 | 0x0000 |
| CHIRP_FREQ_START_VAR | 4 | For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9/2^{26} \approx 53.644$ Hz Valid range: 0 to 8388607 (450MHz) For 60GHz Devices (57GHz to 64Ghz): 1 LSB = $2.7e9/2^{26} \approx 40.23$ Hz Valid range: 0 to 8388607 (337.5MHz) |
| CHIRP_FREQ_SLOPE_VAR | 2 | For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9 \times 900/2^{26} \approx 48.279$ kHz Valid range: 0 to 63 (3MHz/us) For 60GHz Devices (57GHz to 64Ghz): 1 LSB = $2.7e9 \times 900/2^{26} \approx 36.21$ Hz Valid range: 0 to 63 (2.3MHz/us) |
| CHIRP_IDLE_TIME_VAR | 2 | Idle time of each chirp is encoded in 2 bytes (16 bit unsigned number) 1 LSB = 10 ns Valid range: 0 to 4095 |
| CHIRP_ADC_START_TIME_VAR | 2 | ADC start time of each chirp is encoded in 2 bytes (16 bit unsigned number) 1 LSB = 10 ns Valid range: 0 to 4095 |

Continued on next page

Table 5.22 – continued from previous page

| | | | | | | | | | | | | |
|--------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|----|------------|----|------------|----|------------|-------|----------|
| CHIRP_TX_EN | 2 | TX enable selection | | | | | | | | | | |
| | | <table border="0"> <tr><td>Bit</td><td>Definition</td></tr> <tr><td>b0</td><td>TX0 Enable</td></tr> <tr><td>b1</td><td>TX1 Enable</td></tr> <tr><td>b2</td><td>TX2 Enable</td></tr> <tr><td>b15:3</td><td>RESERVED</td></tr> <tr><td colspan="2">0b0_0000_0000_0000</td></tr> </table> <p>NOTE: Maximum number of TXs that can be turned on in a chirp depends on the device data sheet specification</p> | Bit | Definition | b0 | TX0 Enable | b1 | TX1 Enable | b2 | TX2 Enable | b15:3 | RESERVED |
| Bit | Definition | | | | | | | | | | | |
| b0 | TX0 Enable | | | | | | | | | | | |
| b1 | TX1 Enable | | | | | | | | | | | |
| b2 | TX2 Enable | | | | | | | | | | | |
| b15:3 | RESERVED | | | | | | | | | | | |
| 0b0_0000_0000_0000 | | | | | | | | | | | | |

5.5.3 Sub block 0x0102 – AWR_FRAME_CONF_SET_SB

This sub block defines a frame, i.e. a sequence of chirps to be transmitted subsequently, the no. of frames to be transmitted, frame periodicity and how to trigger them.

Table 5.23 describes the contents of this sub block.

Table 5.23: AWR_FRAME_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|--------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0102 |
| SBLKLEN | 2 | Value = 28 |
| RESERVED | 2 | May use to indicate Frame mode or Continuous chirping mode of operation. |
| CHIRP_START_IND_X | 2 | Valid range 0 to 511 |
| CHIRP_END_IND_X | 2 | Valid range CHIRP_START_IND_X to 511 |
| NUM_LOOPS | 2 | Number of times to repeat from CHIRP_START_IND_X to CHIRP_END_IND_X in each frame Valid range 1 to 255 |
| NUM_FRAMES | 2 | Number of frames to transmit 16 bit unsigned number Valid range: 0 to 65535 (0 for infinite frames) |
| RESERVED | 2 | 0x0000 |
| FRAME_PERIODICITY | 4 | 1 LSB = 5 ns Valid range: 40000 + (1 chirp duration) to 268400000 This is the frame repetition period. |

Continued on next page

Table 5.23 – continued from previous page

| | | Value | Definition |
|--------------------------------|---|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TRIGGER_SELECT | 2 | 0x0001 | SWTRIGGER (Software API based triggering): Frame is triggered upon receiving AWR_FRAMESTARTSTOP_CONF_SB. There could be several tens of micro seconds uncertainty in triggering. This mode is not applicable if this device is configured as MULTICHIP_SLAVE in AWR_CHAN_CONF_SB. |
| | | 0x0002 | HWTRIGGER (Hardware SYNC_IN based triggering): Each frame is triggered by rising edge of pulse in SYNC_IN pin, after receiving AWR_FRAMESTARTSTOP_CONF_SB (this is to prevent spurious transmission). W.r.t. the SYNC_IN pulse, the actual transmission has 5ns uncertainty in SINGLECHIP and only a 300 ps uncertainty (due to tight inter-chip synchronization needed) in MULTICHIP sensor applications as defined in AWR_CHAN_CONF_SB. |
| RESERVED | 1 | 0x00 | |
| NUM_OF_DUMMY_CHIRPS_AT_THE_END | 1 | | Number of dummy chirps which will run in the rampgen at the end of the frame. The ADC data, CP and CQ data for these chirps will not be sent out from the device/DFE. 1LSB = 1 dummy chirp in the end of frame Valid Range: 0 to 128 (should be less than the total number of chirps in a frame) |
| FRAME_TRIGGER_DELAY | 4 | | Optional time delay from the SYNC_IN trigger to the occurrence of frame chirps. Applicable only in SINGLECHIP sensor applications, as defined in AWR_CHAN_CONF_SB. It is recommended only for staggering the transmission of multiple radar sensors around the car for interference avoidance, if needed. Typical range is 0 to 100 micro seconds. Units: 1 LSB = 5 ns |

- NOTE1:** If hardware triggered mode is used, the SYNC_IN pulse width should be less than the ON time of the frame (in case of legacy frame config mode) or the ON time of the burst (in case of advanced frame config mode). Also, the minimum pulse width of SYNC_IN should be 25 ns.
- NOTE2:** If frame trigger delay is used with hardware triggered mode, then external SYNC_IN pulse periodicity should take care of the configured frame trigger delay and frame periodicity. The external pulse should be issued only after the sum total of frame trigger delay and frame periodicity. See figure below
- NOTE3:** If dummy chirp is used then programmer should make sure the idle time of dummy chirp $\geq 4\mu\text{s} + \text{DFE spill over time of previous chirp}$ (calculate from rampgen calculator). The first chirp of frame can not be a dummy chirp.
- NOTE4:** In Hw triggered mode, the Hw pulse should be issued or periodicity of pulse is configured such that, the pulse is generated only 150us after the completion of previous frame/burst (The pulse should not be issued before end of previous frame/burst). The time delta between end of previous frame/burst and raising edge of Hw pulse recommended to be $< 300\mu\text{s}$.

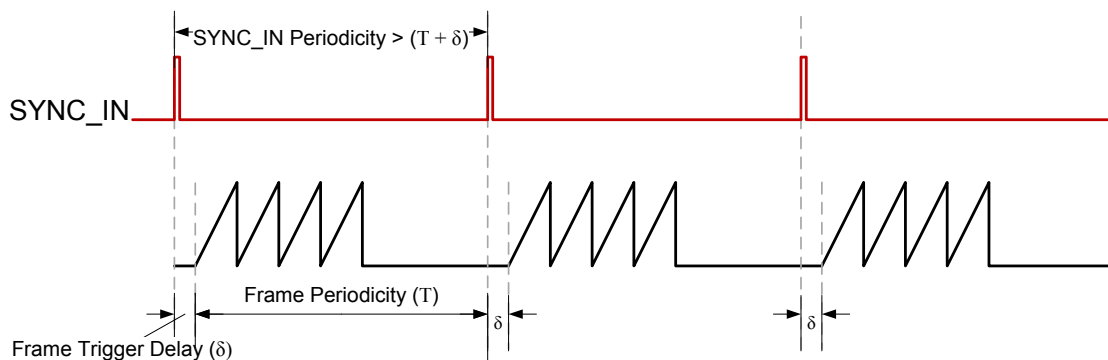


Figure 5.1: Frame trigger delay in case of external hardware trigger

NOTE: The inter-frame blank time should be at least 250 μ s. (100 μ s for frame preparation and 150 μ s for any calibration updates to hardware or to perform monitoring)
 Add below extra time to inter-frame blank time if below feature is enabled:

1. 150 μ s if test source is enabled.
2. 250 μ s if loop-back burst is enabled (Applicable only for advance frame configuration API)
3. 200 μ s if dynamic chirp or dynamic per-chirp phase shifter configuration API is invoked in runtime while framing.
4. 400 μ s if dynamic profile configuration API is invoked in runtime while framing.
5. 500 μ s if Rx saturation and signal image API is invoked in runtime for reconfigurations while framing every FTTI boundary

5.5.4 Sub block 0x0103 – AWR_CONT_STREAMING_MODE_CONF_SET_SB

This sub block contains configuration needed to enable continuous streaming mode from the device.

NOTE: The continuous streaming mode configuration APIs are supported only for debug purpose. Please refer latest DFP release note for more info.

Table 5.24 describes the contents of this sub block.

Table 5.24: AWR_CONT_STREAMING_MODE_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|---------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0103 |
| SBLKLEN | 2 | Value = 24 |
| PF_FREQ_START_CONST | 4 | Frequency start for each profile is encoded in 4 bytes (32 bit unsigned number) For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9/2^{26}$ Hz \approx 53.644 Hz Valid range: 0 to 0x7FFFFFFF For 60GHz Devices (57GHz to 64GHz): 1 LSB = $2.7e9/2^{26}$ Hz \approx 40.23 Hz Valid range: 0 to 0x7FFFFFFF |

Continued on next page

Table 5.24 – continued from previous page

| PF_TX_OUTPUT_POWER_BACKOFF | 4 | <p>Bits Description</p> <p>b7:0 TX0 output power back off</p> <p>b15:8 TX1 output power back off</p> <p>b23:16 TX2 output power back off</p> <p>b31:24 RESERVED (set it to 0x00)</p> <p>This field defines how much the transmit power should be reduced from the maximum. 1 LSB = 1 dB</p> <p>In IWR6843 ES1.0, Tx backoff is supported in 3dB step (uncalibrated).</p> | | | | | | | | | | |
|---------------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------------------------------|------|---------|------|---------|------|---------|------|---------|
| PF_TX_PHASE_SHIFTER | 4 | <p>Bits Description</p> <p>b1:0 RESERVED (set it to 0b00)</p> <p>b7:2 TX0 phase shift value 1 LSB = $360^\circ/2^6 \approx 5.625^\circ$</p> <p>b9:8 RESERVED (set it to 0b00)</p> <p>b15:10 TX1 phase shift value 1 LSB = $360^\circ/2^6 \approx 5.625^\circ$</p> <p>b17:16 RESERVED (set it to 0b00)</p> <p>b23:18 TX2 phase shift value 1 LSB = $360^\circ/2^6 \approx 5.625^\circ$</p> <p>b31:24 RESERVED 0x00</p> <p>This field defines the additional phase shift to be introduced on each transmitter output.</p> <p>In IWR6843 ES1.0, only 0 degree phase is supported.</p> | | | | | | | | | | |
| PF_DIGITAL_OUTPUT_SAMPLING_RATE | 2 | <p>ADC Sampling rate for each profile is encoded in 2 bytes (16 bit unsigned number)</p> <p>1 LSB = 1 ksps</p> <p>Valid range 2000 to 37500</p> | | | | | | | | | | |
| PF_HPF1_CORNER_FREQ | 1 | <p>HPF1 corner frequency for each profile is encoded in 1 byte</p> <table data-bbox="662 1534 1117 1731"> <thead> <tr> <th>Value</th> <th>HPF1 corner frequency definition</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>175 kHz</td> </tr> <tr> <td>0x01</td> <td>235 kHz</td> </tr> <tr> <td>0x02</td> <td>350 kHz</td> </tr> <tr> <td>0x03</td> <td>700 kHz</td> </tr> </tbody> </table> | Value | HPF1 corner frequency definition | 0x00 | 175 kHz | 0x01 | 235 kHz | 0x02 | 350 kHz | 0x03 | 700 kHz |
| Value | HPF1 corner frequency definition | | | | | | | | | | | |
| 0x00 | 175 kHz | | | | | | | | | | | |
| 0x01 | 235 kHz | | | | | | | | | | | |
| 0x02 | 350 kHz | | | | | | | | | | | |
| 0x03 | 700 kHz | | | | | | | | | | | |

Continued on next page

Table 5.24 – continued from previous page

| | | | | | | | | | | | | | | | | | | |
|---------------------|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------------------------------|------|------------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------|-------|----------------|------|---------|----|-------|----|----------|----|-------|
| PF_HPF2_CORNER_FREQ | 1 | <p>HPF2 corner frequency for each profile is encoded in 1 byte</p> <table border="0"> <tr> <td>Value</td> <td>HPF2 corner frequency definition</td> </tr> <tr> <td>0x00</td> <td>350 kHz</td> </tr> <tr> <td>0x01</td> <td>700 kHz</td> </tr> <tr> <td>0x02</td> <td>1.4 MHz</td> </tr> <tr> <td>0x03</td> <td>2.8 MHz</td> </tr> </table> | Value | HPF2 corner frequency definition | 0x00 | 350 kHz | 0x01 | 700 kHz | 0x02 | 1.4 MHz | 0x03 | 2.8 MHz | | | | | | |
| Value | HPF2 corner frequency definition | | | | | | | | | | | | | | | | | |
| 0x00 | 350 kHz | | | | | | | | | | | | | | | | | |
| 0x01 | 700 kHz | | | | | | | | | | | | | | | | | |
| 0x02 | 1.4 MHz | | | | | | | | | | | | | | | | | |
| 0x03 | 2.8 MHz | | | | | | | | | | | | | | | | | |
| PF_RX_GAIN | 1 | <p>This field defines RX gain for continuous streaming mode.</p> <table border="0"> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>5:0</td> <td> RX_GAIN This field defines RX gain for each profile. 1 LSB = 1 dB Valid values: all even values from 24 to 52 </td> </tr> <tr> <td>7:6</td> <td> RF_GAIN_TARGET (NA for IWR6843 ES1.0, In IWR6843 ES1.0 RF gain is fixed to 36 dB) </td> </tr> <tr> <td>Value</td> <td>RF gain target</td> </tr> <tr> <td>00</td> <td>30 dB</td> </tr> <tr> <td>01</td> <td>34 dB</td> </tr> <tr> <td>10</td> <td>RESERVED</td> </tr> <tr> <td>11</td> <td>26 dB</td> </tr> </table> <p>The total RX gain is achieved as a sum of RF gain and IF amplifiers gain. The RF Gain target (30 dB, 34 dB and 26 dB) allows the user to control the RF gain independently from the total RX gain, thus giving flexibility to the user to trade-off linearity vs. noise figure. Out of multiple gain settings for the RF stages, the firmware calibration algorithm uses the one that makes the RF gain as close as possible to the user programmed RF Gain Target.</p> <p>In IWR6843 ES1.0 device, Rx Gain is not accurate and not calibrated. The RF gain is 36dB and total Rx gain can be varied from 30dB to 52dB</p> | Bit | Definition | 5:0 | RX_GAIN This field defines RX gain for each profile. 1 LSB = 1 dB Valid values: all even values from 24 to 52 | 7:6 | RF_GAIN_TARGET (NA for IWR6843 ES1.0, In IWR6843 ES1.0 RF gain is fixed to 36 dB) | Value | RF gain target | 00 | 30 dB | 01 | 34 dB | 10 | RESERVED | 11 | 26 dB |
| Bit | Definition | | | | | | | | | | | | | | | | | |
| 5:0 | RX_GAIN This field defines RX gain for each profile. 1 LSB = 1 dB Valid values: all even values from 24 to 52 | | | | | | | | | | | | | | | | | |
| 7:6 | RF_GAIN_TARGET (NA for IWR6843 ES1.0, In IWR6843 ES1.0 RF gain is fixed to 36 dB) | | | | | | | | | | | | | | | | | |
| Value | RF gain target | | | | | | | | | | | | | | | | | |
| 00 | 30 dB | | | | | | | | | | | | | | | | | |
| 01 | 34 dB | | | | | | | | | | | | | | | | | |
| 10 | RESERVED | | | | | | | | | | | | | | | | | |
| 11 | 26 dB | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.24 – continued from previous page

| | | | |
|------------|---|----------|-----------------------------------------------------------------------------------------------------|
| VCO_SELECT | 1 | Bit | Description |
| | | b0 | FORCE_VCO_SEL 0 Use internal VCO selection 1 Forced external VCO selection |
| | | b1 | VCO_SEL 0 VCO1 (77G: 76 – 78 GHz, 60G: 57 - 61GHz) 1 VCO2 (77G: 77 – 81 GHz, 60G: 60 – 64GHz) |
| | | b7:2 | RESERVED 0b00_0000 |
| | | RESERVED | 2 |

NOTE: Continuous streaming (CW) mode is useful for RF lab characterization and debug. In this mode, the device is configured to transmit a single continuous wave (CW - 0 slope) tone at a specific RF frequency continuously.

5.5.5 Sub block 0x0104 – AWR_CONT_STREAMING_MODE_EN_SB

This sub block contains configuration needed to enable continuous streaming mode from the device.

Table 5.25 describes the contents of this sub block.

Table 5.25: AWR_CONT_STREAMING_MODE_EN_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|---------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0104 |
| SBLKLEN | 2 | Value = 8 |
| CONT_STREAMING_EN | 2 | Value Definition 0x0000 Disable continuous streaming mode 0x0001 Enable continuous streaming mode |
| RESERVED | 2 | 0x0000 |

5.5.6 Sub block 0x0105 – AWR_ADVANCED_FRAME_CONF_SB

This sub block contains advanced frame configuration options.

Table 5.26 describes the contents of this sub block.

Table 5.26: AWR_ADVANCED_FRAME_CONF_SB contents

| Field Name | Number of bytes | Description |
|---------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0105 |
| SBLKLEN | 2 | Value = 152 |
| NUM.SUB-FRAMES | 1 | Number of sub frames enabled in this frame Valid range: 1 to 4 |
| FORCE.SINGLE.PROFILE | 1 | Value Definition 0x0 The profile index set in Chirp Config API message governs which profile is used when that chirp is transmitted 0x1 The profile index indicated in Chirp Config message is ignored and all the chirps in each sub frame use a single profile as indicated by that sub frame's profile index set in this message. |
| LOOPBACK.CFG | 1 | Bit Definition b0 LOOPBACK_CFG_EN 0 Disable 1 Enable b2:1 SUB_FRAME_ID Sub frame ID for which the loopback configuration applies b7:3 RESERVED |
| SUB.FRAME.TRIGGER | 1 | 0 Disabled (default) 1 Enabled (Need to trigger each sub-frame either by SW in software triggered mode or HW in hardware triggered mode) |
| SF1.PROFILE.INDX | 2 | This is applicable only if FORCE_SINGLE_PROFILE is set to 1. Please refer to that field for description. Valid range: 0 to 3 Not applicable for loop-back sub-frame |
| SF1.CHIRP.START.INDX | 2 | Start index of the first chirp for the first burst in sub frame 1 Valid range: 0 to 511 Not applicable for loop-back sub-frame |
| SF1.NUM.UNIQUE.CHIRPS.PER.BURST | 2 | Number of unique chirps per burst Valid range: 1 to 512 Not applicable for loop-back sub-frame |
| SF1.NUM.LOOPS.PER.BURST | 2 | Number of times to loop through the unique chirps in each burst, without gaps, using HW. Valid range: 1 to 255 |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|-----------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF1_BURST_PERIOD | 4 | $BURST_PERIOD \geq (NUM_LOOPS_PER_BURST \times (\text{Sum total of all unique chirp times per burst}) + InterBurstBlankTime)$, where <i>InterBurstBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, and some minimum time needed for triggering next burst. $InterBurstBlankTime \geq 250 \mu s$ With loopback enabled, $InterBurstBlankTime \geq 500 \mu s$ NOTE: Across bursts, if the value (Sum total of all unique chirp times per burst), is not a constant, then the actual available blank time can vary and needs to be accounted for. 1 LSB = 5 ns Valid range: 100 μs to 1.342 s |
| SF1_CHIRP_START_INDX_OFFSET | 2 | The chirp start index for each burst is determined as the chirp start index of the previous burst plus SFx_START_INDX_OFFSET \times BURST_INDX i.e. $CHIRP_START_INDX = SFx_CHIRP_START_INDX + (SFx_CHIRP_START_INDX_OFFSET \times BURST_INDEX)$ Valid range: 0 to 511 A value of 0 can be used to repeat the same set of unique chirps across bursts. Non-zero values allow spanning a larger number of unique chirps (across bursts). Not applicable for loop-back sub-frame |
| SF1_NUM_BURSTS | 2 | Number of bursts constituting this sub frame Valid range: 1 to 512 Not applicable for loop-back sub-frame |
| SF1_NUM_OUTER_LOOPS | 2 | Number of times to loop over the set of above defined bursts, for this sub frame. Valid range: 1 to 64 Not applicable for loop-back sub-frame |
| RESERVED | 2 | 0x0000 |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|---------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF1_PERIOD | 4 | $PERIOD \geq \text{Sum total time of all bursts} + \text{InterSubFrameBlankTime}$, where, $\text{Sum total time of all bursts} = \text{Num Outer Loops} * \text{Num Bursts} * \text{Burst Period}$. <i>InterSubFrameBlankTime</i> is primarily for sensor calibration/monitoring, thermal control, transferring out any safety monitoring data if requested, hardware reconfiguration for next sub frame, retriggering of next SF. $\text{InterSubFrameBlankTime} \geq 250 \mu\text{s}$ With loopback enabled, $\text{InterSubFrameBlankTime} \geq 500 \mu\text{s}$ Add $150 \mu\text{s}$ to <i>InterSubFrameBlankTime</i> for test source configuration if test source is enabled. 1 LSB = 5 ns Valid range 100 μs to 1.342 s |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| SF2_PROFILE_INDXX | 2 | This is applicable only if FORCE_SINGLE_PROFILE is set to 1. Please refer to that field for description. Valid range: 0 to 3 |
| SF2_CHIRP_START_INDXX | 2 | Start index of the first chirp for the first burst in sub frame 2 Valid range: 0 to 511 |
| SF2_NUM_UNIQUE_CHIRPS_PER_BURST | 2 | Number of unique chirps per burst Valid range: 1 to 512 |
| SF2_NUM_LOOPS_PER_BURST | 2 | Number of times to loop through the unique chirps in each burst, without gaps, using HW. Valid range: 1 to 255 |
| SF2_BURST_PERIOD | 4 | $BURST_PERIOD \geq (\text{NUM_LOOPS_PER_BURST}) * (\text{Sum total of all unique chirp times per burst}) + \text{InterBurstBlankTime}$, where <i>InterBurstBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, and some minimum time needed for triggering next burst. $\text{InterBurstBlankTime} \geq 250 \mu\text{s}$ With loopback enabled, $\text{InterBurstBlankTime} \geq 500 \mu\text{s}$ NOTE: Across bursts, if the value (Sum total of all unique chirp times per burst), is not a constant, then the actual available blank time can vary and needs to be accounted for. 1 LSB = 5 ns |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|---------------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF2_CHIRP_START_INDX_OFFSET | 2 | The chirp start index for each burst is determined as the chirp start index of the previous burst plus SFx.START_INDX_OFFSET * BURST_INDX i.e. $CHIRP_START_INDX = SFx_CHIRP_START_INDX + (SFx_CHIRP_START_INDX_OFFSET \times BURST_INDEX)$ Valid range: 0 to 511 A value of 0 can be used to repeat the same set of unique chirps across bursts. Non-zero values allow spanning a larger number of unique chirps (across bursts). |
| SF2_NUM_BURSTS | 2 | Number of bursts constituting this sub frame Valid range: 1 to 512 |
| SF2_NUM_OUTER_LOOPS | 2 | Number of times to loop over the set of above defined bursts, for this sub frame. Valid range: 1 to 64 |
| RESERVED | 2 | 0x0000 |
| SF2_PERIOD | 4 | $PERIOD \geq \text{Sum total time of all bursts} + \text{InterSubFrameBlankTime}$, Where, Sum total time of all bursts = Num Outer Loops * Num Bursts * Burst Period. <i>InterSubFrameBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, transferring out any safety monitoring data if requested, hardware reconfiguration for next sub frame, retriggering of next SF. $InterSubFrameBlankTime \geq 250 \mu s$ With loopback enabled, $InterSubFrameBlankTime \geq 500 \mu s$ Add 150 μs to <i>InterSubFrameBlankTime</i> for test source configuration if test source is enabled. 1 LSB = 5 ns Valid range: 100 μs to 1.342 s |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| SF3_PROFILE_INDX | 2 | This is applicable only if FORCE_SINGLE_PROFILE is set to 1. Please refer to that field for description. Valid range: 0 to 3 |
| SF3_CHIRP_START_INDX | 2 | Start index of the first chirp in this sub frame Valid range: 0 to 511 |
| SF3_NUM_UNIQUE_CHIRPS_PER_BURST | 2 | Number of unique chirps per burst Valid range: 1 to 512 |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|-----------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF3_NUM_LOOPS_PER_BURST | 2 | Number of times to loop through the unique chirps in each burst, without gaps, using HW. Valid range: 1 to 255 |
| SF3_BURST_PERIOD | 4 | $BURST_PERIOD \geq (NUM_LOOPS_PER_BURST) * (\text{Sum total of all unique chirp times per burst}) + InterBurstBlankTime$, where <i>InterBurstBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, and some minimum time needed for triggering next burst. $InterBurstBlankTime \geq 250 \mu s$ With loopback enabled, $InterBurstBlankTime \geq 500 \mu s$ NOTE: Across bursts, if the value (Sum total of all unique chirp times per burst), is not a constant, then the actual available blank time can vary and needs to be accounted for. 1 LSB = 5 ns |
| SF3_CHIRP_START_INDX_OFFSET | 2 | The chirp start index for each burst is determined as the chirp start index of the previous burst plus SFx_START_INDX_OFFSET * BURST_INDX i.e. $CHIRP_START_INDX = SFx_CHIRP_START_INDX + (SFx_CHIRP_START_INDX_OFFSET \times BURST_INDEX)$ Valid range: 0 to 511 A value of 0 can be used to repeat the same set of unique chirps across bursts. Non-zero values allow spanning a larger number of unique chirps (across bursts). |
| SF3_NUM_BURSTS | 2 | Number of bursts constituting this sub frame Valid range: 1 to 512 |
| SF3_NUM_OUTER_LOOPS | 2 | Number of times to loop over the set of above defined bursts, for this sub frame. Valid range: 1 to 64 |
| RESERVED | 2 | 0x0000 |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|---------------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF3_PERIOD | 4 | <p>PERIOD \geq Sum total time of all bursts + <i>InterSubFrameBlankTime</i>, Where, Sum total time of all bursts = Num Outer Loops * Num Bursts * Burst Period. <i>InterSubFrameBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, transferring out any safety monitoring data if requested, hardware reconfiguration for next sub frame, retriggering of next SF. <i>InterSubFrameBlankTime</i> \geq 250 μs With loopback enabled, <i>InterSubFrameBlankTime</i> \geq 500 μs</p> <p>Add 150 μs to <i>InterSubFrameBlankTime</i> for test source configuration if test source is enabled. 1 LSB = 5 ns Valid range: 100 μs to 1.342 s</p> |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| SF4_PROFILE_INDXX | 2 | <p>This is applicable only if FORCE_SINGLE_PROFILE is set to 1. Please refer to that field for description. Valid range: 0 to 3</p> |
| SF4_CHIRP_START_INDXX | 2 | <p>Start index of the first chirp in this sub frame Valid range: 0 to 511</p> |
| SF4_NUM_UNIQUE_CHIRPS_PER_BURST | 2 | <p>Number of unique chirps per burst Valid range: 1 to 512</p> |
| SF4_NUM_LOOPS_PER_BURST | 2 | <p>Number of times to loop through the unique chirps in each burst, without gaps, using HW. Valid range: 1 to 255</p> |
| SF4_BURST_PERIOD | 4 | <p>BURST_PERIOD \geq (NUM_LOOPS_PER_BURST) * (Sum total of all unique chirp times per burst) + <i>InterBurstBlankTime</i>, where <i>InterBurstBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, and some minimum time needed for triggering next burst. <i>InterBurstBlankTime</i> \geq 250 μs With loopback enabled, <i>InterBurstBlankTime</i> \geq 500 μs NOTE: Across bursts, if the value (Sum total of all unique chirp times per burst), is not a constant, then the actual available blank time can vary and needs to be accounted for. 1 LSB = 5 ns</p> |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|-----------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF4_CHIRP_START_INDX_OFFSET | 2 | The chirp start index for each burst is determined as the chirp start index of the previous burst plus SFx.START_INDX_OFFSET * BURST_INDX i.e. $CHIRP_START_INDX = SFx_CHIRP_START_INDX + (SFx_CHIRP_START_INDX_OFFSET \times BURST_INDEX)$ Valid range: 0 to 511 A value of 0 can be used to repeat the same set of unique chirps across bursts. Non-zero values allow spanning a larger number of unique chirps (across bursts). |
| SF4_NUM_BURSTS | 2 | Number of bursts constituting this sub frame Valid range: 1 to 512 |
| SF4_NUM_OUTER_LOOPS | 2 | Number of times to loop over the set of above defined bursts, for this sub frame. Valid range: 1 to 64 |
| RESERVED | 2 | 0x0000 |
| SF4_PERIOD | 4 | $SF_PERIOD \geq \text{Sum total time of all bursts} + \text{InterSubFrameBlankTime}$, where, $\text{Sum total time of all bursts} = \text{Num Outer Loops} * \text{Num Bursts} * \text{Burst Period}$. <i>InterSubFrameBlankTime</i> is primarily for sensor calibration / monitoring, thermal control, transferring out any safety monitoring data if requested, hardware reconfiguration for next sub frame, retriggering of next SF. $InterSubFrameBlankTime \geq 250 \mu s$ With loopback enabled, $InterSubFrameBlankTime \geq 500 \mu s$ Add $150 \mu s$ to <i>InterSubFrameBlankTime</i> for test source configuration if test source is enabled. 1 LSB = 5 ns Valid range: 100 μs to 1.342 s |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| NUM_FRAMES | 2 | Number of frames to transmit (1 frame = all enabled sub frames). If set to 0, frames are transmitted endlessly till Frame Stop message is received. Valid range: 0 to 65535 |

Continued on next page

Table 5.26 – continued from previous page

| | | |
|---------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TRIGGER_SELECT | 2 | <p>0x0001 SWTRIGGER (Software API based triggering): Frame is triggered upon receiving AWR_FRAMESTARTSTOP_CONF_SB. There could be several tens of micro seconds uncertainty in triggering. This mode is not applicable if this device is configured as MULTICHIP_SLAVE in AWR_CHAN_CONF_SB.</p> <p>0x0002 HWTRIGGER (Hardware SYNC_IN based triggering): Each frame is triggered by rising edge of pulse in SYNC_IN pin, after receiving AWR_FRAMESTARTSTOP_CONF_SB (this is to prevent spurious transmission). w.r.t. the SYNC_IN pulse, the actual transmission has 5ns uncertainty in SINGLECHIP and only a 300 ps uncertainty (due to tight inter-chip synchronization needed) in MULTICHIP sensor applications as defined in AWR_CHAN_CONF_SB.</p> |
| FRAME_TRIGGER_DELAY | 4 | <p>Optional time delay from the SYNC_IN trigger to the occurrence of frame chirps. Applicable only in SINGLECHIP sensor applications, as defined in AWR_CHAN_CONF_SB. It is recommended only for staggering the transmission of multiple radar sensors around the car for interference avoidance, if needed.</p> <p>Typical range is 0 to few tens of micro seconds. Units: 1 LSB = 5 ns</p> |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

- NOTE1:** If hardware trigger mode is used with SUBFRAMETRIGGER = 0, then the trigger should be issued for each burst. If SUBFRAMETRIGGER = 1, then the trigger needs to be issued for each sub-frame.
- NOTE2:** If hardware triggered mode is used, the SYNC_IN pulse width should be less than the ON time of the frame (in case of legacy frame config mode) or the ON time of the burst (in case of advanced frame config mode). Also, the minimum pulse width of SYNC_IN should be 25 ns.
- NOTE3:** If frame trigger delay is used with hardware triggered mode, then external SYNC_IN pulse periodicity should take care of the configured frame trigger delay and frame periodicity. The external pulse should be issued only after the sum total of frame trigger delay and frame periodicity. See figure below
- NOTE4:** In Hw triggered mode, the Hw pulse should be issued or periodicity of pulse is configured such that, the pulse is generated only 150us after the completion of previous frame/burst (The pulse should not be issued before end of previous frame/burst). The time delta between end of previous frame/burst and raising edge of Hw pulse recommended to be < 300us.

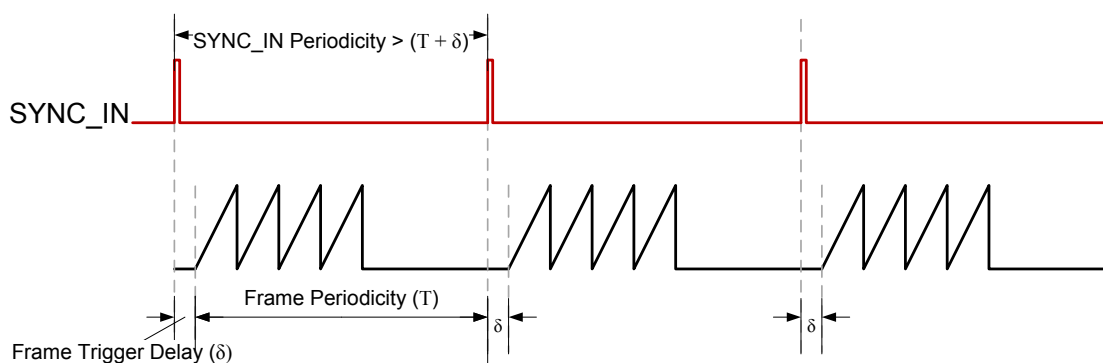


Figure 5.2: Frame trigger delay in case of external hardware trigger

NOTE: The inter-frame blank time should be at least 250 μ s. (100 μ s for frame preparation and 150 μ s for any calibration updates to hardware or to perform monitoring)
 Add below extra time to inter-frame blank time if below feature is enabled:

1. 150 μ s if test source is enabled.
2. 250 μ s if loop-back burst is enabled (Applicable only for advance frame configuration API)
3. 200 μ s if dynamic chirp or dynamic per-chirp phase shifter configuration API is invoked in runtime while framing.
4. 400 μ s if dynamic profile configuration API is invoked in runtime while framing.
5. 500 μ s if Rx saturation and signal image API is invoked in run time for reconfigurations while framing every FTTI boundary

NOTE: The loop-back configuration in this API is not supported in DFP 1.x (1st generation devices). Please refer latest DFP release note for more info.

5.5.7 Sub block 0x0106 – AWR_PERCHIRPPHASESHIFT_CONF_SB

This sub block defines static phase shift configurations per chirp in each of the TXs. The API is applicable only in certain devices (Please refer data sheet). This API will be honored after enabling PERCHIRP_PHASESHIFTER_EN in AWR_RF_RADAR_MISC_CTL_SB.

Table 5.27 describes the contents of this sub block.

Table 5.27: AWR_PERCHIRPPHASESHIFT_CONF_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0106 |
| SBLKLEN | 2 | Value = 12 |
| CHIRP_START_INDXX | 2 | Start index of the chirp for configuring the phase shifter Valid range 0 to 511 |
| CHIRP_END_INDXX | 2 | End index of the chirp for configuring the phase shifter Valid range 0 to 511 |

Continued on next page

Table 5.27 – continued from previous page

| | | |
|-------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX0_PHASE_SHIFTER | 1 | TX0 phase shift value Bits TX0 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX0 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| TX1_PHASE_SHIFTER | 1 | TX1 phase shift value Bits TX1 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX1 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| TX2_PHASE_SHIFTER | 1 | TX2 phase shift value Bits TX2 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX2 phase shift value $1 \text{ LSB} = 360^\circ / 2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| RESERVED | 1 | 0x00 |

NOTE1: Phase shifters are applied at the knee of the ramp.

NOTE2: This API is not supported in IWR6843 ES1.0

5.5.8 Sub block 0x0107 – AWR_PROG_FILT_COEFF_RAM_SET_SB

This sub block can be used to program the coefficients for the external programmable filter. The API is applicable only in xWR1642/IWR6843/xWR1843.

Note that the programmable filter is applicable in Complex 1X and Real-only output modes for sampling rates under 6.25 Msps (Complex 1X) and under 12.5 Msps (Real). This is to allow for a trade-off between digital filter chain setting time and close-in anti-alias attenuation. A real-coefficient FIR with up to 26 taps (16-bit coefficients) is supported in the Complex 1X output mode, and a real-coefficient FIR with up to 20 taps (16-bit coefficients) is supported in the Real output mode

NOTE: The programmable filter configuration APIs are not supported in DFP 1.x (1st generation devices). Please refer latest DFP release note for more info.

NOTE: This API should be issued before AWR_PROFILE_CONF_SET_SB.

Table 5.28 describes the contents of this sub block.

Table 5.28: AWR_PROG_FILT_COEFF_RAM_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0107 |
| SBLKLEN | 2 | Value = 212 |
| COEFF_ARRAY | 208 | <p>The array of coefficients for the programmable filter, across all profiles, to be stored in the coefficient RAMS. Each tap is a 16-bit signed <1.15, s> number. The exact set of taps to be used for a given profile can be specified through AWR_PROG_FILT_CONF_SB</p> <p>NOTE: All the filter taps across profiles are to be provided in one shot. There is a HW constraint that each profile's filter taps should start at four 32-bit word aligned address (i.e., the coefficients corresponding to any profile should start at array index which is a multiple of 8). Unused coefficients shall be initialized to zero.</p> |

5.5.9 Sub block 0x0108 – AWR_PROG_FILT_CONF_SET_SB

This sub block can be used to configure the coefficients for the external programmable filter and associate them to a certain profile. The API is applicable only in xWR1642/IWR6843/xWR1843. This API should be issued before AWR_PROFILE_CONF_SET_SB.

NOTE: The programmable filter configuration APIs are not supported in this release. Please refer latest DFP release note for more info.

Table 5.29 describes the contents of this sub block.

Table 5.29: AWR_PROG_FILT_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0108 |
| SBLKLEN | 2 | Value = 8 |
| PROFILE_INDX | 1 | This field indicates the profile Index for which this configuration applies. |

Continued on next page

Table 5.29 – continued from previous page

| | | |
|-----------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PROG_FILT_COEFF_START_INDEX | 1 | The index of the first coefficient of the programmable filter taps corresponding to this profile in the coefficient RAM programmed using AWR_PROG_FILT_COEFF_SET_SB NOTE: The profile's filter tap start index shall be 8 tap aligned (four 32-bit word aligned address). |
| PROG_FILT_LENGTH | 1 | The length (number of taps) of the filter corresponding to this profile. Together with the previous field, this determines the set of coefficients picked up from the coefficient RAM to form the filter taps for this profile. NOTE: This has to be an even number. For odd-length filters, a 0 (zero) tap needs to be appended at the end to make the length even. This is a HW constraint. |
| PROG_FILT_FREQ_SHIFT_FACTOR | 1 | Relevant only for the Complex 1x output mode with the programmable filter. Determines the magnitude of the frequency shift do be done before filtering using the real-coefficient programmable filter. 1 LSB = $0.01 \times F_s$ shift, where F_s is the output sampling rate, specified as PF_DIGITAL_OUTPUT_SAMPLING_RATE in AWR_PROFILE_CONF_SET_SB |

| | |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NOTE1: | PROG_FILT_COEFF_START_INDEX should be 8 tap aligned (four 32-bit word aligned address) |
| NOTE2: | Programmable filter APIs (AWR_PROG_FILT_COEFF_RAM_SET_SB and AWR_PROG_FILT_CONF_SET_SB) should not be issued when frames are ongoing. |

5.5.10 Sub block 0x0109 – AWR_CALIB_MON_TIME_UNIT_CONF_SB

This API sub block is used to set calibration and monitoring time unit.

Table 5.30: AWR_CALIB_MON_TIME_UNIT_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0109 |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 5.30 – continued from previous page

| | | |
|---------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALIB_MON_TIME_UNIT | 2 | <p>Defines the basic time unit, in terms of which calibration and/or monitoring periodicities are to be defined.</p> <p>If any monitoring functions are desired and enabled, the monitoring infrastructure automatically inherits this time unit as the period over which the various monitors are cyclically executed; so this should be set to the desired FTTL.</p> <p>For calibrations, a separate CALIB_PERIODICITY can be specified, as a multiple of this time unit, in AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB NOTE: Even though calibrations may not be desired every time unit, every time unit shall be made long enough to include active chirping time, time required for all enabled calibrations and monitoring functions.</p> <p>1 LSB = Duration of one frame Recommendation: See examples in Section 10 Default value: 100 Valid range : 40ms to 250ms (Derive actual count value from programmed frame period)</p> |
| NUM_OF_CASCADED_DEV | 1 | <p>Applicable only in cascaded mode. In non-cascaded mode set this to 1 Default value: 1</p> |
| DEVICE_ID | 1 | <p>Applicable only in cascaded mode. In non-cascaded mode set this to 0 Default value: 0</p> |
| RESERVED | 4 | 0x0000_0000 |

NOTE: The Minimum total blank time in a CAL_MON_TIME_UNIT shall be 1ms to run internal APLL and SYNTH calibrations + ~12.5% of CAL_MON_TIME_UNIT for WDT clearing time if WDT is enabled.

5.5.11 Sub block 0x010A – AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB

This API is used to trigger one time calibrations instantaneously or schedule periodic run time calibrations, which will be scheduled while framing in inter-burst idle time (Min available idle time of 250 μ s is required).

Table 5.31: AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|----|----------|----|----------|----|----------|----|----------|----|-----------------------|----|----------|----|----------|----|----------|----|-------------------|----|-------------------------|-----|------------------------|-----|----------|-----|----------|--------|----------|
| SBLKID | 2 | Value = 0x010A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ONE_TIME_CALIB_ENABLE_MASK | 4 | <p>Upon receiving this trigger message, one time calibration of various RF/analog aspects are triggered if the corresponding bits in this field are set to 1. The response is in the form of an asynchronous event sent to the host. The calibrations, if enabled, are performed after the completion of any ongoing calibration cycle, and the calibration results take effect from the frame that begins after the asynchronous event response is sent from the BSS. APLL and SYNTH calibrations are done always internally irrespective of bits are enabled or not, the time required for these calibrations must be allocated.</p> <table border="0"> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>b0</td> <td>RESERVED</td> </tr> <tr> <td>b1</td> <td>RESERVED</td> </tr> <tr> <td>b2</td> <td>RESERVED</td> </tr> <tr> <td>b3</td> <td>RESERVED</td> </tr> <tr> <td>b4</td> <td>LODIST_CALIBRATION.EN</td> </tr> <tr> <td>b5</td> <td>RESERVED</td> </tr> <tr> <td>b6</td> <td>RESERVED</td> </tr> <tr> <td>b7</td> <td>RESERVED</td> </tr> <tr> <td>b8</td> <td>PD_CALIBRATION.EN</td> </tr> <tr> <td>b9</td> <td>TX_POWER_CALIBRATION.EN</td> </tr> <tr> <td>b10</td> <td>RX_GAIN_CALIBRATION.EN</td> </tr> <tr> <td>b11</td> <td>RESERVED</td> </tr> <tr> <td>b12</td> <td>RESERVED</td> </tr> <tr> <td>b31:13</td> <td>RESERVED</td> </tr> </table> <p>0b0000_0000_0000_0000_000 Default value: 0</p> | Bit | Definition | b0 | RESERVED | b1 | RESERVED | b2 | RESERVED | b3 | RESERVED | b4 | LODIST_CALIBRATION.EN | b5 | RESERVED | b6 | RESERVED | b7 | RESERVED | b8 | PD_CALIBRATION.EN | b9 | TX_POWER_CALIBRATION.EN | b10 | RX_GAIN_CALIBRATION.EN | b11 | RESERVED | b12 | RESERVED | b31:13 | RESERVED |
| Bit | Definition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b1 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b2 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b4 | LODIST_CALIBRATION.EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b5 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b6 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b8 | PD_CALIBRATION.EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b9 | TX_POWER_CALIBRATION.EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b10 | RX_GAIN_CALIBRATION.EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b11 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b12 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b31:13 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.31 – continued from previous page

| PERIODIC_ CALIB_ENABLE_ MASK | 4 | <p>Automatic periodic triggering of calibrations of various RF/analog aspects can be set up by the host issuing this message with corresponding bits in this field set to 1.</p> <table border="0"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>b0</td><td>RESERVED</td></tr> <tr><td>b1</td><td>RESERVED</td></tr> <tr><td>b2</td><td>RESERVED</td></tr> <tr><td>b3</td><td>RESERVED</td></tr> <tr><td>b4</td><td>LODIST_CALIBRATION_EN</td></tr> <tr><td>b5</td><td>RESERVED</td></tr> <tr><td>b6</td><td>RESERVED</td></tr> <tr><td>b7</td><td>RESERVED</td></tr> <tr><td>b8</td><td>PD_CALIBRATION_EN</td></tr> <tr><td>b9</td><td>TX_POWER_CALIBRATION_EN</td></tr> <tr><td>b10</td><td>RX_GAIN_CALIBRATION_EN</td></tr> <tr><td>b11</td><td>RESERVED</td></tr> <tr><td>b12</td><td>RESERVED</td></tr> <tr><td>b31:13</td><td>RESERVED</td></tr> </tbody> </table> <p>APLL and SYNTH calibrations are done always internally (at a periodicity of 1 second) irrespective of bits are enabled or not, the time required for these calibrations must be allocated. Refer to Table 10.2 for the duration of run time calibrations</p> <p>Default value: 0</p> | Bit | Definition | b0 | RESERVED | b1 | RESERVED | b2 | RESERVED | b3 | RESERVED | b4 | LODIST_CALIBRATION_EN | b5 | RESERVED | b6 | RESERVED | b7 | RESERVED | b8 | PD_CALIBRATION_EN | b9 | TX_POWER_CALIBRATION_EN | b10 | RX_GAIN_CALIBRATION_EN | b11 | RESERVED | b12 | RESERVED | b31:13 | RESERVED |
|------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|----|----------|----|----------|----|----------|----|----------|----|-----------------------|----|----------|----|----------|----|----------|----|-------------------|----|-------------------------|-----|------------------------|-----|----------|-----|----------|--------|----------|
| Bit | Definition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b1 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b2 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b4 | LODIST_CALIBRATION_EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b5 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b6 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b8 | PD_CALIBRATION_EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b9 | TX_POWER_CALIBRATION_EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b10 | RX_GAIN_CALIBRATION_EN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b11 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b12 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b31:13 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.31 – continued from previous page

| | | | | | | | | | | | | | | |
|-------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|----|-----------------------|---|------------------------------|---|-----------------------------|--|------------------|------|----------|
| CALIBRATION_PERIODICITY | 4 | <p>This field is applicable only for those calibrations which are enabled to be done periodically in the PERIODIC_CALIB_ENABLE_MASK field. This field indicates the desired periodicity of calibrations.</p> <p>If this field is set to N, the results of the first calibration (based on ONE_TIME_CALIB_ENABLE_MASK) are applicable for the first N CALIB_MON_TIME_UNITS. The results of the next calibration are applicable for the next N CALIB_MON_TIME_UNITS, and so on.</p> <p>Recommendation: Set CALIBRATION_PERIODICITY such that frequency of calibrations is greater than or equal to 1 second.</p> <p>1 LSB = 1 CALIB_MON_TIME_UNIT, as specified in AWR_CALIB_MON_TIME_UNIT_CONF_SB.</p> <p>If the user does not wish to receive calibration reports when periodic calibrations are not enabled, then the user should set CALIBRATION_PERIODICITY to 0 Default value: 0 Valid Range: 0 (Disable), 4 to 100 (value 1 is not a valid value, this will cause internal APLL and SYNTH calibrations to stop)</p> | | | | | | | | | | | | |
| ENABLE_CAL_REPORT | 1 | <table border="0"> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>b0</td> <td>ENABLE_SUMMARY_REPORT</td> </tr> <tr> <td>0</td> <td>Summary reports are disabled</td> </tr> <tr> <td>1</td> <td>Summary reports are enabled</td> </tr> <tr> <td></td> <td>Default value: 0</td> </tr> <tr> <td>b7:1</td> <td>RESERVED</td> </tr> </table> <p>NOTE1: If calibration reports are enabled, the reports will be sent every 1 second whenever internal calibrations (APLL and SYNTH) are triggered and at every CALIBRATION_PERIODICITY when the user enabled calibrations are triggered.</p> <p>NOTE2: If user has not enabled any one time calibrations, but if calibration report is enabled, then after issuing this API, the firmware will attempt to run the APLL and SYNTH calibrations and the calibration report will be immediately sent out.</p> | Bit | Definition | b0 | ENABLE_SUMMARY_REPORT | 0 | Summary reports are disabled | 1 | Summary reports are enabled | | Default value: 0 | b7:1 | RESERVED |
| Bit | Definition | | | | | | | | | | | | | |
| b0 | ENABLE_SUMMARY_REPORT | | | | | | | | | | | | | |
| 0 | Summary reports are disabled | | | | | | | | | | | | | |
| 1 | Summary reports are enabled | | | | | | | | | | | | | |
| | Default value: 0 | | | | | | | | | | | | | |
| b7:1 | RESERVED | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | |

Continued on next page

Table 5.31 – continued from previous page

| | | | |
|-------------------|---|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX_POWER_CAL_MODE | 1 | Bit | Definition |
| | | b0 | TX_POWER_CAL_MODE (Not valid for IWR6843 ES1.0) |
| | | 0 | Update TX gain setting from LUT and do a closed loop calibration (OLPC + CLPC) |
| | | 1 | Update TX gain settings from LUT only (OLPC only) OLPC: Open Loop Power Control. In this mode the TX stage codes are set based on a coarse measurement and a LUT generated for every temperature and the stage codes are picked from the LUT CLPC: Closed Loop Power Control. In this mode the TX stage codes are picked from the coarse LUT as generated in OLPC step. Later the TX power is measured and the TX stage codes are corrected to achieve the desired TX power accuracy. Default value: 0 |
| | | b7:1 | RESERVED |
| RESERVED | 1 | 0x00 | |
| RESERVED | 4 | 0x00000000 | |

NOTE1: The API AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB can be issued when the device is framing, the calibration periodicity update or one time calibrations can be done while frames are running.

NOTE2: Only APLL, SYNTH1, SYNTH2 and LODIST run time calibrations are supported in IWR6843 ES1.0 device

5.5.12 Sub block 0x010B – AWR_INTER_RX_GAIN_PHASE_CONTROL_SB

This API can be used to induce different gain/phase offsets on the different RXs, for inter-RX mismatch compensation.

NOTE: The Inter RX gain phase control configuration API is not supported in this release. Please refer latest DFP release note for more info.

Table 5.32: AWR_INTER_RX_GAIN_PHASE_CONTROL_SB contents

| Field Name | Number of bytes | Description |
|---------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x010B |
| SBLKLEN | 2 | Value = 28 |
| PROFILE.INDX | 1 | This field indicates the profile Index for which this configuration applies. |
| RESERVED | 3 | 0x000000 |
| DIGITAL_GAIN | 4 | One byte per RX (8-bit signed number) Bits Assignment b7:0 RX0 digital gain b15:8 RX1 digital gain b23:16 RX2 digital gain b31:24 RX3 digital gain 1 LSB = 0.1 dB Valid Range: -120 to 119 |
| DIGITAL_PHASE_SHIFT | 8 | Two bytes per RX Bits Assignment b15:0 RX0 digital phase shift b31:16 RX1 digital phase shift b47:32 RX2 digital phase shift b63:48 RX3 digital phase shift 1 LSB = $360^\circ / 2^{16} \approx 0.0055^\circ$ Valid Range: 0 to 65535 NOTE: This field is NOT applicable when ADC_OUT_FMT is 00 (real output) |
| RESERVED | 8 | 0x00000000 |

5.5.13 Sub block 0x010C – AWR_RX_GAIN_TEMPLUT_SET_SB

This API can be used to overwrite the RX gain temperature LUT used in firmware. This API should be issued after profile configuration API.

Table 5.33: AWR_RX_GAIN_TEMPLUT_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x010C |
| SBLKLEN | 2 | Value = 28 |

Continued on next page

Table 5.33 – continued from previous page

| | | | | | | | | | | | | | | | | | | | | |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|------------|-----------------|--------------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|---------|------------------------|-----------------|------------------------|------------------------|---|------------------------|---|------------------------|---|------------------------|
| PROFILE_INDEX | 1 | This field indicates the profile Index for which this configuration applies. | | | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | | |
| RX_GAIN_CODE | 19 | <p>Byte0: RX gain code for temperature <-30 °C</p> <p>Byte1: RX gain code for temperature [-30, -20) °C</p> <p>Byte2: RX gain code for temperature [-20, -10) °C</p> <p>Byte3: RX gain code for temperature [-10, 0) °C</p> <p>Byte4: RX gain code for temperature [0, 10) °C</p> <p>Byte5: RX gain code for temperature [10, 20) °C</p> <p>Byte6: RX gain code for temperature [20, 30) °C</p> <p>Byte7: RX gain code for temperature [30, 40) °C</p> <p>Byte8: RX gain code for temperature [40, 50) °C</p> <p>Byte9: RX gain code for temperature [50, 60) °C</p> <p>Byte10: RX gain code for temperature [60, 70) °C</p> <p>Byte11: RX gain code for temperature [70, 80) °C</p> <p>Byte12: RX gain code for temperature [80, 90) °C</p> <p>Byte13: RX gain code for temperature [90, 100) °C</p> <p>Byte14: RX gain code for temperature [100, 110) °C</p> <p>Byte15: RX gain code for temperature [110, 120) °C</p> <p>Byte16: RX gain code for temperature [120, 130) °C</p> <p>Byte17: RX gain code for temperature [130, 140) °C</p> <p>Byte18: RX gain code for temperature ≥140 °C</p> <p>Each byte is encoded as follows</p> <table border="0"> <tr> <td>Bits</td> <td>Definition</td> </tr> <tr> <td>b4:0</td> <td> IF_GAIN_CODE IF gain is IF_GAIN_CODE ×2 – 6 dB Valid values: 0 to 17 1 LSB = 2 dB </td> </tr> <tr> <td>b7:5</td> <td> RF_GAIN_CODE <table border="0"> <tr> <td>Value</td> <td>RF Gain</td> </tr> <tr> <td>0</td> <td>Maximum RF gain</td> </tr> <tr> <td>1</td> <td>Maximum RF gain – 2 dB</td> </tr> <tr> <td>2</td> <td>Maximum RF gain – 4 dB</td> </tr> <tr> <td>3</td> <td>Maximum RF gain – 6 dB</td> </tr> <tr> <td>4</td> <td>Maximum RF gain – 8 dB</td> </tr> </table> </td> </tr> </table> | Bits | Definition | b4:0 | IF_GAIN_CODE IF gain is IF_GAIN_CODE ×2 – 6 dB Valid values: 0 to 17 1 LSB = 2 dB | b7:5 | RF_GAIN_CODE <table border="0"> <tr> <td>Value</td> <td>RF Gain</td> </tr> <tr> <td>0</td> <td>Maximum RF gain</td> </tr> <tr> <td>1</td> <td>Maximum RF gain – 2 dB</td> </tr> <tr> <td>2</td> <td>Maximum RF gain – 4 dB</td> </tr> <tr> <td>3</td> <td>Maximum RF gain – 6 dB</td> </tr> <tr> <td>4</td> <td>Maximum RF gain – 8 dB</td> </tr> </table> | Value | RF Gain | 0 | Maximum RF gain | 1 | Maximum RF gain – 2 dB | 2 | Maximum RF gain – 4 dB | 3 | Maximum RF gain – 6 dB | 4 | Maximum RF gain – 8 dB |
| Bits | Definition | | | | | | | | | | | | | | | | | | | |
| b4:0 | IF_GAIN_CODE IF gain is IF_GAIN_CODE ×2 – 6 dB Valid values: 0 to 17 1 LSB = 2 dB | | | | | | | | | | | | | | | | | | | |
| b7:5 | RF_GAIN_CODE <table border="0"> <tr> <td>Value</td> <td>RF Gain</td> </tr> <tr> <td>0</td> <td>Maximum RF gain</td> </tr> <tr> <td>1</td> <td>Maximum RF gain – 2 dB</td> </tr> <tr> <td>2</td> <td>Maximum RF gain – 4 dB</td> </tr> <tr> <td>3</td> <td>Maximum RF gain – 6 dB</td> </tr> <tr> <td>4</td> <td>Maximum RF gain – 8 dB</td> </tr> </table> | Value | RF Gain | 0 | Maximum RF gain | 1 | Maximum RF gain – 2 dB | 2 | Maximum RF gain – 4 dB | 3 | Maximum RF gain – 6 dB | 4 | Maximum RF gain – 8 dB | | | | | | | |
| Value | RF Gain | | | | | | | | | | | | | | | | | | | |
| 0 | Maximum RF gain | | | | | | | | | | | | | | | | | | | |
| 1 | Maximum RF gain – 2 dB | | | | | | | | | | | | | | | | | | | |
| 2 | Maximum RF gain – 4 dB | | | | | | | | | | | | | | | | | | | |
| 3 | Maximum RF gain – 6 dB | | | | | | | | | | | | | | | | | | | |
| 4 | Maximum RF gain – 8 dB | | | | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | | | |

NOTE: This API is not supported in IWR6843 ES1.0

5.5.14 Sub block 0x010D – AWR_TX_GAIN_TEMPLUT_SET_SB

This API can be used to overwrite the TX gain temperature LUT used in firmware. This API should be issued after profile configuration API.

Table 5.34: AWR_TX_GAIN_TEMPLUT_SET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x010D |
| SBLKLEN | 2 | Value = 68 |
| PROFILE_INDX | 1 | This field indicates the profile Index for which this configuration applies |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 5.34 – continued from previous page

| | | |
|---------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX0_GAIN_CODE | 19 | Byte0: TX0 gain code for temperature <-30 °C Byte1: TX0 gain code for temperature [-30, -20) °C Byte2: TX0 gain code for temperature [-20, -10) °C Byte3: TX0 gain code for temperature [-10, 0) °C Byte4: TX0 gain code for temperature [0, 10) °C Byte5: TX0 gain code for temperature [10, 20) °C Byte6: TX0 gain code for temperature [20, 30) °C Byte7: TX0 gain code for temperature [30, 40) °C Byte8: TX0 gain code for temperature [40, 50) °C Byte9: TX0 gain code for temperature [50, 60) °C Byte10: TX0 gain code for temperature [60, 70) °C Byte11: TX0 gain code for temperature [70, 80) °C Byte12: TX0 gain code for temperature [80, 90) °C Byte13: TX0 gain code for temperature [90, 100) °C Byte14: TX0 gain code for temperature [100, 110) °C Byte15: TX0 gain code for temperature [110, 120) °C Byte16: TX0 gain code for temperature [120, 130) °C Byte17: TX0 gain code for temperature [130, 140) °C Byte18: TX0 gain code for temperature ≥140 °C Each byte is encoded as follows Bits Definition b5:0 STG_CODE Higher values for higher gain b7:6 RESERVED |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 5.34 – continued from previous page

| | | |
|---------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX1_GAIN_CODE | 19 | Byte0: TX1 gain code for temperature <-30 °C Byte1: TX1 gain code for temperature [-30, -20) °C Byte2: TX1 gain code for temperature [-20, -10) °C Byte3: TX1 gain code for temperature [-10, 0) °C Byte4: TX1 gain code for temperature [0, 10) °C Byte5: TX1 gain code for temperature [10, 20) °C Byte6: TX1 gain code for temperature [20, 30) °C Byte7: TX1 gain code for temperature [30, 40) °C Byte8: TX1 gain code for temperature [40, 50) °C Byte9: TX1 gain code for temperature [50, 60) °C Byte10: TX1 gain code for temperature [60, 70) °C Byte11: TX1 gain code for temperature [70, 80) °C Byte12: TX1 gain code for temperature [80, 90) °C Byte13: TX1 gain code for temperature [90, 100) °C Byte14: TX1 gain code for temperature [100, 110) °C Byte15: TX1 gain code for temperature [110, 120) °C Byte16: TX1 gain code for temperature [120, 130) °C Byte17: TX1 gain code for temperature [130, 140) °C Byte18: TX1 gain code for temperature ≥140 °C Each byte is encoded as follows Bits Definition b5:0 STG_CODE Higher values for higher gain b7:6 RESERVED |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 5.34 – continued from previous page

| | | | | | | | | |
|---------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|------|-------------------------------------------|------|----------|
| TX2_GAIN_CODE | 19 | Byte0: TX2 gain code for temperature <-30 °C Byte1: TX2 gain code for temperature [-30, -20) °C Byte2: TX2 gain code for temperature [-20, -10) °C Byte3: TX2 gain code for temperature [-10, 0) °C Byte4: TX2 gain code for temperature [0, 10) °C Byte5: TX2 gain code for temperature [10, 20) °C Byte6: TX2 gain code for temperature [20, 30) °C Byte7: TX2 gain code for temperature [30, 40) °C Byte8: TX2 gain code for temperature [40, 50) °C Byte9: TX2 gain code for temperature [50, 60) °C Byte10: TX2 gain code for temperature [60, 70) °C Byte11: TX2 gain code for temperature [70, 80) °C Byte12: TX2 gain code for temperature [80, 90) °C Byte13: TX2 gain code for temperature [90, 100) °C Byte14: TX2 gain code for temperature [100, 110) °C Byte15: TX2 gain code for temperature [110, 120) °C Byte16: TX2 gain code for temperature [120, 130) °C Byte17: TX2 gain code for temperature [130, 140) °C Byte18: TX2 gain code for temperature ≥140 °C Each byte is encoded as follows <table border="0"> <tr> <td>Bits</td> <td>Definition</td> </tr> <tr> <td>b5:0</td> <td>STG_CODE Higher values for higher gain</td> </tr> <tr> <td>b7:6</td> <td>RESERVED</td> </tr> </table> | Bits | Definition | b5:0 | STG_CODE Higher values for higher gain | b7:6 | RESERVED |
| Bits | Definition | | | | | | | |
| b5:0 | STG_CODE Higher values for higher gain | | | | | | | |
| b7:6 | RESERVED | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | |

NOTE: This API is not supported in IWR6843 ES1.0

5.5.15 Sub block 0x010E – AWR_LOOPBACK_BURST_CONF_SET_SB

This API can be used to introduce loopback chirps within the functional frames. This loopback chirps will be introduced only if advanced frame configuration is used where user can define which sub-frame contains loopback chirps. The following loopback configuration will apply to one burst and user can program up to 16 different loopback configurations in 16 different bursts of a given sub-frame. User has to ensure that the corresponding sub-frame is defined in AWR_ADVANCED_FRAME_CONF_SB and sufficient time is given to allow the loopback bursts to be transmitted.

NOTE: The loop-back configuration API is not supported in this release. Please refer latest DFP release note for more info.

NOTE1: If user desires to enable loopback chirps within functional frames, then this API should be issued after AWR_PROFILE_CONF_SET_SB

NOTE2: Only profile based phase shifter is supported in loopback configuration. Per-chirp phase shifter if enabled will not be reflected in loopback chirps.

NOTE3: For the sub-frame in which loopback is desired, user should set SFx_NUM_UNIQUE_CHIRPS_PER_BURST as 1 and can use SFx_NUM_LOOPS_PER_BURST for multiple chirps in the burst.

Table 5.35: AWR_LOOPBACK_BURST_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x010E |
| SBLKLEN | 2 | Value = 48 |
| LOOPBACK_SEL | 1 | Value Definition 0 No loopback 1 IF loopback 2 PS loopback 3 PA loopback Others RESERVED |
| BASE_PROFILE_INDX | 1 | Base profile used for loopback chirps Valid values 0 to 3 |
| BURST_INDX | 1 | Indicates the index of the burst in the loopback sub-frame for which this configuration applies Valid values 0 to 15 |
| RESERVED | 1 | 0x00 |
| FREQ_CONST | 4 | Start frequency for loopback. The start frequency configured here should be within profile's sweep bandwidth. For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9/2^{26}$ Hz \approx 53.644 Hz Valid range: 0x5471C71B to 0x5A000000 For 60GHz Devices (57GHz to 64GHz): 1 LSB = $2.7e9/2^{26}$ Hz \approx 40.233 Hz Valid range: 0x5471C71C to 0x5ED097B4 |

Continued on next page

Table 5.35 – continued from previous page

| SLOPE_CONST | 2 | Frequency slope for loopback burst (32 bit signed number) For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9 \times 900/2^{26} \approx 48.279 \text{ kHz}/\mu\text{s}$ Valid range: -2072 to 2072 For 60GHz Devices (57GHz to 64GHz): 1 LSB = $2.7e9 \times 900/2^{26} \approx 36.21 \text{ kHz}/\mu\text{s}$ Valid range: -6905 to 6905 | | | | | | | | | | |
|-------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|------|-----------------------------------------------------------------------------------------------------------------------|-------|-------------------------------------------------------------------------------------------|--------|------------------------------|--------|----------|
| RESERVED | 2 | 0x0000 | | | | | | | | | | |
| TX_BACKOFF | 4 | <table border="0"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>b7:0</td> <td>TX0 back off 1 LSB = 1 dB</td> </tr> <tr> <td>b15:8</td> <td>TX1 back off 1 LSB = 1 dB</td> </tr> <tr> <td>b23:16</td> <td>TX2 back off 1 LSB = 1 dB</td> </tr> <tr> <td>b31:24</td> <td>RESERVED</td> </tr> </tbody> </table> | Bits | Definition | b7:0 | TX0 back off 1 LSB = 1 dB | b15:8 | TX1 back off 1 LSB = 1 dB | b23:16 | TX2 back off 1 LSB = 1 dB | b31:24 | RESERVED |
| Bits | Definition | | | | | | | | | | | |
| b7:0 | TX0 back off 1 LSB = 1 dB | | | | | | | | | | | |
| b15:8 | TX1 back off 1 LSB = 1 dB | | | | | | | | | | | |
| b23:16 | TX2 back off 1 LSB = 1 dB | | | | | | | | | | | |
| b31:24 | RESERVED | | | | | | | | | | | |
| RX_GAIN | 2 | <table border="0"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>b5:0</td> <td>RX_GAIN This field defines RX gain for each profile 1 LSB = 1 dB Valid values: all even values from 24 to 52</td> </tr> <tr> <td>b7:6</td> <td>RF_GAIN_TARGET Value RF gain target 00 30 dB 01 34 dB 10 RESERVED 11 26 dB</td> </tr> <tr> <td>b15:8</td> <td>RESERVED</td> </tr> </tbody> </table> | Bits | Definition | b5:0 | RX_GAIN This field defines RX gain for each profile 1 LSB = 1 dB Valid values: all even values from 24 to 52 | b7:6 | RF_GAIN_TARGET Value RF gain target 00 30 dB 01 34 dB 10 RESERVED 11 26 dB | b15:8 | RESERVED | | |
| Bits | Definition | | | | | | | | | | | |
| b5:0 | RX_GAIN This field defines RX gain for each profile 1 LSB = 1 dB Valid values: all even values from 24 to 52 | | | | | | | | | | | |
| b7:6 | RF_GAIN_TARGET Value RF gain target 00 30 dB 01 34 dB 10 RESERVED 11 26 dB | | | | | | | | | | | |
| b15:8 | RESERVED | | | | | | | | | | | |
| TX_ENABLE | 1 | <table border="0"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>TX0 Enable</td> </tr> <tr> <td>b1</td> <td>TX1 Enable</td> </tr> <tr> <td>b2</td> <td>TX2 Enable</td> </tr> <tr> <td>b7:3</td> <td>RESERVED</td> </tr> </tbody> </table> | Bits | Definition | b0 | TX0 Enable | b1 | TX1 Enable | b2 | TX2 Enable | b7:3 | RESERVED |
| Bits | Definition | | | | | | | | | | | |
| b0 | TX0 Enable | | | | | | | | | | | |
| b1 | TX1 Enable | | | | | | | | | | | |
| b2 | TX2 Enable | | | | | | | | | | | |
| b7:3 | RESERVED | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | |

Continued on next page

Table 5.35 – continued from previous page

| | | | | | |
|----------------------------|---|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|--------|-----------------------|
| BPM.CONFIG | 2 | Bit | Definition | | |
| | | b0 | CONST_BPM_VAL_TX0_OFF Value of Binary Phase Shift value for TX0, during idle time | | |
| | | b1 | CONST_BPM_VAL_TX0_ON Value of Binary Phase Shift value for TX0, during chirp | | |
| | | b2 | CONST_BPM_VAL_TX1_OFF For TX1 | | |
| | | b3 | CONST_BPM_VAL_TX1_ON For TX1 | | |
| | | b4 | CONST_BPM_VAL_TX2_OFF For TX2 | | |
| | | b5 | CONST_BPM_VAL_TX2_ON For TX2 | | |
| | | b15:6 | RESERVED | | |
| DIGITAL_CORRECTION_DISABLE | 2 | Bits | Digital corrections | | |
| | | b0 | IQMM correction disable (only for PS and PA loopback, for IF loopback IQMM is disabled by firmware) 0 - Enable, 1 - Disable | | |
| | | b1 | Inter-RX Gain and Phase correction disable 0 - Enable, 1 - Disable | | |
| | | b15:2 | RESERVED | | |
| IF_LOOPBACK_FREQ | 1 | Value | IF Loopback frequency | Value | IF Loopback frequency |
| | | 0 | 180 kHz | 8 | 4.02 MHz |
| | | 1 | 240 kHz | 9 | 5 MHz |
| | | 2 | 360 kHz | 10 | 6 MHz |
| | | 3 | 720 kHz | 11 | 8.03 MHz |
| | | 4 | 1 MHz | 12 | 9 MHz |
| | | 5 | 2 MHz | 13 | 10 MHz |
| | | 6 | 2.5 MHz | 255-14 | RESERVED |
| | | 7 | 3 MHz | | |
| IF_LOOPBACK_MAG | 1 | 1 LSB = 10 mV Valid range: 1 to 63 | | | |

Continued on next page

Table 5.35 – continued from previous page

| PS1_PGA_GAIN_INDEX | 1 | Value | PGA | gain | Value | PGA | gain |
|--------------------|---|-------|------------|------|--------|----------|------|
| | | | value | | | value | |
| | | 0 | PGA is OFF | | 15 | -3 dB | |
| | | 1 | -22 dB | | 16 | -2 dB | |
| | | 2 | -16 dB | | 17 | -1 dB | |
| | | 3 | -15 dB | | 18 | 0 dB | |
| | | 4 | -14 dB | | 19 | 1 dB | |
| | | 5 | -13 dB | | 20 | 2 dB | |
| | | 6 | -12 dB | | 21 | 3 dB | |
| | | 7 | -11 dB | | 22 | 4 dB | |
| | | 8 | -10 dB | | 23 | 5 dB | |
| | | 9 | -9 dB | | 24 | 6 dB | |
| | | 10 | -8 dB | | 25 | 7 dB | |
| | | 11 | -7 dB | | 26 | 8 dB | |
| | | 12 | -6 dB | | 27 | 9 dB | |
| | | 13 | -5 dB | | 255-28 | RESERVED | |
| | | 14 | -4 dB | | | | |

| PS2_PGA_GAIN_INDEX | 1 | Value | PGA | gain | Value | PGA | gain |
|--------------------|---|-------|------------|------|--------|----------|------|
| | | | value | | | value | |
| | | 0 | PGA is OFF | | 15 | -3 dB | |
| | | 1 | -22 dB | | 16 | -2 dB | |
| | | 2 | -16 dB | | 17 | -1 dB | |
| | | 3 | -15 dB | | 18 | 0 dB | |
| | | 4 | -14 dB | | 19 | 1 dB | |
| | | 5 | -13 dB | | 20 | 2 dB | |
| | | 6 | -12 dB | | 21 | 3 dB | |
| | | 7 | -11 dB | | 22 | 4 dB | |
| | | 8 | -10 dB | | 23 | 5 dB | |
| | | 9 | -9 dB | | 24 | 6 dB | |
| | | 10 | -8 dB | | 25 | 7 dB | |
| | | 11 | -7 dB | | 26 | 8 dB | |
| | | 12 | -6 dB | | 27 | 9 dB | |
| | | 13 | -5 dB | | 255-28 | RESERVED | |
| | | 14 | -4 dB | | | | |

Continued on next page

Table 5.35 – continued from previous page

| | | |
|------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PS_LOOPBACK_FREQ | 4 | Phase shifter loop back frequency in kHz 1 LSB = 1 kHz Bits Definition b15:0 TX0 Loopback Frequency [b31:16] TX1 Loopback Frequency |
| RESERVED | 4 | RESERVED |
| PA_LOOPBACK_FREQ | 2 | This value is a 100 MHz divider which sets the loopback frequency For e.g. for a 1 MHz frequency, set this to 100 For a 2 MHz frequency, set this to 50 NOTE: To ensure no leakage of signal power, user has to ensure that 100MHz/LOOPBACK_FREQ is an integer multiple of bin width For e.g. if user choses 25Msps sampling rate and 2048 samples/chirp, then LOOPBACK_FREQ of 64 (=1.5625 MHz) will ensure no leakage |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |

NOTE: This API is not supported in IWR6843 ES1.0

5.5.16 Sub block 0x010F – AWR_DYN_CHIRP_CONF_SET_SB

This API can be used to dynamically change the chirp configuration while frames are on-going. The configuration will be stored in software and the new configuration will be applied after receiving the AWR_DYN_CHIRP_ENABLE_SB API.

Table 5.36: AWR_DYN_CHIRP_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x010F |
| SBLKLEN | 2 | Value = 200 |

Continued on next page

Table 5.36 – continued from previous page

| CHIRP_ROW_SELECT | 1 | Bits Description b3:0 RESERVED b7:4 If user does not wish to reconfigure all 3 chirp rows, then the following mode can be used to configure only one row per chirp which enables the user to configure 48 chirps in one API, effectively saving on the reconfiguration time. If CHIRP_ROW_SELECT[7:4] is non-zero, then the API parameters CHIRP _x _R1, CHIRP _x _R2 and CHIRP _x _R3 for $1 \leq x \leq 16$ in this API would mean CHIRP(3x - 2)_R _y , CHIRP(3x - 1)_R _y and CHIRP(3x)_R _y where <i>y</i> is as per the below table (Not supported in IWR6843 ES1.0, Value shall be 0) <table data-bbox="762 801 1284 1169"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0b0000</td> <td>Enables all 3 chirp rows to be reconfigured (default)</td> </tr> <tr> <td>0b0001</td> <td>Enables only chirp row 1 to be reconfigured</td> </tr> <tr> <td>0b0010</td> <td>Enables only chirp row 2 to be reconfigured</td> </tr> <tr> <td>0b0011</td> <td>Enables only chirp row 3 to be reconfigured</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </tbody> </table> | Value | Definition | 0b0000 | Enables all 3 chirp rows to be reconfigured (default) | 0b0001 | Enables only chirp row 1 to be reconfigured | 0b0010 | Enables only chirp row 2 to be reconfigured | 0b0011 | Enables only chirp row 3 to be reconfigured | Others | RESERVED |
|----------------------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|--------|-------------------------------------------------------|--------|---------------------------------------------|--------|---------------------------------------------|--------|---------------------------------------------|--------|----------|
| Value | Definition | | | | | | | | | | | | | |
| 0b0000 | Enables all 3 chirp rows to be reconfigured (default) | | | | | | | | | | | | | |
| 0b0001 | Enables only chirp row 1 to be reconfigured | | | | | | | | | | | | | |
| 0b0010 | Enables only chirp row 2 to be reconfigured | | | | | | | | | | | | | |
| 0b0011 | Enables only chirp row 3 to be reconfigured | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | |
| CHIRP_SEGMENT_SELECT | 1 | Valid range 0 to 31. Indicates the segment of the chirp RAM that the 16 chirp definitions in this sub block map to | | | | | | | | | | | | |
| PROGRAM_MODE | 2 | Bits Description b0 Value Definition 0 Program the new configuration when AWR_DYN_CHIRP_ENABLE API is issued 1 Program the new configuration immediately NOTE: User has to ensure that the chirps which are being reconfigured are not the ones which are currently in use for chirping b15:1 RESERVED | | | | | | | | | | | | |

Continued on next page

Table 5.36 – continued from previous page

| | | | |
|-----------|---|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHIRP1_R1 | 4 | Bits | Definition |
| | | b3:0 | PROFILE_INDX Valid range 0 to 3 |
| | | b7:4 | RESERVED |
| | | b13:8 | FREQ_SLOPE_VAR For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9 \times 900/2^{26} \approx 48.279$ kHz Valid range: 0 to 63 For 60GHz Devices (57GHz to 64GHz): 1 LSB = $2.7e9 \times 900/2^{26} \approx 36.21$ kHz Valid range: 0 to 63 |
| | | b15:14 | RESERVED |
| | | b18:16 | TX_ENABLE Bit Definition b0 TX0 Enable b1 TX1 Enable b2 TX2 Enable |
| | | b23:19 | RESERVED |
| | | b29:24 | BPM_CONSTANT_BITS Bit Definition b0 CONST_BPM_VAL_TX0_OFF Value of Binary Phase Shift value for TX0, during idle time b1 CONST_BPM_VAL_TX0_ON Value of Binary Phase Shift value for TX0, during chirp b2 CONST_BPM_VAL_TX1_OFF For TX1 b3 CONST_BPM_VAL_TX1_ON For TX1 b4 CONST_BPM_VAL_TX2_OFF For TX2 b5 CONST_BPM_VAL_TX2_ON For TX2 |
| | | b31:30 | RESERVED |

Continued on next page

Table 5.36 – continued from previous page

| | | | |
|------------|-----|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHIRP1_R2 | 4 | Bits | Definition |
| | | b22:0 | FREQ_START_VAR For 77GHz Devices (76GHz to 81GHz): 1 LSB = $3.6e9/2^{26} \approx 53.644$ Hz Valid range: 0 to 8388607 For 60GHz Devices (57GHz to 64GHz): 1 LSB = $2.7e9/2^{26} \approx 40.233$ Hz Valid range: 0 to 8388607 |
| | | b31:23 | RESERVED |
| CHIRP1_R3 | 4 | Bits | Definition |
| | | b11:0 | IDLE_TIME_VAR 1 LSB = 10 ns Valid range: 0 to 4095 |
| | | b15:12 | RESERVED |
| | | b27:16 | ADC_START_TIME_VAR 1 LSB = 10 ns Valid range: 0 to 4095 |
| | | b31:28 | RESERVED |
| CHIRP2_R1 | 4 | See description for CHIRP1_R1 | |
| CHIRP2_R2 | 4 | See description for CHIRP1_R2 | |
| CHIRP2_R3 | 4 | See description for CHIRP1_R3 | |
| ... | ... | ... | |
| CHIRP16_R1 | 4 | See description for CHIRP1_R1 | |
| CHIRP16_R2 | 4 | See description for CHIRP1_R2 | |
| CHIRP16_R3 | 4 | See description for CHIRP1_R3 | |

NOTE:

If user wants to update the chirp ram rows using dynamic chirp config API in runtime then it is must to use same dynamic chirp config API (instead of legacy chirp config API) to configure all chirp parameters during sensor initialization.

5.5.17 Sub block 0x0110 – AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SET_SB

This API can be used to dynamically change the per-chirp phase shifter configuration (applicable only in certain devices) while frames are on-going. The configuration will be stored in software and the new configuration will be applied after receiving the AWR_DYN_CHIRP_ENABLE_SB API.

Table 5.37: AWR_DYN_PERCHIRP_PHASESHIFTER_CONF_SB contents

| Field Name | Number of bytes | Description |
|-------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0110 |
| SBLKLEN | 2 | Value = 56 |
| RESERVED | 1 | 0x00 |
| CHIRP_SEG- MENT.SELECT | 1 | Indicates the segment of the chirp RAM that the 16 chirp definitions in this sub block map to. Valid range 0 to 31 |
| CHIRP1_TX0_ PHASE.SHIFTER | 1 | TX0 phase shift value Bits TX0 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX0 phase shift value 1 LSB = $360^\circ/2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| CHIRP1_TX1_ PHASE.SHIFTER | 1 | TX1 phase shift value Bits TX1 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX1 phase shift value 1 LSB = $360^\circ/2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| CHIRP1_TX2_ PHASE.SHIFTER | 1 | TX2 phase shift value Bits TX1 phase shift definition b1:0 RESERVED (set it to 0b00) b7:2 TX1 phase shift value 1 LSB = $360^\circ/2^6 = 5.625^\circ$ Valid range: 0 to 63 |
| CHIRP2_TX0_ PHASE.SHIFTER | 1 | See description for CHIRP1_TX0_PHASE.SHIFTER |
| CHIRP2_TX1_ PHASE.SHIFTER | 1 | See description for CHIRP2_TX1_PHASE.SHIFTER |
| CHIRP2_TX2_ PHASE.SHIFTER | 1 | See description for CHIRP3_TX2_PHASE.SHIFTER |
| ... | ... | ... |
| CHIRP16_TX0_ PHASE.SHIFTER | 1 | See description for CHIRP1_TX0_PHASE.SHIFTER |
| CHIRP16_TX1_ PHASE.SHIFTER | 1 | See description for CHIRP2_TX1_PHASE.SHIFTER |
| CHIRP16_TX2_ PHASE.SHIFTER | 1 | See description for CHIRP3_TX2_PHASE.SHIFTER |

Continued on next page

Table 5.37 – continued from previous page

| | | | |
|---------------|---|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PROGRAM_ MODE | 2 | Bits | Description |
| | | b0 | Value Definition |
| | | 0 | Program the new configuration when AWR_DYN_CHIRP_ENABLE API is issued |
| | | 1 | Program the new configuration immediately NOTE: User has to ensure that the chirps which are being reconfigured are not the ones which are currently in use for chirping |
| | | b15:1 | RESERVED |

NOTE: This API is not supported in IWR6843 ES1.0

5.5.18 Sub block 0x0111 – AWR_DYN_CHIRP_ENABLE_SB

This API can be used to trigger the copy of chirp configuration from software to hardware. The copy will be performed at the end of the ongoing frame.

Table 5.38: AWR_DYN_CHIRP_ENABLE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0111 |
| SBLKLEN | 2 | Value = 8 |
| RESERVED | 4 | 0x00000000 |

NOTE: HW reconfiguration time (as shown in the figure below) is around 200 μ s. User has to ensure that AWR_DYN_CHIRP_ENABLE_SB API is issued at least 200 μ s before the start of the next frame

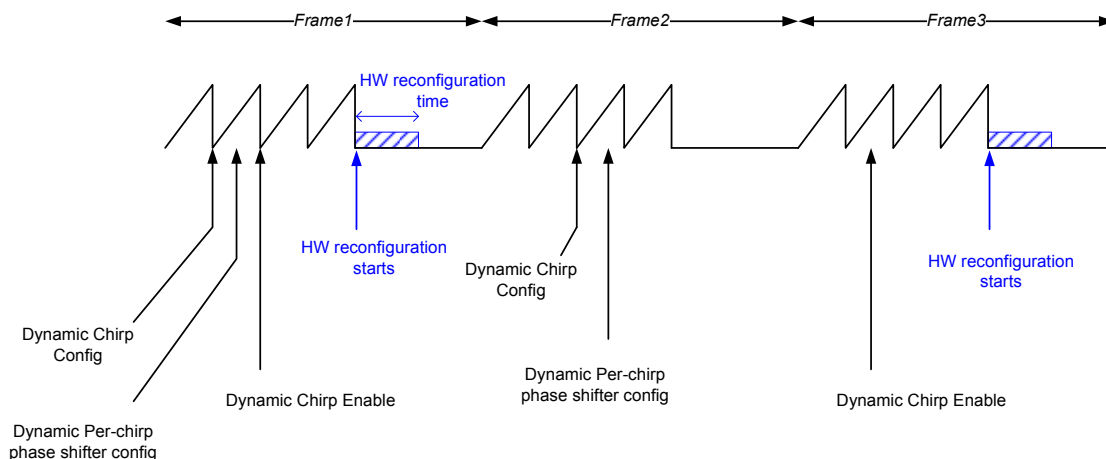


Figure 5.3: Dynamic chirp configuration use case timing diagram

5.5.19 Sub block 0x0112 – AWR_INTERCHIRP_BLOCKCONTROLS_SB

This API can be used to program the inter-chirp turn on and turn off times or various RF blocks.

NOTE: The inter-chirp timing control configuration API is supported in this release. Please refer latest DFP release note for more info.

Table 5.39: AWR_INTERCHIRP_BLOCKCONTROLS_SB contents

| Field Name | Number of bytes | Description |
|-----------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0112 |
| SBLKLEN | 2 | Value = 44 |
| RX02_RF_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off RX0 and RX2 RF stages. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX13_RF_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off RX1 and RX3 RF stages. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX02_BB_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off RX0 and RX2 baseband stages. 1 LSB = 10 ns Valid range: -1024 to 1023 |

Continued on next page

Table 5.39 – continued from previous page

| | | |
|---------------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------|
| RX13_BB_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off RX1 and RX3 baseband stages. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX02_RF_PRE_ENABLE_TIME | 2 | Time before TX Start Time when RX0 and RX2 RF stages are to be put in fast-charge state. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX13_RF_PRE_ENABLE_TIME | 2 | Time before TX Start Time when RX1 and RX3 RF stages are to be put in fast-charge state. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX02_BB_PRE_ENABLE_TIME | 2 | Time before TX Start Time when RX1 and RX3 baseband stages are to be put in fast-charge state. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX13_BB_PRE_ENABLE_TIME | 2 | Time before TX Start Time when RX2 and RX4 baseband stages are to be put in fast-charge state. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX02_RF_TURN_ON_TIME | 2 | Time before TX Start Time when RX1 and RX3 RF stages are to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX13_RF_TURN_ON_TIME | 2 | Time before TX Start Time when RX2 and RX4 RF stages are to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX02_BB_TURN_ON_TIME | 2 | Time before TX Start Time when RX1 and RX3 baseband stages are to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX13_BB_TURN_ON_TIME | 2 | Time before TX Start Time when RX2 and RX4 baseband stages are to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RX_LO_CHAIN_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off RX LO chain. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| TX_LO_CHAIN_TURN_OFF_TIME | 2 | Time to wait after ramp end before turning off TX LO chain. 1 LSB = 10 ns Valid range: -1024 to 1023 |

Continued on next page

Table 5.39 – continued from previous page

| | | |
|--------------------------|---|-----------------------------------------------------------------------------------------------------------------|
| RX_LO_CHAIN_TURN_ON_TIME | 2 | Time before TX Start Time when the RX LO chain is to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| TX_LO_CHAIN_TURN_ON_TIME | 2 | Time before TX Start Time when the TX LO chain is to be enabled. 1 LSB = 10 ns Valid range: -1024 to 1023 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

NOTE: The minimum inter-chirp time should be greater than maximum of the following

1. $\text{abs}(\text{RX02_RF_TURN_OFF_TIME}) + \max(\text{abs}(\text{RX02_RF_PRE_ENABLE_TIME}), \text{abs}(\text{RX02_RF_TURN_ON_TIME}))$
2. $\text{abs}(\text{RX13_RF_TURN_OFF_TIME}) + \max(\text{abs}(\text{RX13_RF_PRE_ENABLE_TIME}), \text{abs}(\text{RX13_RF_TURN_ON_TIME}))$
3. $\text{abs}(\text{RX02_BB_TURN_OFF_TIME}) + \max(\text{abs}(\text{RX02_BB_PRE_ENABLE_TIME}), \text{abs}(\text{RX02_BB_TURN_ON_TIME}))$
4. $\text{abs}(\text{RX13_BB_TURN_OFF_TIME}) + \max(\text{abs}(\text{RX13_BB_PRE_ENABLE_TIME}), \text{abs}(\text{RX13_BB_TURN_ON_TIME}))$
5. $\text{abs}(\text{RX_LO_TURN_OFF_TIME}) + \text{abs}(\text{RX_LO_TURN_ON_TIME})$
6. $\text{abs}(\text{TX_LO_TURN_OFF_TIME}) + \text{abs}(\text{TX_LO_TURN_ON_TIME})$

5.5.20 Sub block 0x0113 – AWR_SUBFRAME_START_CONF_SB

This API can be used to trigger each sub-frame individually in software triggered mode. This API takes effect only when the advanced frame configuration indicates that each sub-frame needs to be individually triggered by the user.

Table 5.40: AWR_SUBFRAME_START_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|-------|---------------------|--|---------------------|--|-----------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0113 | | | | | | | | |
| SBLKLEN | 2 | Value = 8 | | | | | | | | |
| START_CMD | 2 | <table border="0"> <tr> <td>Bits</td> <td>Definition</td> </tr> <tr> <td>b15:0</td> <td>Value Definition</td> </tr> <tr> <td></td> <td>0x0000 No effect</td> </tr> <tr> <td></td> <td>0x0001 Trigger next sub-frame in software triggered sub-frame mode</td> </tr> </table> | Bits | Definition | b15:0 | Value Definition | | 0x0000 No effect | | 0x0001 Trigger next sub-frame in software triggered sub-frame mode |
| Bits | Definition | | | | | | | | | |
| b15:0 | Value Definition | | | | | | | | | |
| | 0x0000 No effect | | | | | | | | | |
| | 0x0001 Trigger next sub-frame in software triggered sub-frame mode | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |

NOTE1: If the user wishes to trigger each sub-frame independently, then after advanced frame config, the FRAME START command should be issued once using AWR_FRAMESTARTSTOP_CONF_SB. This does not start any sub-frames but it will prepare the hardware for sub-frame trigger. Next any subsequent sub-frame trigger will start the sub-frames

NOTE2: If the user wishes to use sub-frame trigger, he has to ensure that sub-frame trigger command is issued $k \cdot N$ times where k is the number of sub-frames in each frame and N is the number of frames. If the user wishes to stop frames in between, then he has to issue the FRAME STOP command (using AWR_FRAMESTARTSTOP_CONF_SB) only after $k \cdot M$ triggers of sub-frame trigger command (where M is an integer). i.e. FRAME STOP command can be issued only at frame boundaries

NOTE3: If software based sub-frame trigger mode is chosen by the user, watchdog feature will not be available. User has to ensure that the watchdog is disabled before enabling the software based sub-frame trigger mode.

NOTE4: If sub-frame trigger or hardware trigger mode is used to trigger the frames/sub-frames and if frames need to be stopped before the specified number of frames, then the the FRAME_STOP command using [AWR_FRAMESTARTSTOP_CONF_SB](#) API should be issued while the frame is on-going. If the frames are stopped while the device is idle, it can lead to errors.

NOTE5: This API is not supported in IWR6843 ES1.0

5.6 Sub blocks related to AWR_RF_DYNAMIC_CONF_GET_SB

5.6.1 Sub block 0x0120 – AWR_PROFILE_CONF_GET_SB

This sub block reads the parameters of a given profile. The profile details are available as part of the acknowledgment. The structure is same as AWR_PROFILE_CONF_SET_SB

Table 5.41 describes the contents of this sub block.

Table 5.41: AWR_PROFILE_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|----------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0120 |
| SBLKLEN | 2 | Value = 8 |
| PROFILE_INDX | 2 | Valid range 0 to 3 Index of the profile which is to be read |
| RESERVED | 2 | 0x0000 |

5.6.2 Sub block 0x0121 – AWR_CHIRP_CONF_GET_SB

This sub block reads the parameters of a given chirp. The profile details are available as part of the acknowledgment. The structure is same as AWR_CHIRP_CONF_SET_SB

Table 5.42 describes the contents of this sub block.

Table 5.42: AWR_CHIRP_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------------|-----------------|-------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0121 |
| SBLKLEN | 2 | Value = 8 |
| CHIRP_START_INDX | 2 | Valid range 0 to 511 Starting index of the chirp which is to be read |
| CHIRP_END_INDX | 2 | Valid range 0 to 511 Ending index of the chirp which is to be read |

5.6.3 Sub block 0x0122 – AWR_FRAME_CONF_GET_SB

This sub block reads the parameters of the configured frame. The profile details are available as part of the acknowledgement. The structure is same as AWR_FRAME_CONF_SET_SB

Table 5.43 describes the contents of this sub block.

Table 5.43: AWR_FRAME_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0122 |
| SBLKLEN | 2 | Value = 4 |

5.6.4 Sub block 0x0123 – RESERVED

5.6.5 Sub block 0x0124 – RESERVED

5.6.6 Sub block 0x0125 – AWR_ADV_FRAME_CONF_GET_SB

This sub block reads the parameters of the configured frame. The profile details are available as part of the acknowledgement. The structure is same as AWR_ADVANCED_FRAME_CONF_SET_SB

Table 5.44 describes the contents of this sub block.

Table 5.44: AWR_ADV_FRAME_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0125 |
| SBLKLEN | 2 | Value = 4 |

5.6.7 Sub block 0x0126 – RESERVED

5.6.8 Sub block 0x0127 – RESERVED

5.6.9 Sub block 0x0128 – RESERVED

5.6.10 Sub block 0x0129 – RESERVED

5.6.11 Sub block 0x012A – RESERVED

5.6.12 Sub block 0x012B – RESERVED

5.6.13 Sub block 0x012C – AWR_RX_GAIN_TEMPLUT_GET_SB

This API is issued to read the temperature based RX gain LUT used by the firmware. This API should be issued after the profile configuration API. The acknowledgement packet sent in response to this API will contain the LUT. The structure is same as AWR_RX_GAIN_LUT_SET_SB.

Table 5.45: AWR_RX_GAIN_TEMPLUT_GET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|----------------------------------------------------|
| SBLKID | 2 | Value = 0x012C |
| SBLKLEN | 2 | Value = 8 |
| PROFILE.INDX | 1 | Profile index for which the RX gain LUT is desired |
| RESERVED | 3 | 0x000000 |

NOTE4: This API is not supported in IWR6843 ES1.0

5.6.14 Sub block 0x012D – AWR_TX_GAIN_TEMPLUT_GET_SB

This API is issued to read the temperature based TX gain LUT used by the firmware. This API should be issued after the profile configuration API. The acknowledgement packet sent in response to this API will contain the LUT. The structure is same as AWR_TX_GAIN_LUT_SET_SB.

Table 5.46: AWR_TX_GAIN_TEMPLUT_GET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|----------------------------------------------------|
| SBLKID | 2 | Value = 0x012D |
| SBLKLEN | 2 | Value = 8 |
| PROFILE.INDX | 1 | Profile index for which the TX gain LUT is desired |
| RESERVED | 3 | 0x000000 |

NOTE: This API is not supported in IWR6843 ES1.0

5.7 Sub blocks related to AWR_FRAME_TRIG_MSG

5.7.1 Sub block 0x0140 – AWR_FRAMESTARTSTOP_CONF_SB

This sub block starts or stops transmission of frames.
 Table 5.47 describes the contents of this sub block.

Table 5.47: AWR_FRAMESTARTSTOP_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0140 |
| SBLKLEN | 2 | Value = 8 |
| START_STOP_CMD | 2 | Value Definition 0x0000 Stop the transmission of frames after the current frame is over 0x0001 Trigger a frame in software triggered mode. In hardware SYNC.IN triggered mode, this command allows subsequent SYNC.IN trigger to be honored |
| RESERVED | 2 | 0x0000 |

NOTE: When Frame Stop command is sent to RadarSS, the frame will be stopped after completing all the chirps of a Frame/Advance frame. In non periodic Hw triggered mode if frame stop command is issued when Hw pulses are paused (i.e after completing previous frame) then a Hw pulse is required to trigger next frame/bursts and frame will be stopped at the end of this triggered frame. In Hw triggered mode, the forced frame stop is not supported, the frame end command is honored only if frames are actively running.

5.8 Sub blocks related to AWR_RF_ADVANCED_FEATURES_CONF_SET_MSG

5.8.1 Sub block 0x0180 – AWR_BPM_COMMON_CONF_SET_SB

This API sub block defines static configurations related to BPM (Binary Phase Modulation) feature in each of the TXs. E.g. the source of the BPM pattern (one constant value for each chirp as defined, or intra-chirp pseudo random BPM pattern as found by a programmable LFSR or a programmable sequence inside each chirp), are defined here.

Table 5.48 describes the contents of this sub block.

Table 5.48: AWR_BPM_COMMON_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0180 |

Continued on next page

Table 5.48 – continued from previous page

| | | | | | | | | | | | | | | | | | | |
|--------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|------|--------------------------------------------|-------|------------|----|---------------------------------------------------|----|----------|----|----------|----|----------|-------|----------|
| SBLKLEN | 2 | Value = 20 | | | | | | | | | | | | | | | | |
| BPM.MODE_CFG | 2 | <table border="0"> <tr> <td>Bits</td> <td>Description</td> </tr> <tr> <td>b1:0</td> <td>BPM_SRC_SEL (select source of BPM pattern)</td> </tr> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>00</td> <td>CHIRP_CONFIG_BPM (refer to AWR_BPM_CHIRP_CONF_SB)</td> </tr> <tr> <td>01</td> <td>RESERVED</td> </tr> <tr> <td>10</td> <td>RESERVED</td> </tr> <tr> <td>11</td> <td>RESERVED</td> </tr> <tr> <td>b15:2</td> <td>RESERVED</td> </tr> </table> | Bits | Description | b1:0 | BPM_SRC_SEL (select source of BPM pattern) | Value | Definition | 00 | CHIRP_CONFIG_BPM (refer to AWR_BPM_CHIRP_CONF_SB) | 01 | RESERVED | 10 | RESERVED | 11 | RESERVED | b15:2 | RESERVED |
| Bits | Description | | | | | | | | | | | | | | | | | |
| b1:0 | BPM_SRC_SEL (select source of BPM pattern) | | | | | | | | | | | | | | | | | |
| Value | Definition | | | | | | | | | | | | | | | | | |
| 00 | CHIRP_CONFIG_BPM (refer to AWR_BPM_CHIRP_CONF_SB) | | | | | | | | | | | | | | | | | |
| 01 | RESERVED | | | | | | | | | | | | | | | | | |
| 10 | RESERVED | | | | | | | | | | | | | | | | | |
| 11 | RESERVED | | | | | | | | | | | | | | | | | |
| b15:2 | RESERVED | | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | |

NOTE: This API is not supported in IWR6843 ES1.0

5.8.2 Sub block 0x0181 – AWR_BPM_CHIRP_CONF_SET_SB

This sub block defines static configurations related to BPM (Binary Phase Modulation) feature in each of the TXs.

Table 5.49 describes the contents of this sub block.

Table 5.49: AWR_BPM_CHIRP_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|-----------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0181 |
| SBLKLEN | 2 | Value = 12 |
| CHIRP_START_IND_X | 2 | Start index of the chirp for configuring the constant BPM Valid range 0 to 511 |
| CHIRP_END_IND_X | 2 | End index of the chirp for configuring the constant BPM Valid range 0 to 511 |

Continued on next page

Table 5.49 – continued from previous page

| | | | |
|---------------|---|--------|---------------------------------------------------------------------------------------------|
| CONST_BPM_VAL | 2 | Bit | Definition |
| | | b0 | CONST_BPM_VAL_TX0_TXOFF Value of Binary Phase Shift value for TX0, when during idle time |
| | | b1 | CONST_BPM_VAL_TX0_TXON Value of Binary Phase Shift value for TX0, during chirp |
| | | b2 | CONST_BPM_VAL_TX1_TXOFF Value of Binary Phase Shift value for TX1, when during idle time |
| | | b3 | CONST_BPM_VAL_TX1_TXON Value of Binary Phase Shift value for TX1, during chirp |
| | | b4 | CONST_BPM_VAL_TX2_TXOFF Value of Binary Phase Shift value for TX2, when during idle time |
| | | b5 | CONST_BPM_VAL_TX2_TXON Value of Binary Phase Shift value for TX2, during chirp |
| | | b15:6 | RESERVED |
| RESERVED | 2 | 0x0000 | |

NOTE1: BPM values are applied at TX_START_TIME.

NOTE2: This API is not supported in IWR6843 ES1.0

5.9 Sub blocks related to AWR_RF_STATUS_GET_MSG

5.9.1 Sub block 0x0220 – AWR_RF_VERSION_GET_SB

This sub block reads RF HW and FW versions. The information returned by the device will be in the format as given in AWR_RFVERSION_SB.

Table 5.50 describes the contents of the request sub block

Table 5.50: AWR_RF_VERSION_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0220 |
| SBLKLEN | 2 | Value = 4 |

Response to AWR_RFVERSION_GET_SB

AWR_RFVERSION_SB sub block is sent by the radar device in response to AWR_RFVERSION_GET_SB. Note that SBLKID for both AWR_RFVERSION_GET_SB and AWR_RFVERSION_SB are same.

Table 5.51 describes the contents of the response sub block.

Table 5.51: AWR_RF_VERSION_SB response contents

| Field Name | Number of bytes | Description |
|----------------------------|-----------------|-----------------------------------|
| SBLKID | 2 | Value = 0x0220 |
| SBLKLEN | 2 | Value = 20 |
| HW_VARIANT | 1 | HW variant number |
| HW_VERSION_MAJOR | 1 | HW version major number |
| HW_VERSION_MINOR | 1 | HW version minor number |
| BSS_FW_VERSION_MAJOR | 1 | BSS FW version major number |
| BSS_FW_VERSION_MINOR | 1 | BSS FW version minor number |
| BSS_FW_VERSION_BUILD | 1 | BSS FW version build number |
| BSS_FW_VERSION_DEBUG | 1 | BSS FW version debug number |
| BSS_FW_VERSION_YEAR | 1 | Year of BSS FW version release |
| BSS_FW_VERSION_MONTH | 1 | Month of BSS FW version release |
| BSS_FW_VERSION_DAY | 1 | Day of BSS FW version release |
| BSS_FW_VERSION_PATCH_MAJOR | 1 | BSS FW version patch major number |
| BSS_FW_VERSION_PATCH_MINOR | 1 | BSS FW version patch minor number |
| BSS_FW_VERSION_PATCH_YEAR | 1 | Year of BSS FW patch release |

Continued on next page

Table 5.51 – continued from previous page

| | | | | | | | | |
|----------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|------|----------------------|------|----------------------|
| BSS_FW_VERSION_PATCH_MONTH | 1 | Month of BSS FW patch release | | | | | | |
| BSS_FW_VERSION_PATCH_DAY | 1 | Day of BSS FW patch release | | | | | | |
| BSS_FW_PATCH_BUILD_DEBUG_VERSION | 1 | <table border="0"> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>b3:0</td> <td>DEBUG version number</td> </tr> <tr> <td>b7:4</td> <td>BUILD version number</td> </tr> </table> | Bit | Definition | b3:0 | DEBUG version number | b7:4 | BUILD version number |
| Bit | Definition | | | | | | | |
| b3:0 | DEBUG version number | | | | | | | |
| b7:4 | BUILD version number | | | | | | | |

5.9.2 Sub block 0x0221 – AWR_RF_CPUFAULT_STATUS_GET_SB

This sub block provides the RF BSS CPU fault information.

Table 5.52 describes the content of this sub block.

Table 5.52: AWR_RF_CPUFAULT_STATUS_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0221 |
| SBLKLEN | 2 | Value = 4 |

AWR_RF_CPUFAULT_STATUS_SB is sent in response to AWR_RF_CPUFAULT_STATUS_GET_SB.

Table 5.53 describes the content of AWR_RF_CPUFAULT_STATUS_SB

Table 5.53: AWR_RF_CPUFAULT_STATUS_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0221 |
| SBLKLEN | 2 | Value = 36 |

Continued on next page

Table 5.53 – continued from previous page

| | | | |
|----------------------|---|-----------------------------------------------------------------------------------------------------------|------------------------------------------|
| FAULT_TYPE | 1 | Value | Definition |
| | | 0 | RF Processor Undefined Instruction Abort |
| | | 1 | RF Processor Instruction pre-fetch Abort |
| | | 2 | RF Processor Data Access Abort |
| | | 3 | RF Processor Firmware Fatal Error |
| | | 0x4 - 0xFE | RESERVED |
| | | 0xFF | No fault |
| RESERVED | 1 | 0x00 | |
| LINE_NUM | 2 | Valid only in case of FAULT type is 0x3, provides the firmware line number at which fatal error occurred. | |
| FAULT_LR | 4 | The instruction PC address at which Fault occurred | |
| FAULT_PREV_LR | 4 | The return address of the function from which fault function has been called (Call stack LR) | |
| FAULT_SPSR | 4 | The CPSR register value at which fault occurred | |
| FAULT_SP | 4 | The SP register value at which fault occurred | |
| FAULT_CAUSE_ADDRESS | 4 | The address access at which Fault occurred (valid only for fault type 0x0 to 0x2) | |
| FAULT_ERROR_STATUS | 2 | The status of Error (Error Cause type – valid only for fault type 0x0 to 0x2) | |
| | | 0x000 | BACKGROUND_ERR |
| | | 0x001 | ALIGNMENT_ERR |
| | | 0x002 | DEBUG_EVENT |
| | | 0x00D | PERMISSION_ERR |
| | | 0x008 | SYNCH_EXTER_ERR |
| | | 0x406 | ASYNCH_EXTER_ERR |
| | | 0x409 | SYNCH_ECC_ERR |
| | | 0x408 | ASYNCH_ECC_ERR |
| FAULT_ERROR_SOURCE | 1 | The Source of the Error (Error Source type - valid only for fault type 0x0 to 0x2) | |
| | | 0x0 | ERR_SOURCE_AXI_MASTER |
| | | 0x1 | ERR_SOURCE_ATCM |
| | | 0x2 | ERR_SOURCE_BTCM |
| FAULT_AXI_ERROR_TYPE | 1 | The AXI Error type (Error Source type - valid only for fault type 0x0 to 0x2) | |
| | | 0x0 | AXI_DECOD_ERR |
| | | 0x1 | AXI_SLAVE_ERR |

Continued on next page

Table 5.53 – continued from previous page

| | | |
|---------------------|---|------------------------------------------------------------------------------------------------------------------------|
| FAULT_ACCESS_TYPE | 1 | The Error Access type (Error Access type - valid only for fault type 0x0 to 0x2) 0x0 READ_ERR 0x1 WRITE_ERR |
| FAULT_RECOVERY_TYPE | 1 | The Error Recovery type (Error Recovery type - Valid only for fault type 0x0 to 0x2) 0x0 UNRECOVERY 0x1 RECOVERY |
| RESERVED | 2 | 0x0000 |

5.9.3 Sub block 0x0222 – AWR_RF_ESMFAULT_STATUS_GET_SB

This sub block provides the information regarding additional RF sub system faults. Table 5.54 describes the content of this sub block.

Table 5.54: AWR_RF_ESMFAULT_STATUS_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0222 |
| SBLKLEN | 2 | Value = 4 |

The response to above request is given in the AWR_RF_ESMFAULT_STATUS_SB. Table 5.55 describes the contents of AWR_RF_ESMFAULT_STATUS_SB.

Table 5.55: AWR_RF_ESMFAULT_STATUS_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0222 |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 5.55 – continued from previous page

| | | | |
|--------------------|----------|-----|----------------------------|
| ESM.GROUP1_ ERRORS | 4 | Bit | Error Information |
| | | b0 | RAMPGEN_SB_ERROR |
| | | b1 | RESERVED |
| | | b2 | GPADC_RAM_SB_ERROR |
| | | b3 | VIM_RAM_SB_ERROR |
| | | b4 | DFE_SELFTEST_ERROR |
| | | b5 | VIM_SELFTEST_ERROR |
| | | b6 | B0TCM_SB_ERROR |
| | | b7 | B1TCM_SB_ERROR |
| | | b8 | CCMR4_SELFTEST_ERROR |
| | | b9 | ATCM_SB_ERROR |
| | | b10 | RAMPGEN_SELFTEST_ERROR |
| | | b11 | RAMPGEN_PAR_SELFTEST_ERROR |
| | | b12 | SEQ_EXT_SELFTEST_ERROR |
| | | b13 | SEQ_EXT_SB_ERROR |
| | | b14 | RESERVED |
| | | b15 | AGC_RAM_SB_ERROR |
| | | b16 | B1TCM_PAR_CHK_ERROR |
| | | b17 | B0TCM_PAR_CHK_ERROR |
| | | b18 | ATCM_PAR_CHK_ERROR |
| | | b19 | MB_MSS2BSS_SB_ERROR |
| | | b20 | MB_BSS2MSS_SB_ERROR |
| b31:21 | RESERVED | | |

Continued on next page

Table 5.55 – continued from previous page

| | | | |
|-----------------------|-----------|-----|------------------------|
| ESM.GROUP2_ ERRORS | 4 | Bit | Error Information |
| | | b0 | DFE.STC.ERROR |
| | | b1 | CR4.STC.ERROR |
| | | b2 | CCMR4.COMP.ERROR |
| | | b3 | B0TCM.DB.ERROR |
| | | b4 | B1TCM.DB.ERROR |
| | | b5 | ATCM.DB.ERROR |
| | | b6 | DCC.ERROR |
| | | b7 | SEQ.EXT.ERROR |
| | | b8 | SYNT.FREQ.MON.ERROR |
| | | b9 | DFE.PARITY.ERROR |
| | | b10 | RAMPGEN.DB.ERROR |
| | | b11 | BUBBLE.CORRECTION.FAIL |
| | | b12 | RAMPGEN.LOCSTEP.ERROR |
| | | b13 | RTI.RESET.ERROR |
| | | b14 | GPADC.RAM.DB.ERROR |
| | | b15 | VIM.COMP.ERROR |
| | | b16 | CR4.LIVE.LOCK.ERROR |
| | | b17 | WDT.NMI.ERROR |
| | | b18 | VIM.RAM.DB.ERROR |
| | | b19 | RAMPGEN.PAR.ERROR |
| | | b20 | SEQ.EXT.DB.ERROR |
| | | b21 | DMA.MPU.ERROR |
| | | b22 | AGC.RAM.DB.ERROR |
| | | b23 | CRC.COMP.ERROR |
| | | b24 | WAKEUP.STS.ERROR |
| | | b25 | SHORT.CIRCUIT.ERROR |
| | | b26 | B1TCM.PAR.ERROR |
| | | b27 | B0TCM.PAR.ERROR |
| | | b28 | ATCM.PAR.ERROR |
| | | b29 | MB.MSS2BSS.DB.ERROR |
| | | b30 | MB.BSS2MSS.DB.ERROR |
| b31 | CCC.ERROR | | |

5.9.4 Sub block 0x0223 – AWR_RF_DIEID_GET_SB

This sub block provides the information regarding the Die ID of the device.

Table 5.56: AWR_RF_DIEID_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0223 |
| SBLKLEN | 2 | Value = 4 |

The response to above request is given in the AWR_RF_DIEID_STATUS_SB. Table 5.57 describes the contents of AWR_RF_DIEID_STATUS_SB.

Table 5.57: AWR_RF_DIEID_STATUS_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-------------------------------------|
| SBLKID | 2 | Value = 0x0223 |
| SBLKLEN | 2 | Value = 36 |
| LOT_NO | 4 | Lot number |
| WAFER_NO | 4 | Wafer number |
| DEV_X | 4 | X cordinate of the die in the wafer |
| DEV_Y | 4 | Y cordinate of the die in the wafer |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

NOTE: Die ID read API is not supported in IWR6843 ES1.0

5.9.5 Sub block 0x0224 – AWR_RF_BOOTUPBIST_STATUS_GET_SB

This sub block provides the information regarding boot up self-test status. Table 5.58 describes the content of this sub block.

Table 5.58: AWR_RF_BOOTUPBIST_STATUS_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0224 |
| SBLKLEN | 2 | Value = 4 |

The response of this sub block will be AWR_RF_BOOTUPBIST_STATUS_DATA_SB with content as shown in Table 5.59

Table 5.59: AWR_RF_BOOTUPBIST_STATUS_DATA_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0224 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.59 – continued from previous page

| | | |
|--------------------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RF_POWERUP_ BIST_STATUS_ FLAGS | 4 | 1 - PASS, 0 - FAIL Bit Status Information b0 ROM CRC check b1 CR4 and VIM lockstep test b2 RESERVED b3 VIM test (Not supported in 1st Gen devices, refer latest release note) b4 STC test of diagnostic b5 CR4 STC b6 CRC test b7 RAMPGEN memory ECC test b8 DFE Parity test b9 DFE memory ECC b10 RAMPGEN lockstep test b11 FRC lockstep test b12 DFE memory PBIST b13 RAMPGEN memory PBIST b14 PBIST test b15 WDT test b16 ESM test b17 DFE STC b18 RESERVED b19 ATCM, BTCM ECC test b20 ATCM, BTCM parity test b21 RESERVED b22 RESERVED b23 RESERVED b24 FFT test b25 RTI test b26 PCR test b31:27 RESERVED |
| POWERUP_TIME | 4 | RF BIST SS power up time 1 LSB = 5 ns |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

NOTE: Bootup digital monitoring status are not applicable for QM devices

5.10 Sub blocks related to AWR_RF_MONITORING_REPORT_GET_MSG

5.10.1 Sub block 0x0260 – AWR_RF_DFE_STATISTICS_REPORT_GET_SB

NOTE: The DFE statistics GET API is not supported in this release. Please refer latest DFP release note for more info.

Table 5.60 describes the content of this sub block.

Table 5.60: AWR_RF_DFE_STATISTICS_REPORT_GET_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x0260 |
| SBLKLEN | 2 | Value = 4 |

The response of this sub block will be AWR_RF_DFE_STATISTICS_REPORT_SB with content as shown in Table 5.61

Table 5.61: AWR_RF_DFE_STATISTICS_REPORT_SB response contents

| Field Name | Number of bytes | Description |
|-------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x0260 |
| SBLKLEN | 2 | Value = 196 |
| PF0_RX0_ICH | 2 | Residual DC value in I chain for profile 0, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX0_QCH | 2 | Residual DC value in Q chain for profile 0, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX0_ISQ | 2 | RMS power in I chain for profile 0, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX0_QSQ | 2 | RMS power in Q chain for profile 0, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF0_RX0_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 0, RX channel 0 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF0_RX1_ICH | 2 | Residual DC value in I chain for profile 0, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX1_QCH | 2 | Residual DC value in Q chain for profile 0, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX1_ISQ | 2 | RMS power in I chain for profile 0, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX1_QSQ | 2 | RMS power in Q chain for profile 0, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX1_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 0, RX channel 1 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF0_RX2_ICH | 2 | Residual DC value in I chain for profile 0, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX2_QCH | 2 | Residual DC value in Q chain for profile 0, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX2_ISQ | 2 | RMS power in I chain for profile 0, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX2_QSQ | 2 | RMS power in Q chain for profile 0, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF0_RX2_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 0, RX channel 2 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF0_RX3_ICH | 2 | Residual DC value in I chain for profile 0, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX3_QCH | 2 | Residual DC value in Q chain for profile 0, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF0_RX3_ISQ | 2 | RMS power in I chain for profile 0, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX3_QSQ | 2 | RMS power in Q chain for profile 0, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF0_RX3_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 0, RX channel 3 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF1_RX0_ICH | 2 | Residual DC value in I chain for profile 1, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX0_QCH | 2 | Residual DC value in Q chain for profile 1, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX0_ISQ | 2 | RMS power in I chain for profile 1, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX0_QSQ | 2 | RMS power in Q chain for profile 1, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF1_RX0_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 1, RX channel 0 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF1_RX1_ICH | 2 | Residual DC value in I chain for profile 1, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX1_QCH | 2 | Residual DC value in Q chain for profile 1, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX1_ISQ | 2 | RMS power in I chain for profile 1, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX1_QSQ | 2 | RMS power in Q chain for profile 1, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX1_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 1, RX channel 1 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF1_RX2_ICH | 2 | Residual DC value in I chain for profile 1, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX2_QCH | 2 | Residual DC value in Q chain for profile 1, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX2_ISQ | 2 | RMS power in I chain for profile 1, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX2_QSQ | 2 | RMS power in Q chain for profile 1, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF1_RX2_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 1, RX channel 2 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF1_RX3_ICH | 2 | Residual DC value in I chain for profile 1, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX3_QCH | 2 | Residual DC value in Q chain for profile 1, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF1_RX3_ISQ | 2 | RMS power in I chain for profile 1, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX3_QSQ | 2 | RMS power in Q chain for profile 1, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF1_RX3_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 1, RX channel 3 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF2_RX0_ICH | 2 | Residual DC value in I chain for profile 2 RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX0_QCH | 2 | Residual DC value in Q chain for profile 2, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX0_ISQ | 2 | RMS power in I chain for profile 2, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX0_QSQ | 2 | RMS power in Q chain for profile 2, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF2_RX0_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 2, RX channel 0 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF2_RX1_ICH | 2 | Residual DC value in I chain for profile 2, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX1_QCH | 2 | Residual DC value in Q chain for profile 2, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX1_ISQ | 2 | RMS power in I chain for profile 2, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX1_QSQ | 2 | RMS power in Q chain for profile 2, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX1_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 2, RX channel 1 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF2_RX2_ICH | 2 | Residual DC value in I chain for profile 2, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX2_QCH | 2 | Residual DC value in Q chain for profile 2, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX2_ISQ | 2 | RMS power in I chain for profile 2, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX2_QSQ | 2 | RMS power in Q chain for profile 2, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF2_RX2_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 2, RX channel 2 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF2_RX3_ICH | 2 | Residual DC value in I chain for profile 2, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX3_QCH | 2 | Residual DC value in Q chain for profile 2, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF2_RX3_ISQ | 2 | RMS power in I chain for profile 2, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX3_QSQ | 2 | RMS power in Q chain for profile 2, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF2_RX3_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 2, RX channel 3 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF3_RX0_ICH | 2 | Residual DC value in I chain for profile 3 RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX0_QCH | 2 | Residual DC value in Q chain for profile 3, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX0_ISQ | 2 | RMS power in I chain for profile 3, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX0_QSQ | 2 | RMS power in Q chain for profile 3, RX channel 0 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF3_RX0_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 3, RX channel 0 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF3_RX1_ICH | 2 | Residual DC value in I chain for profile 3, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX1_QCH | 2 | Residual DC value in Q chain for profile 3, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX1_ISQ | 2 | RMS power in I chain for profile 3, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX1_QSQ | 2 | RMS power in Q chain for profile 3, RX channel 1 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX1_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 3, RX channel 1 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF3_RX2_ICH | 2 | Residual DC value in I chain for profile 3, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX2_QCH | 2 | Residual DC value in Q chain for profile 3, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX2_ISQ | 2 | RMS power in I chain for profile 3, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX2_QSQ | 2 | RMS power in Q chain for profile 3, RX channel 2 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |

Continued on next page

Table 5.61 – continued from previous page

| | | |
|-----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PF3_RX2_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 3, RX channel 2 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |
| PF3_RX3_ICH | 2 | Residual DC value in I chain for profile 3, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX3_QCH | 2 | Residual DC value in Q chain for profile 3, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit signed number 1 LSB = $1V/2^{15}$ referred to ADC input |
| PF3_RX3_ISQ | 2 | RMS power in I chain for profile 3, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX3_QSQ | 2 | RMS power in Q chain for profile 3, RX channel 3 (post DC and IQ mismatch correction) represented by a 16 bit unsigned number 1 LSB = $1V^2/2^{15}$ referred to ADC input |
| PF3_RX3_IQ-CORR | 4 | Cross correlation between I and Q chains for profile 3, RX channel 3 (post DC and IQ mismatch correction) represented by a 32 bit signed number 1 LSB = $1V^2/2^{30}$ referred to ADC input |

5.11 Sub blocks related to AWR_RF_MISC_CONF_SET_MSG

5.11.1 Sub block 0x02C0 – RESERVED

5.11.2 Sub block 0x02C1 – RESERVED

5.11.3 Sub block 0x02C2 – AWR_RF_TEST_SOURCE_CONFIG_SET_SB

This sub block is used to configure the test source of BSS

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------|
| NOTE: | The test source configuration APIs are supported only for debug purpose. Please refer latest DFP release note for more info. |
|--------------|------------------------------------------------------------------------------------------------------------------------------|

Table 5.62 describes the content of this sub block.

Table 5.62: AWR_RF_TEST_SOURCE_CONFIG_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x02C2 |
| SBLKLEN | 2 | Value = 72 |
| POSITION.VEC1 | [2+2+2] | Relative position in Cartesian coordinate from radar to objects, [x, y, z] (all signed, though for y, only unsigned makes sense if forward looking: our radar is on y=0 plane). Object 0 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ± 32767 cm |
| VELOCITY.VEC1 | [2+2+2] | Relative velocity in Cartesian coordinate, similar to position vector (all signed) Object 0 1 LSB = 1 cm/s Valid Range = +/- 5000 (i.e. +/-180 kmph) |
| SIG.LEV.VEC1 | [2] | Reflecting objects' signal level at ADC output, relative to ADC Full Scale 1 LSB = -0.1 dBFS Valid range: 0 to 950 The same field may be used to emulate enable/disable each object by programming appropriate levels. |
| BOUNDARY_MIN.VEC1 | [2+2+2] | Boundary minimum limit for each of x, y, z. When the current position crosses this boundary, the emulator returns the corresponding coordinate to the originally programmed value. Object 1 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ± 32767 cm |
| BOUNDARY_MAX.VEC1 | [2+2+2] | Boundary maximum limit for each of x, y, z. When the current position crosses this boundary, the emulator returns the corresponding coordinate to the originally programmed value. Object 1 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ± 32767 cm |
| POSITION.VEC2 | [2+2+2] | Relative position in Cartesian coordinate from radar to objects, [x, y, z] (all signed, though for y, only unsigned makes sense if forward looking: our radar is on y=0 plane). Object 1 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ± 32767 cm |

Continued on next page

Table 5.62 – continued from previous page

| | | |
|-------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VELOCITY_VEC2 | [2+2+2] | Relative velocity in Cartesian coordinate, similar to position vector (all signed) Object 1 1 LSB = 1 cm/s Valid Range = +/- 5000 (i.e. +/-180 kmph) |
| SIG_LEV_VEC2 | [2] | Reflecting objects' signal level at ADC output, relative to ADC Full Scale 1 LSB = -0.1 dBFS Valid range: 0 to 950 The same field may be used to emulate enable/disable each object by programming appropriate levels. |
| BOUNDARY_MIN_VEC2 | [2+2+2] | Boundary minimum limit for each of x, y, z. When the current position crosses this boundary, the emulator returns the corresponding coordinate to the originally programmed value. Object 1 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ±32767 cm |
| BOUNDARY_MAX_VEC2 | [2+2+2] | Boundary maximum limit for each of x, y, z. When the current position crosses this boundary, the emulator returns the corresponding coordinate to the originally programmed value. Object 1 [x,y,z] 1 LSB = 1 cm Valid Range: y: 0 to 32767 cm, x & z: ±32767 cm |
| RX_ANT_POS_XZ | 8 | Receiver Antenna positions to be modeled. The radar is on y=0 plane. Only x and z coordinates to be provided. 1 LSB = Wavelength/8 Valid range = ±15 wave lengths Byte 0: RX0 X coordinate (may be 0 as reference) Byte 1: RX0 Z (may be 0 as reference) Byte 2: RX1 X Byte 3: RX1 Z Byte 4: RX2 X Byte 5: RX2 Z Byte 6: RX3 X Byte 7: RX3 Z |
| RESERVED | 6 | RESERVED |
| RESERVED | 2 | Reserved for 4 bytes alignment |

NOTE: Test source is not characterized and tuned to 60GHz in IWR6843 ES1.0 device.

5.11.4 Sub block 0x02C3 – AWR_RF_TEST_SOURCE_ENABLE_SET_SB

This sub block is used to enable test source of BSS
 Table 5.63 describes the content of this sub block.

Table 5.63: AWR_RF_TEST_SOURCE_ENABLE_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x02C3 |
| SBLKLEN | 2 | Value = 8 |
| TS_EN | 2 | Bit Definition b0 0 Disable (revert to normal functionality) 1 Enable (enter test source functionality) b15:1 RESERVED |
| RESERVED | 2 | 0x0000 |

5.11.5 Sub block 0x02C4 – 0x02CB RESERVED

5.11.6 Sub block 0x02CC – AWR_RF_LDO_BYPASS_SB

This sub block enables LDO bypass option within BSS.

CAUTION: Do not enable RF LDO bypass option when the PMIC is configured to supply 1.3V to VIN_13RF1 and VIN_13RF2 analog and RF power supply inputs. This may damage the device. Typically in TI EVMs, PMIC is configured to supply 1.3V to the RF supplies.

Table 5.64 describes the content of this sub block.

Table 5.64: AWR_RF_LDO_BYPASS_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x02CC |
| SBLKLEN | 2 | Value = 8 |

Continued on next page

Table 5.64 – continued from previous page

| RFLDO_BYPASS_EN | 2 | <p>Bit Description</p> <p>b0 Value Description</p> <p>0 RF LDO not bypassed</p> <p>1 RF LDO bypassed</p> <p>b1 Value Description</p> <p>0 PA LDO enabled</p> <p>1 PA LDO disabled</p> <p>When simultaneous 3 TX are to be used, to avoid package reliability issues, VIN_13RF2 is shorted to VOUT_PA on the board and the PA LDO should be disabled.</p> <p>b15:2 RESERVED</p> <p>The usage of these configurations is as per the table below</p> <table border="1" data-bbox="662 817 1276 1160"> <thead> <tr> <th>USECASE</th> <th>LDO_BYPASS</th> <th>PA_LDO_DISABLE</th> </tr> </thead> <tbody> <tr> <td>1.3V VIN_13RF1 and VIN_13RF2 supplies</td> <td>0</td> <td>0</td> </tr> <tr> <td>1.0V VIN_13RF1 and VIN_13RF2 supplies</td> <td>1</td> <td>0</td> </tr> <tr> <td>1.0V VIN_13RF1 and VIN_13RF2 supplies and VIN_13RF2 shorted to VOUT_PA</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | USECASE | LDO_BYPASS | PA_LDO_DISABLE | 1.3V VIN_13RF1 and VIN_13RF2 supplies | 0 | 0 | 1.0V VIN_13RF1 and VIN_13RF2 supplies | 1 | 0 | 1.0V VIN_13RF1 and VIN_13RF2 supplies and VIN_13RF2 shorted to VOUT_PA | 1 | 1 |
|------------------------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|------------|----------------|---------------------------------------|---|---|---------------------------------------|---|---|------------------------------------------------------------------------|---|---|
| USECASE | LDO_BYPASS | PA_LDO_DISABLE | | | | | | | | | | | | |
| 1.3V VIN_13RF1 and VIN_13RF2 supplies | 0 | 0 | | | | | | | | | | | | |
| 1.0V VIN_13RF1 and VIN_13RF2 supplies | 1 | 0 | | | | | | | | | | | | |
| 1.0V VIN_13RF1 and VIN_13RF2 supplies and VIN_13RF2 shorted to VOUT_PA | 1 | 1 | | | | | | | | | | | | |
| SUPPLY_MONITOR_IRDROP | 1 | <p>IR drop is the voltage drop from the PMIC output to the device pin. The user should program the voltage drop in percentage units which will be used for adjusting the thresholds for measuring the external supplies.</p> <p>Value Description</p> <p>0 IR drop of 0%</p> <p>1 IR drop of 3%</p> <p>2 IR drop of 6%</p> <p>3 IR drop of 9%</p> | | | | | | | | | | | | |
| IO_SUPPLY_INDICATOR | 1 | <p>IO supply indicator for correct monitoring of IO supply</p> <p>Value Description</p> <p>0 3.3 V IO supply</p> <p>1 1.8 V IO supply</p> | | | | | | | | | | | | |

5.11.7 Sub block 0x02CD – AWR_RF_PALOOPBACK_CFG_SB

This sub block enables/disables PA loopback for all enabled profiles. This is used to debug both the TX and RX chains are working correctly.

NOTE: The PA loop-back configuration API is supported only for debug purpose. Please refer latest DFP release note for more info.

Table 5.65 describes the content of this sub block.

NOTE: If monitoring is enabled with the loopback APIs (subblock 0x02CD, 0x02CE, 0x02CF), then loopback will not work after monitoring is complete. To use loopback with monitoring, use AWR_ADVANCED_FRAME_CONF_SB with AWR_LOOPBACK_BURST_CONF_SB.

Table 5.65: AWR_RF_PALOOPBACK_CFG_SB contents

| Field Name | Number of bytes | Description |
|------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x02CD |
| SBLKLEN | 2 | Value = 8 |
| PA_LOOPBACK_FREQ | 2 | This value is a 100 MHz divider which sets the loopback frequency For e.g. for a 1 MHz frequency, set this to 100 For a 2 MHz frequency, set this to 50 NOTE: To ensure no leakage of signal power, user has to ensure that 100 MHz/LOOPBACK_FREQ is an integer multiple of bin width For e.g. if user chooses 25 Msps sampling rate and 2048 samples/chirp, then LOOPBACK_FREQ of 64 (=1.5625 MHz) will ensure no leakage |
| PA_LOOPBACK_EN | 1 | Value Description 0 PA loopback is not enabled 1 PA loopback is enabled |
| RESERVED | 1 | 0x00 |

NOTE: This API is not supported in IWR6843 ES1.0

5.11.8 Sub block 0x02CE – AWR_RF_PSLOOPBACK_CFG_SB

This sub block enables/disables PS (phase shifter) loopback for all enabled profiles. This is used to debug the TX (before the PA) and RX chains.

NOTE: The PS loop-back configuration API is supported only for debug purpose. Please refer latest DFP release note for more info.

Table 5.66 describes the content of this sub block.

NOTE: If monitoring is enabled with the loopback APIs (subblock 0x02CD, 0x02CE, 0x02CF), then loopback will not work after monitoring is complete. To use loopback with monitoring, use AWR_ADVANCED_FRAME_CONF_SB with AWR_LOOPBACK_BURST_CONF_SB.

Table 5.66: AWR_RF_PSLOOPBACK_CFG_SB contents

| Field Name | Number of bytes | Description |
|------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x02CE |
| SBLKLEN | 2 | Value = 12 |
| PS_LOOPBACK_FREQ | 2 | Loop back frequency in kHz 1 LSB = 1 kHz |
| RESERVED | 2 | 0x0000 |
| PS_LOOPBACK_EN | 1 | Value Definition 0 PS loopback is not enabled 1 PS loopback is enabled |
| PS_LOOPBACK_TXID | 1 | Bit Definition b0 TX0 is used for loopback b1 TX1 is used for loopback b7:2 RESERVED |

Continued on next page

Table 5.66 – continued from previous page

| PGA_GAIN_INDEX | 1 | Value | | PGA gain | |
|----------------|---|-------|------------|----------|-----------|
| | | Value | PGA value | Value | PGA value |
| | | 0 | PGA is OFF | 15 | -3 dB |
| | | 1 | -22 dB | 16 | -2 dB |
| | | 2 | -16 dB | 17 | -1 dB |
| | | 3 | -15 dB | 18 | 0 dB |
| | | 4 | -14 dB | 19 | 1 dB |
| | | 5 | -13 dB | 20 | 2 dB |
| | | 6 | -12 dB | 21 | 3 dB |
| | | 7 | -11 dB | 22 | 4 dB |
| | | 8 | -10 dB | 23 | 5 dB |
| | | 9 | -9 dB | 24 | 6 dB |
| | | 10 | -8 dB | 25 | 7 dB |
| | | 11 | -7 dB | 26 | 8 dB |
| | | 12 | -6 dB | 27 | 9 dB |
| | | 13 | -5 dB | 255-28 | RESERVED |
| | | 14 | -4 dB | | |
| RESERVED | 1 | 0x00 | | | |

NOTE: This API is not supported in IWR6843 ES1.0

5.11.9 Sub block 0x02CF – AWR_RF_IFLOOPBACK_CFG_SB

This sub block enables/disables IF loopback for all enabled profiles. This is used to debug the RX IF chain.

NOTE: The IF loop-back configuration API is supported only for debug purpose. Please refer latest DFP release note for more info.

Table 5.67 describes the content of this sub block.

NOTE: If monitoring is enabled with the loopback APIs (subblock 0x02CD, 0x02CE, 0x02CF), then loopback will not work after monitoring is complete. To use loopback with monitoring, use AWR_ADVANCED_FRAME_CONF_SB with AWR_LOOPBACK_BURST_CONF_SB.

Table 5.67: AWR_RF_IFLOOPBACK_CFG_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----------------------------|---|----------------------------|---|------------------------|---|---------|---|---------|---|-------|---|-------|---|---------|---|-------|---|--------------|---|-------|----|-------|----|--------------|----|-------|----|--------|----------|----------|
| SBLKID | 2 | Value = 0x02CF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IF_LOOPBACK_FREQ | 2 | <table border="0"> <tr> <td>Value</td> <td>IF Loopback frequency value</td> </tr> <tr> <td>0</td> <td>180 kHz</td> </tr> <tr> <td>1</td> <td>240 kHz</td> </tr> <tr> <td>2</td> <td>360 kHz</td> </tr> <tr> <td>3</td> <td>720 kHz</td> </tr> <tr> <td>4</td> <td>1 MHz</td> </tr> <tr> <td>5</td> <td>2 MHz</td> </tr> <tr> <td>6</td> <td>2.5 MHz</td> </tr> <tr> <td>7</td> <td>3 MHz</td> </tr> <tr> <td>8</td> <td>4.017857 MHz</td> </tr> <tr> <td>9</td> <td>5 MHz</td> </tr> <tr> <td>10</td> <td>6 MHz</td> </tr> <tr> <td>11</td> <td>8.035714 MHz</td> </tr> <tr> <td>12</td> <td>9 MHz</td> </tr> <tr> <td>13</td> <td>10 MHz</td> </tr> <tr> <td>65535-14</td> <td>RESERVED</td> </tr> </table> | Value | IF Loopback frequency value | 0 | 180 kHz | 1 | 240 kHz | 2 | 360 kHz | 3 | 720 kHz | 4 | 1 MHz | 5 | 2 MHz | 6 | 2.5 MHz | 7 | 3 MHz | 8 | 4.017857 MHz | 9 | 5 MHz | 10 | 6 MHz | 11 | 8.035714 MHz | 12 | 9 MHz | 13 | 10 MHz | 65535-14 | RESERVED |
| Value | IF Loopback frequency value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 180 kHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 240 kHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 360 kHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 720 kHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 1 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 2 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2.5 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 3 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 4.017857 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 5 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | 6 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | 8.035714 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | 9 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | 10 MHz | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 65535-14 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IF_LOOPBACK_EN | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>IF loopback is not enabled</td> </tr> <tr> <td>1</td> <td>IF loopback is enabled</td> </tr> </table> | Value | Definition | 0 | IF loopback is not enabled | 1 | IF loopback is enabled | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | Definition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | IF loopback is not enabled | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | IF loopback is enabled | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

NOTE: This API is not supported in IWR6843 ES1.0

5.11.10 Sub block 0x02D0 – AWR_RF_GPADC_CFG_SET_SB

This sub block enables the GPADC reads for external inputs (available only in xWR1642/xWR1843/IWR6843). Table 5.68 describes the content of this sub block.

Table 5.68: AWR_RF_GPADC_CFG_SET_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | |
|-----------------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|------|------------------------------------------------------------|-------|----------------------------------------------------------------------|---|-------------|---|-------------|---|--------|--------|----------|--------|----------|
| SBLKID | 2 | Value = 0x02D0 | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 32 | | | | | | | | | | | | | | | | |
| SIGNAL.INPUT_ENABLES | 1 | <p>This field indicates the sets of externally fed DC signals which are to be monitored using GPADC. When each bit in this field is set, the corresponding signal is monitored.</p> <table> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>ANALOGTEST1</td> </tr> <tr> <td>1</td> <td>ANALOGTEST2</td> </tr> <tr> <td>2</td> <td>ANALOGTEST3</td> </tr> <tr> <td>3</td> <td>ANALOGTEST4</td> </tr> <tr> <td>4</td> <td>ANAMUX</td> </tr> <tr> <td>5</td> <td>VSENSE</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> | Bit | Definition | 0 | ANALOGTEST1 | 1 | ANALOGTEST2 | 2 | ANALOGTEST3 | 3 | ANALOGTEST4 | 4 | ANAMUX | 5 | VSENSE | Others | RESERVED |
| Bit | Definition | | | | | | | | | | | | | | | | | |
| 0 | ANALOGTEST1 | | | | | | | | | | | | | | | | | |
| 1 | ANALOGTEST2 | | | | | | | | | | | | | | | | | |
| 2 | ANALOGTEST3 | | | | | | | | | | | | | | | | | |
| 3 | ANALOGTEST4 | | | | | | | | | | | | | | | | | |
| 4 | ANAMUX | | | | | | | | | | | | | | | | | |
| 5 | VSENSE | | | | | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | | | | | |
| SIGNAL_BUFFER_ENABLES | 1 | <p>This field indicates the sets of externally fed DC signals which are to be buffered before being fed to the GPADC. When each bit in this field is set, the corresponding signal is buffered before the GPADC.</p> <table> <tr> <td>Bit</td> <td>SIGNAL</td> </tr> <tr> <td>0</td> <td>ANALOGTEST1</td> </tr> <tr> <td>1</td> <td>ANALOGTEST2</td> </tr> <tr> <td>2</td> <td>ANALOGTEST3</td> </tr> <tr> <td>3</td> <td>ANALOGTEST4</td> </tr> <tr> <td>4</td> <td>ANAMUX</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> | Bit | SIGNAL | 0 | ANALOGTEST1 | 1 | ANALOGTEST2 | 2 | ANALOGTEST3 | 3 | ANALOGTEST4 | 4 | ANAMUX | Others | RESERVED | | |
| Bit | SIGNAL | | | | | | | | | | | | | | | | | |
| 0 | ANALOGTEST1 | | | | | | | | | | | | | | | | | |
| 1 | ANALOGTEST2 | | | | | | | | | | | | | | | | | |
| 2 | ANALOGTEST3 | | | | | | | | | | | | | | | | | |
| 3 | ANALOGTEST4 | | | | | | | | | | | | | | | | | |
| 4 | ANAMUX | | | | | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | | | | | |
| ANATEST1_CFG | 2 | <table> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>b7:0</td> <td>Number of samples to collect 1 sample takes 1.6 μs</td> </tr> <tr> <td>b15:8</td> <td>Settling time 1 LSB = 0.8 μs Valid range: 0 to 12 μs</td> </tr> </table> <p>Valid programming condition: all the signals that are enabled should take a total of < 100 μs including the programmed settling times and measurement time per enabled signal.</p> | Bit | Definition | b7:0 | Number of samples to collect 1 sample takes 1.6 μ s | b15:8 | Settling time 1 LSB = 0.8 μ s Valid range: 0 to 12 μ s | | | | | | | | | | |
| Bit | Definition | | | | | | | | | | | | | | | | | |
| b7:0 | Number of samples to collect 1 sample takes 1.6 μ s | | | | | | | | | | | | | | | | | |
| b15:8 | Settling time 1 LSB = 0.8 μ s Valid range: 0 to 12 μ s | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.68 – continued from previous page

| | | |
|--------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ANATEST2_CFG | 2 | <p>Bit Definition</p> <p>b7:0 Number of samples to collect 1 sample takes 1.6 μs</p> <p>b15:8 Settling time 1 LSB = 0.8 μs Valid range: 0 to 12 μs</p> <p>Valid programming condition: all the signals that are enabled should take a total of < 100 μs including the programmed settling times and measurement time per enabled signal.</p> |
| ANATEST3_CFG | 2 | <p>Bit Definition</p> <p>b7:0 Number of samples to collect 1 sample takes 1.6 μs</p> <p>b15:8 Settling time 1 LSB = 0.8 μs Valid range: 0 to 12 μs</p> <p>Valid programming condition: all the signals that are enabled should take a total of < 100 μs including the programmed settling times and measurement time per enabled signal.</p> |
| ANATEST4_CFG | 2 | <p>Bit Definition</p> <p>b7:0 Number of samples to collect 1 sample takes 1.6 μs</p> <p>b15:8 Settling time 1 LSB = 0.8 μs Valid range: 0 to 12 μs</p> <p>Valid programming condition: all the signals that are enabled should take a total of < 100 μs including the programmed settling times and measurement time per enabled signal.</p> |
| ANAMUX_CFG | 2 | <p>Bit Definition</p> <p>b7:0 Number of samples to collect 1 sample takes 1.6 μs</p> <p>b15:8 Settling time 1 LSB = 0.8 μs Valid range: 0 to 12 μs</p> <p>Valid programming condition: all the signals that are enabled should take a total of < 100 μs including the programmed settling times and measurement time per enabled signal.</p> |

Continued on next page

Table 5.68 – continued from previous page

| VSENSE_CFG | 2 | Bit | Definition |
|------------|---|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | b7:0 | Number of samples to collect 1 sample takes 1.6 μ s |
| | | b15:8 | Settling time 1 LSB = 0.8 μ s Valid range: 0 to 12 μ s Valid programming condition: all the signals that are enabled should take a total of < 100 μ s including the programmed settling times and measurement time per enabled signal. |
| RESERVED | 2 | 0x0000 | |
| RESERVED | 4 | 0x00000000 | |
| RESERVED | 4 | 0x00000000 | |
| RESERVED | 4 | 0x00000000 | |

The response to the AWR_RF_GPADC_CFG.SET_SB is an async event AWR_AE_RF_GPADC_RESULT_DATA_SB which contains the measured values for each of the enabled channels.

NOTE: The actual measurement of these GPADC signal are done in inter-burst or frame idle time and the result AE sub block will be sent only after completing all the measurements.

5.11.11 Sub block 0x02D1 – RESERVED

5.11.12 Sub block 0x02D2 – RESERVED

5.11.13 Sub block 0x02D3 – RESERVED

5.12 Sub blocks related to AWR_RF_MISC_CONF_GET_MSG

5.12.1 Sub block 0x02E0 to 0x2E9 – RESERVED

5.12.2 Sub block 0x02EA – AWR_RF_TEMPERATURE_GET_SB

This sub block provides the device temperature sensor information.
 Table 5.69 describes the content of this sub block.

Table 5.69: AWR_RF_TEMPERATURE_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x02EA |
| SBLKLEN | 2 | Value = 4 |

AWR_RF_TEMPERATURE_DATA_SB sub block is sent by the radar device in response to AWR_RF_TEMPERATURE_GET_SB.

Table 5.70: AWR_RF_TEMPERATURE_DATA_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|---------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x02EA |
| SBLKLEN | 2 | Value = 28 |
| TIME | 4 | BSS local Time from device power up 1 LSB = 1 ms |
| TEMP_RX0_SENS | 2 | RX0 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_RX1_SENS | 2 | RX1 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_RX2_SENS | 2 | RX2 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_RX3_SENS | 2 | RX3 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_TX0_SENS | 2 | TX0 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_TX1_SENS | 2 | TX1 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_TX2_SENS | 2 | TX2 temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_PM_SENS | 2 | PM temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_DIG1_SENS | 2 | Digital temperature sensor reading (signed value) 1 LSB = 1°C |
| TEMP_DIG2_SENS | 2 | Digital temperature sensor reading (signed value) [Applicable only in xWR1642/IWR6843/xWR1843] 1 LSB = 1°C |

NOTE: In IWR6843 ES1.0, only Tx1, Tx2, Tx3 and PM temperature sensors are supported, rest of the temperature sensors always reads zero.

5.13 Sub blocks related to AWR_RF_ASYNC_EVENT_MSG1

NOTE1: All the Monitoring Async events will be sent out periodically at CAL_MON_TIME_UNIT frame rate (FTTI). The RadarSS/BSS has a queue to hold max 8 transmit API messages (AEs or Responses), the host shall service all the AEs before start of the next FTTI epoch to avoid RadarSS Queue full CPU fault fatal error.

NOTE2: In reporting mode 1 (Quiet mode) if any failure in RadarSS analog or digital monitors the AWR1243 device will send AWR_AE_MSS_RFERROR_STATUS_SB AE with error code 0x7 along with failure monitoring report.

5.13.1 Sub block 0x1000 – RESERVED

5.13.2 Sub block 0x1001 – RESERVED

5.13.3 Sub block 0x1002 – AWR_AE_RF_CPUFAULT_SB

This sub block indicates CPU fault status of BIST SS.

Table 5.71 describes the content of this sub block.

Table 5.71: AWR_AE_RF_CPUFAULT_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1002 |
| SBLKLEN | 2 | Value = 36 |
| FAULT_TYPE | 1 | Value Definition 0 RF Processor Undefined Instruction Abort 1 RF Processor Instruction pre-fetch Abort 2 RF Processor Data Access Abort 3 RF Processor Firmware Fatal Error 0x4 - RESERVED 0xFE 0xFF No fault |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 5.71 – continued from previous page

| | | |
|----------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LINE_NUM | 2 | Valid only in case of FAULT type is 0x3, provides the firmware line number at which fatal error occurred. |
| FAULT_LR | 4 | The instruction PC address at which Fault occurred |
| FAULT_PREV_LR | 4 | The return address of the function from which fault function has been called (Call stack LR) |
| FAULT_SPSR | 4 | The CPSR register value at which fault occurred |
| FAULT_SP | 4 | The SP register value at which fault occurred |
| FAULT_CAUSE_ADDRESS | 4 | The address access at which Fault occurred (valid only for fault type 0x0 to 0x2) |
| FAULT_ERROR_STATUS | 2 | The status of Error (Error Cause type – valid only for fault type 0x0 to 0x2) 0x000 BACKGROUND_ERR 0x001 ALIGNMENT_ERR 0x002 DEBUG_EVENT 0x00D PERMISSION_ERR 0x008 SYNCH_EXTER_ERR 0x406 ASYNCH_EXTER_ERR 0x409 SYNCH_ECC_ERR 0x408 ASYNCH_ECC_ERR |
| FAULT_ERROR_SOURCE | 1 | The Source of the Error (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 ERR_SOURCE_AXI_MASTER 0x1 ERR_SOURCE_ATCM 0x2 ERR_SOURCE_BTCM |
| FAULT_AXI_ERROR_TYPE | 1 | The AXI Error type (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 AXI_DECOD_ERR 0x1 AXI_SLAVE_ERR |
| FAULT_ACCESS_TYPE | 1 | The Error Access type (Error Access type - valid only for fault type 0x0 to 0x2) 0x0 READ_ERR 0x1 WRITE_ERR |
| FAULT_RECOVERY_TYPE | 1 | The Error Recovery type (Error Recovery type - Valid only for fault type 0x0 to 0x2) 0x0 UNRECOVERY 0x1 RECOVERY |
| RESERVED | 2 | 0x0000 |

5.13.4 Sub block 0x1003 – AWR_AE_RF_ESMFAULT_SB

This sub block indicates the status of any other faults in the BIST SS.
Table 5.72 describes the content of this sub block.

Table 5.72: AWR_AE_RF_ESMFAULT_STATUS_SB response contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1003 |
| SBLKLEN | 2 | Value = 12 |
| ESM_GROUP1_ERRORS | 4 | Bit Error Information b0 RAMPGEN_SB_ERROR b1 RESERVED b2 GPADC_RAM_SB_ERROR b3 VIM_RAM_SB_ERROR b4 DFE_SELFTEST_ERROR b5 VIM_SELFTEST_ERROR b6 B0TCM_SB_ERROR b7 B1TCM_SB_ERROR b8 CCMR4_SELFTEST_ERROR b9 ATCM_SB_ERROR b10 RAMPGEN_SELFTEST_ERROR b11 RAMPGEN_PAR_SELFTEST_ERROR b12 SEQ_EXT_SELFTEST_ERROR b13 SEQ_EXT_SB_ERROR b14 PROG_FILT_FATAL_PARITY_ERROR b15 AGC_RAM_SB_ERROR b16 B1TCM_PAR_CHK_ERROR b17 B0TCM_PAR_CHK_ERROR b18 ATCM_PAR_CHK_ERROR b19 MB_MSS2BSS_SB_ERROR b20 MB_BSS2MSS_SB_ERROR b24:21 RESERVED b25 PROG_FILT_FATAL_DB_ECC_ERROR b31:26 RESERVED |

Continued on next page

Table 5.72 – continued from previous page

| | | | |
|-----------------------|-----------|-----|------------------------|
| ESM.GROUP2_ ERRORS | 4 | Bit | Error Information |
| | | b0 | DFE.STC.ERROR |
| | | b1 | CR4.STC.ERROR |
| | | b2 | CCMR4.COMP.ERROR |
| | | b3 | B0TCM.DB.ERROR |
| | | b4 | B1TCM.DB.ERROR |
| | | b5 | ATCM.DB.ERROR |
| | | b6 | DCC.ERROR |
| | | b7 | SEQ.EXT.ERROR |
| | | b8 | SYNT.FREQ.MON.ERROR |
| | | b9 | DFE.PARITY.ERROR |
| | | b10 | RAMPGEN.DB.ERROR |
| | | b11 | BUBBLE.CORRECTION.FAIL |
| | | b12 | RAMPGEN.LOCSTEP.ERROR |
| | | b13 | RTI.RESET.ERROR |
| | | b14 | GPADC.RAM.DB.ERROR |
| | | b15 | VIM.COMP.ERROR |
| | | b16 | CR4.LIVE.LOCK.ERROR |
| | | b17 | WDT.NMI.ERROR |
| | | b18 | VIM.RAM.DB.ERROR |
| | | b19 | RAMPGEN.PAR.ERROR |
| | | b20 | SEQ.EXT.DB.ERROR |
| | | b21 | DMA.MPU.ERROR |
| | | b22 | AGC.RAM.DB.ERROR |
| | | b23 | CRC.COMP.ERROR |
| | | b24 | WAKEUP.STS.ERROR |
| | | b25 | SHORT.CIRCUIT.ERROR |
| | | b26 | B1TCM.PAR.ERROR |
| | | b27 | B0TCM.PAR.ERROR |
| | | b28 | ATCM.PAR.ERROR |
| | | b29 | MB.MSS2BSS.DB.ERROR |
| | | b30 | MB.BSS2MSS.DB.ERROR |
| b31 | CCC.ERROR | | |

NOTE: The Programmable filter Parity error and double bit ECC fatal errors are connected to ESM Group 1 lines, these fatal errors must be handled in Host (MSS or DSS in xWR1642, xWR1843 and IWR6843 devices)

5.13.5 Sub block 0x1004 – AWR_AE_RF_INITCALIBSTATUS_SB

This sub block indicates the initial calibrations of RF BIST SS are complete. Table 5.73 describes the content of this sub block.

Table 5.73: AWR_AE_RF_INITCALIBSTATUS_SB response contents

| Field Name | Number of bytes | Description |
|--------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1004 |
| SBLKLEN | 2 | Value = 24 |
| CALIBRATION_STATUS | 4 | This field indicates the status of each calibration (0 – FAIL, 1 – PASS). If a particular calibration was not enabled, then its corresponding field should be ignored. Bit Definition (0 – FAIL, 1 – PASS) b0 RESERVED b1 APLL tuning b2 SYNTH VCO1 tuning b3 SYNTH VCO2 tuning b4 LODIST calibration b5 RX ADC DC offset calibration b6 HPF cutoff calibration b7 LPF cutoff calibration b8 Peak detector calibration b9 TX Power calibration b10 RX gain calibration b11 TX Phase calibration b12 RX IQMM calibration b31:13 RESERVED |

Continued on next page

Table 5.73 – continued from previous page

| | | |
|--------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALIBRATION_UPDATE | 4 | <p>This field indicates if a particular calibration data has been updated in hardware. (0 – no update, 1 – updated)</p> <p>Bit Definition</p> <p>b0 RESERVED</p> <p>b1 APLL tuning</p> <p>b2 SYNTH VCO1 tuning</p> <p>b3 SYNTH VCO2 tuning</p> <p>b4 LODIST calibration</p> <p>b5 RX ADC DC offset calibration</p> <p>b6 HPF cutoff calibration</p> <p>b7 LPF cutoff calibration</p> <p>b8 Peak detector calibration</p> <p>b9 TX Power calibration</p> <p>b10 RX gain calibration</p> <p>b11 TX Phase calibration</p> <p>b12 RX IQMM calibration</p> <p>b31:13 RESERVED</p> |
| TEMPERATURE | 2 | <p>Measured temperature, based on average of temperature sensors near all enabled TX and RX channels at the time of calibration.</p> <p>1 LSB = 1°C</p> |
| RESERVED | 2 | 0x0000 |
| TIME_STAMP | 4 | <p>This field indicates time stamp at the time of performing calibration updates.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> |
| RESERVED | 4 | 0x00000000 |

NOTE: In IWR6843 ES1.0, only APLL, SYNTH1, SYNTH2 and LODIST calibrations are supported.

5.13.6 Sub block 0x1005 – RESERVED
5.13.7 Sub block 0x1006 – RESERVED
5.13.8 Sub block 0x1007 – RESERVED
5.13.9 Sub block 0x1008 – RESERVED
5.13.10 Sub block 0x1009 – RESERVED
5.13.11 Sub block 0x100A – RESERVED
5.13.12 Sub block 0x100B – AWR_AE_RF_FRAME_TRIGGER_RDY_SB

This sub block indicates that the slave device is now ready to receive the external sync in for frame triggers. In SW triggered mode, this async event indicates that frames are triggered. Table 5.74 describes the content of this sub block.

Table 5.74: AWR_AE_RF_FRAME_TRIGGER_RDY_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x100B |
| SBLKLEN | 2 | Value = 4 |

5.13.13 Sub block 0x100C – AWR_AE_RF_GPADC_RESULT_DATA_SB

This sub block indicates that GPADC measurement is complete and it also contains the measured data of each of the enabled channels. The data for channels which are not enabled can be ignored.

Table 5.75 describes the content of this sub block.

Table 5.75: AWR_AE_RF_GPADC_RESULT_DATA_SB response contents

| Field Name | Number of bytes | Description |
|-------------------|-----------------|-------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x100C |
| SBLKLEN | 2 | Value = 76 |
| ANATEST1_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for ANATEST1 input 1 LSB = 1.8V/1024 |
| ANATEST1_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for ANATEST1 input 1 LSB = 1.8V/1024 |

Continued on next page

Table 5.75 – continued from previous page

| | | |
|-------------------|---|-------------------------------------------------------------------------------------------|
| ANATEST1_AVG_DATA | 2 | Average GPADC reading across the captured samples for ANATEST1 input 1 LSB = 1.8V/1024 |
| ANATEST2_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for ANATEST2 input 1 LSB = 1.8V/1024 |
| ANATEST2_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for ANATEST2 input 1 LSB = 1.8V/1024 |
| ANATEST2_AVG_DATA | 2 | Average GPADC reading across the captured samples for ANATEST2 input 1 LSB = 1.8V/1024 |
| ANATEST3_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for ANATEST3 input 1 LSB = 1.8V/1024 |
| ANATEST3_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for ANATEST3 input 1 LSB = 1.8V/1024 |
| ANATEST3_AVG_DATA | 2 | Average GPADC reading across the captured samples for ANATEST3 input 1 LSB = 1.8V/1024 |
| ANATEST4_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for ANATEST4 input 1 LSB = 1.8V/1024 |
| ANATEST4_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for ANATEST4 input 1 LSB = 1.8V/1024 |
| ANATEST4_AVG_DATA | 2 | Average GPADC reading across the captured samples for ANATEST4 input 1 LSB = 1.8V/1024 |
| ANAMUX_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for ANAMUX input 1 LSB = 1.8V/1024 |
| ANAMUX_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for ANAMUX input 1 LSB = 1.8V/1024 |
| ANAMUX_AVG_DATA | 2 | Average GPADC reading across the captured samples for ANAMUX input 1 LSB = 1.8V/1024 |
| VSENSE_MIN_DATA | 2 | Minimum GPADC reading across the captured samples for VSENSE input 1 LSB = 1.8V/1024 |

Continued on next page

Table 5.75 – continued from previous page

| | | |
|-----------------|---|-----------------------------------------------------------------------------------------|
| VSENSE_MAX_DATA | 2 | Maximum GPADC reading across the captured samples for VSENSE input 1 LSB = 1.8V/1024 |
| VSENSE_AVG_DATA | 2 | Average GPADC reading across the captured samples for VSENSE input 1 LSB = 1.8V/1024 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

5.13.14 Sub block 0x100E – RESERVED
5.13.15 Sub block 0x100D – RESERVED
5.13.16 Sub block 0x100E – RESERVED
5.13.17 Sub block 0x100F – AWR_FRAME_END_AE_SB

This sub block indicates end of the frames.

Table 5.76 describes the content of this sub block.

Table 5.76: AWR_FRAME_END_AE_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x100F |
| SBLKLEN | 2 | Value = 4 |

5.13.18 Sub block 0x1010 – AWR_ANALOGFAULT_AE_SB

This sub block indicates fault in analog supplies or LDO short circuit condition. Once a fault is detected the functionality cannot be resumed from then on and the sensor needs to be re-started.

Table 5.77: AWR_ANALOGFAULT_AE_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1010 |
| SBLKLEN | 2 | Value = 16 |
| FAULT_TYPE | 1 | Value Definition 0 NO FAULT 1 ANALOG_SUPPLY_FAULT Others RESERVED |
| RESERVED | 1 | 0x00 |
| RESERVED | 2 | 0x0000 |
| FAULT_SIG | 4 | Bit Definition b0 1.8V BB ANA supply fault detected b1 13V/1.0V RF supply fault detected b2 Synth VCO LDO short circuit detected b3 PA LDO short circuit detected b31:4 RESERVED |
| RESERVED | 4 | 0x00000000 |

5.13.19 Sub block 0x1011 – AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB

This sub block indicates any timing failure related to calibration or monitoring.

Table 5.78 describes the content of this sub block.

Table 5.78: AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1011 |
| SBLKLEN | 2 | Value = 8 |

Continued on next page

Table 5.78 – continued from previous page

| | | | | |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIMING.FAIL- URE.CODE | 2 | Bit | Definition | |
| | | b0 | RESERVED | |
| | | b1 | 0 | No Failure |
| | | | 1 | Total monitoring and calibration time do not fit in one CALIB.MON.TIME.UNIT in AWR.RUN.TIME.CALIBRATION.CONF.AND.TRIGGER when ONE.TIME.CALIB is enabled |
| | | b2 | 0 | No Failure |
| 1 | Total monitoring and calibration time do not fit in one CALIB.MON.TIME.UNIT in AWR.RUN.TIME.CALIBRATION.CONF.AND.TRIGGER when PERIODIC.CALIB is enabled | | | |
| b3 | 0 | No Failure | | |
| | 1 | Runtime timing violation: Monitoring functions or calibrations could not be completed in one CALIB.MON.TIME.UNIT | | |
| | | b15:4 | RESERVED | |
| RESERVED | 2 | 0x0000 | | |

NOTE: In QM devices (non safety), Periodic Digital and Analog Monitoring are not supported.

5.13.20 Sub block 0x1012 – AWR_RUN_TIME_CALIB_SUMMARY_REPORT_AE_SB

This sub block indicates the calibration status (one time or run time) if the calibration reports are enabled in the AWR.RUN.TIME.CALIBRATION.CONF.AND.TRIGGER.SB.

NOTE: The calibration report is sent if the calibrations are triggered due to temperature change or whenever the internal calibrations are triggered i.e. every 1 s

Table 5.79: AWR_RUN_TIME_CALIB_SUMMARY_REPORT_AE_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1012 |
| SBLKLEN | 2 | Value = 24 |

Continued on next page

Table 5.79 – continued from previous page

| | | |
|---------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CALIBRATION_ERROR_FLAG | 4 | <p>This field indicates the status of each calibration. 1 - calibration is passed, 0 - calibration is failed or not enabled/performed at least once.</p> <p>Bit Definition</p> <p>b0 RESERVED</p> <p>b1 APLL tuning</p> <p>b2 SYNTH VCO1 tuning</p> <p>b3 SYNTH VCO2 tuning</p> <p>b4 LODIST calibration</p> <p>b5 RESERVED</p> <p>b6 RESERVED</p> <p>b7 RESERVED</p> <p>b8 PD calibration</p> <p>b9 TX power calibration</p> <p>b10 RX gain calibration</p> <p>b11 RESERVED</p> <p>b12 RESERVED</p> <p>b31:13 RESERVED</p> |
| CALIBRATION_UPDATE_STATUS | 4 | <p>Each bit corresponding to a calibration indicates if each calibration resulted in a reconfiguration of RF is indicated by a value of 1 in the respective bit in this field. 0 – Analog/RF is not updated 1 – Analog/RF is updated after a respective calibration</p> <p>Bit Definition</p> <p>b0 RESERVED</p> <p>b1 APLL tuning</p> <p>b2 SYNTH VCO1 tuning</p> <p>b3 SYNTH VCO2 tuning</p> <p>b4 LODIST calibration</p> <p>b5 RESERVED</p> <p>b6 RESERVED</p> <p>b7 RESERVED</p> <p>b8 PD calibration</p> <p>b9 TX power calibration</p> <p>b10 RX gain calibration</p> <p>b11 RESERVED</p> <p>b12 RESERVED</p> <p>b31:13 RESERVED</p> |

Continued on next page

Table 5.79 – continued from previous page

| | | |
|-------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TEMPERATURE | 2 | Measured temperature, based on average of temperature sensors near all enabled TX and RX channels at the time of calibration. 1 LSB = 1°C |
| RESERVED | 2 | RESERVED |
| TIME_STAMP | 4 | This field indicates time stamp at the time of performing calibration updates. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |
| RESERVED | 4 | 0x00000000 |

NOTE: In IWR6843 ES1.0, only APLL, SYNTH1, SYNTH2 and LODIST runtime calibrations are supported.

NOTE: None of the Periodic Monitoring are supported in QM devices (IWR6843, xWR1443, IWR1642 and IWR1843), The Async Event sub-blocks defined below from ID 0x1015 to 0x1031 are not valid in QM devices.

5.13.21 Sub block 0x1013 – AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB

This async event contains the status of digital monitoring for latent faults.

Table 5.80: AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB response contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1013 |
| SBLKLEN | 2 | Value = 8 |

Continued on next page

Table 5.80 – continued from previous page

| | | |
|-----------------------------|---|------------------------------------------------------------------------------------------|
| DIG_MON_LATENT_FAULT_STATUS | 4 | 1 – PASS, 0 – FAIL |
| | | Bit Definition |
| | | b0 RESERVED |
| | | b1 CR4 and VIM lockstep test |
| | | b2 RESERVED |
| | | b3 VIM test (Not supported in 1st Gen devices, refer latest release note) |
| | | b4 RESERVED |
| | | b5 RESERVED |
| | | b6 CRC test |
| | | b7 RAMPGEN memory ECC test (Not supported in 1st Gen devices, refer latest release note) |
| | | b8 DFE Parity test |
| | | b9 DFE memory ECC test |
| | | b10 RAMPGEN lockstep test |
| | | b11 FRC lockstep test |
| | | b12 RESERVED |
| | | b13 RESERVED |
| | | b14 RESERVED |
| | | b15 RESERVED |
| | | b16 ESM test |
| | | b17 DFE STC |
| | | b18 RESERVED |
| | | b19 ATCM, BTCM ECC test |
| | | b20 ATCM, BTCM parity test |
| | | b21 RESERVED |
| | | b22 RESERVED |
| | | b23 RESERVED |
| | | b24 FFT test |
| | | b25 RTI test |
| b26 RESERVED | | |
| b31:27 RESERVED | | |

5.13.22 Sub block 0x1014 – RESERVED

5.13.23 Sub block 0x1015 – AWR_MONITOR_REPORT_HEADER_AE_SB

The report header includes common information across all enabled monitors like current FTTI number and current temperature.

Table 5.81: AWR_MONITORING_REPORT_HEADER_AE_SB response contents

| Field Name | Number of bytes | Description |
|-----------------|-----------------|----------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1015 |
| SBLKLEN | 2 | Value = 12 |
| FTTI_COUNT | 4 | FTTI free running counter value, incremented every CAL_MON_TIME_UNIT |
| AVG_TEMPERATURE | 2 | Average temperature at which was monitoring performed |
| RESERVED | 2 | 0x0000 |

5.13.24 Sub block 0x1016 – AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB

This async event is sent periodically to indicate the status of periodic digital monitoring tests.

Table 5.82: AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|----------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1016 |
| SBLKLEN | 2 | Value = 12 |
| RF_DIG_MON_PERIODIC_STATUS | 4 | 1 – PASS, 0 – FAIL Bit Monitoring type b0 PERIODIC_CONFIG_REGISTER_READ b1 RESERVED b2 DFE.STC b3 FRAME_TIMING_MONITORING b31:4 RESERVED |
| TIMESTAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.13.25 Sub block 0x1017 – AWR_MONITOR_TEMPERATURE_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured temperature near various RF analog and digital modules. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.83: AWR_MONITORING_TEMPERATURE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1017 |
| SBLKLEN | 2 | Value = 36 |
| STATUS_FLAGS | 2 | Status flags indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_ANA_TEMP_MIN b1 STATUS_ANA_TEMP_MAX b2 STATUS_DIG_TEMP_MIN b3 STATUS_DIG_TEMP_MAX b4 STATUS_TEMP_DIFF_THRESH b15:5 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |

Continued on next page

Table 5.83 – continued from previous page

| | | |
|-------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TEMP_VALUES | 20 | <p>The measured onchip temperature is reported here. Byte numbers corresponding to different temperature sensors reported in this field are here:</p> <p>Bytes Temperature sensor</p> <p>1:0 TEMP_RX0</p> <p>3:2 TEMP_RX1</p> <p>5:4 TEMP_RX2</p> <p>7:6 TEMP_RX3</p> <p>9:8 TEMP_TX0</p> <p>11:10 TEMP_TX1</p> <p>13:12 TEMP_TX2</p> <p>15:14 TEMP_PM</p> <p>17:16 TEMP_DIG1</p> <p>19:18 TEMP_DIG2 (Applicable only in xWR1642/IWR6843/xWR1843)</p> <p>1 LSB = 1°C, signed number</p> |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> |

5.13.26 Sub block 0x1018 – AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB

This sub block is a monitoring report which the AWR device sends to the host, containing the measured RX Gain and Phase values, Loopback Power and Noise Power. Noise Power can be used by the Host to detect the presence of interference. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.84: AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1018 |
| SBLKLEN | 2 | Value = 72 |

Continued on next page

Table 5.84 – continued from previous page

| | | | | | | | | | | | |
|----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|----------|----------|----------|-------|-------|-------|
| STATUS_FLAGS | 2 | <p>Status flag indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_RX_GAIN_ABS</p> <p>b1 STATUS_RX_GAIN_MISMATCH</p> <p>b2 STATUS_RX_GAIN_FLATNESS</p> <p>b3 STATUS_RX_PHASE_MISMATCH</p> <p>b15:4 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> | | | | | | | | | |
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> | | | | | | | | | |
| PROFILE_INDX | 1 | <p>Profile Index for which this monitoring report applies.</p> | | | | | | | | | |
| LOOPBACK_POWER | 3 | <p>The measured average loop-back power across RX channels at each enabled RF frequency (i.e., lowest, center and highest with 60MHz dither in the profile's RF band) at LNA input is reported here.</p> <p>Byte numbers corresponding to different RF, in this field are here:</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">RF1</td> <td style="text-align: center;">RF2</td> <td style="text-align: center;">RF3</td> </tr> <tr> <td style="text-align: center;">(Byte 0)</td> <td style="text-align: center;">(Byte 1)</td> <td style="text-align: center;">(Byte 2)</td> </tr> <tr> <td style="text-align: center;">b4:b0</td> <td style="text-align: center;">b4:b0</td> <td style="text-align: center;">b4:b0</td> </tr> </table> <p>b7-b5 : RESERVED in each bytes</p> <p>1 LSB = -2 dBm</p> <p>Valid Range = -62dBm to 0dBm</p> <p>Only the entries of enabled RF Frequencies are valid.</p> <p>The LB power might be bad/corrupted under interference conditions.</p> | RF1 | RF2 | RF3 | (Byte 0) | (Byte 1) | (Byte 2) | b4:b0 | b4:b0 | b4:b0 |
| RF1 | RF2 | RF3 | | | | | | | | | |
| (Byte 0) | (Byte 1) | (Byte 2) | | | | | | | | | |
| b4:b0 | b4:b0 | b4:b0 | | | | | | | | | |

Continued on next page

Table 5.84 – continued from previous page

| RX_GAIN_VALUE | 24 | <p>The measured RX gain for each enabled channel, at each enabled RF frequency (i.e., lowest, center and highest in the profile's RF band) is reported here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="0" data-bbox="662 526 1029 716"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>1:0</td> <td>9:8</td> <td>17:16</td> </tr> <tr> <td>RX1</td> <td>3:2</td> <td>11:10</td> <td>19:18</td> </tr> <tr> <td>RX2</td> <td>5:4</td> <td>13:12</td> <td>21:20</td> </tr> <tr> <td>RX3</td> <td>7:6</td> <td>15:14</td> <td>23:22</td> </tr> </tbody> </table> <p>1 LSB = 0.1 dB</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> <p>NOTE: The RX_GAIN_VALUE is measured using only ADC power and under assumption of fixed LB power of -38dBm for all RF1, RF2 and RF3 frequencies to avoid interference impact. Therefore, RX_GAIN_VALUE shows around 10 dB deviation from programmed RX gain across full temperature range.</p> <p>The actual Rx gain of the device in dB = RX GAIN VALUE(Rf freq, Rx) + (-38dBm) – LOOPBACK_POWER_TEMP_dBm(Temp[C])</p> <p>Where LOOPBACK_POWER_TEMP_dBm (Temp[C]) = -0.07*Temp[C] – 34</p> <p>Loopback power is -38 dBm around 50 C, and decreases with slope of -7 dBm/100C. These formulae can be used for monitoring reports not affected by interference.</p> | | RF1 | RF2 | RF3 | RX0 | 1:0 | 9:8 | 17:16 | RX1 | 3:2 | 11:10 | 19:18 | RX2 | 5:4 | 13:12 | 21:20 | RX3 | 7:6 | 15:14 | 23:22 |
|---------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-------|-------|-----|-----|-------|-------|-----|-----|-------|-------|
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | 1:0 | 9:8 | 17:16 | | | | | | | | | | | | | | | | | | | |
| RX1 | 3:2 | 11:10 | 19:18 | | | | | | | | | | | | | | | | | | | |
| RX2 | 5:4 | 13:12 | 21:20 | | | | | | | | | | | | | | | | | | | |
| RX3 | 7:6 | 15:14 | 23:22 | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.84 – continued from previous page

| RX_PHASE_VALUE | 24 | <p>The measured RX phase for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="1" data-bbox="662 488 1029 683"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>1:0</td> <td>9:8</td> <td>17:16</td> </tr> <tr> <td>RX1</td> <td>3:2</td> <td>11:10</td> <td>19:18</td> </tr> <tr> <td>RX2</td> <td>5:4</td> <td>13:12</td> <td>21:20</td> </tr> <tr> <td>RX3</td> <td>7:6</td> <td>15:14</td> <td>23:22</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ/2^{16}$</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> <p>NOTE: These phases include an unknown bias common to all RX channels.</p> | | RF1 | RF2 | RF3 | RX0 | 1:0 | 9:8 | 17:16 | RX1 | 3:2 | 11:10 | 19:18 | RX2 | 5:4 | 13:12 | 21:20 | RX3 | 7:6 | 15:14 | 23:22 |
|-----------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-------|---------|-------|-----|-------|---------|-------|-----|---------|-------|-------|-----|---------|-------|-------|
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | 1:0 | 9:8 | 17:16 | | | | | | | | | | | | | | | | | | | |
| RX1 | 3:2 | 11:10 | 19:18 | | | | | | | | | | | | | | | | | | | |
| RX2 | 5:4 | 13:12 | 21:20 | | | | | | | | | | | | | | | | | | | |
| RX3 | 7:6 | 15:14 | 23:22 | | | | | | | | | | | | | | | | | | | |
| RX_NOISE_POWER1 | 4 | <p>The measured RX noise for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Bit fields corresponding to different RX in RF1 and RF2 (partial in this word) are defined in this field:</p> <table border="1" data-bbox="662 1025 1061 1220"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>b4:b0</td> <td>b24:b20</td> <td>-</td> </tr> <tr> <td>RX1</td> <td>b9:b5</td> <td>b29:b25</td> <td>-</td> </tr> <tr> <td>RX2</td> <td>b14:b10</td> <td>-</td> <td>-</td> </tr> <tr> <td>RX3</td> <td>b19:b15</td> <td>-</td> <td>-</td> </tr> </tbody> </table> <p>b31-b30 : RESERVED</p> <p>1 LSB = -2dBm</p> <p>Valid Range: -62dbm to 0dBm</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> <p>NOTE: Noise Power shows a decreasing trend with temperature with average noise power being -43 dBm at 50 C and decreasing with slope = -8 dBm/100C .</p> <p>Noise_Power_dBm (Temp[C]) = -0.08 * Temp [C] -39</p> <p>Noise Power can be used for detecting the monitoring reports impacted by interference.</p> | | RF1 | RF2 | RF3 | RX0 | b4:b0 | b24:b20 | - | RX1 | b9:b5 | b29:b25 | - | RX2 | b14:b10 | - | - | RX3 | b19:b15 | - | - |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | b4:b0 | b24:b20 | - | | | | | | | | | | | | | | | | | | | |
| RX1 | b9:b5 | b29:b25 | - | | | | | | | | | | | | | | | | | | | |
| RX2 | b14:b10 | - | - | | | | | | | | | | | | | | | | | | | |
| RX3 | b19:b15 | - | - | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.84 – continued from previous page

| RX_NOISE_POWER2 | 4 | <p>The measured RX noise for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Bit fields corresponding to different RX in RF2 (partial in this word) and RF3 are defined in this field:</p> <table border="1" data-bbox="662 492 1085 694"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>-</td> <td>-</td> <td>b14:b10</td> </tr> <tr> <td>RX1</td> <td>-</td> <td>-</td> <td>b19:b15</td> </tr> <tr> <td>RX2</td> <td>-</td> <td>b4:b0</td> <td>b24:b20</td> </tr> <tr> <td>RX3</td> <td>-</td> <td>b9:b5</td> <td>b29:b25</td> </tr> </tbody> </table> <p>b31-b30 : RESERVED 1 LSB = -2dBm Valid Range: -62dbm to 0dBm Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> <p>NOTE: Noise Power shows a decreasing trend with temperature with average noise power being -43 dBm at 50 C and decreasing with slope = -8 dBm/100C .</p> <p>Noise_Power_dBm (Temp[C]) = -0.08 * Temp [C] -39</p> <p>Noise Power can be used for detecting the monitoring reports impacted by interference.</p> | | RF1 | RF2 | RF3 | RX0 | - | - | b14:b10 | RX1 | - | - | b19:b15 | RX2 | - | b4:b0 | b24:b20 | RX3 | - | b9:b5 | b29:b25 |
|-----------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-----|-----|-----|-----|---|---|---------|-----|---|---|---------|-----|---|-------|---------|-----|---|-------|---------|
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | - | - | b14:b10 | | | | | | | | | | | | | | | | | | | |
| RX1 | - | - | b19:b15 | | | | | | | | | | | | | | | | | | | |
| RX2 | - | b4:b0 | b24:b20 | | | | | | | | | | | | | | | | | | | |
| RX3 | - | b9:b5 | b29:b25 | | | | | | | | | | | | | | | | | | | |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> | | | | | | | | | | | | | | | | | | | | |

5.13.27 Sub block 0x1019 – AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured RX noise figure values corresponding to the full IF band of a profile. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.85: AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1019 |
| SBLKLEN | 2 | Value = 52 |

Continued on next page

Table 5.85 – continued from previous page

| STATUS_FLAGS | 2 | <p>Status flag indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_RX_NOISE_FIGURE</p> <p>b15:1 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> | | | | | | | | | | | | | | | | | | | | |
|----------------------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-------|-------|-----|-----|-------|-------|-----|-----|-------|-------|
| ERROR.CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> | | | | | | | | | | | | | | | | | | | | |
| PROFILE.INDX | 1 | <p>Profile Index for which this monitoring report applies.</p> | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 3 | <p>0x000000</p> | | | | | | | | | | | | | | | | | | | | |
| RX_NOISE_FIG- URE.VALUE | 24 | <p>The measured RX input referred for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="0"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>1:0</td> <td>9:8</td> <td>17:16</td> </tr> <tr> <td>RX1</td> <td>3:2</td> <td>11:10</td> <td>19:18</td> </tr> <tr> <td>RX2</td> <td>5:4</td> <td>13:12</td> <td>21:20</td> </tr> <tr> <td>RX3</td> <td>7:6</td> <td>15:14</td> <td>23:22</td> </tr> </tbody> </table> <p>1 LSB = 0.1 dB, signed number</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> | | RF1 | RF2 | RF3 | RX0 | 1:0 | 9:8 | 17:16 | RX1 | 3:2 | 11:10 | 19:18 | RX2 | 5:4 | 13:12 | 21:20 | RX3 | 7:6 | 15:14 | 23:22 |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | 1:0 | 9:8 | 17:16 | | | | | | | | | | | | | | | | | | | |
| RX1 | 3:2 | 11:10 | 19:18 | | | | | | | | | | | | | | | | | | | |
| RX2 | 5:4 | 13:12 | 21:20 | | | | | | | | | | | | | | | | | | | |
| RX3 | 7:6 | 15:14 | 23:22 | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | <p>0x00000000</p> | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | <p>0x00000000</p> | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | <p>0x00000000</p> | | | | | | | | | | | | | | | | | | | | |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> | | | | | | | | | | | | | | | | | | | | |

NOTE: The noise monitor reports the real baseband receivers' noise figure with LNA disabled (to suppress external interference's influence). In complex receiver modes (i.e., complex 1x, complex 2x and pseudo real), the system noise figure is 3dB lower (better) than the reported number.

5.13.28 Sub block 0x101A – AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured RX IF filter attenuation values at the given IF frequencies. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.86: AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x101A |
| SBLKLEN | 2 | Value = 48 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_RX_HPF_ERROR b1 STATUS_RX_LPF_ERROR b2 STATUS_RX_IFA_GAIN_ERROR b15:3 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE.INDX | 1 | Profile Index for which this monitoring report applies. |
| RESERVED | 3 | 0x000000 |

Continued on next page

Table 5.86 – continued from previous page

| HPF_CUTOFF_FREQ_ERROR_VALUE | 8 | <p>The deviations of RX IFA HPF cutoff frequency from the ideally expected values for all the enabled RX channels are reported here.</p> <p>HPF_CUTOFF_FREQ_ERROR = $100 \times (\text{Measured Cutoff Frequency} / \text{Expected Cutoff Frequency}) - 100$, for RX IF filter in the HPF region.</p> <p>Byte numbers corresponding to measured cutoff frequency error on different RX channels, in this field are here:</p> <table border="0"> <thead> <tr> <th></th> <th>I channel</th> <th>Q channel</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>0</td> <td>4</td> </tr> <tr> <td>RX1</td> <td>1</td> <td>5</td> </tr> <tr> <td>RX2</td> <td>2</td> <td>6</td> </tr> <tr> <td>RX3</td> <td>3</td> <td>7</td> </tr> </tbody> </table> <p>1 LSB = 1%, signed number Applicable only for the enabled channels.</p> | | I channel | Q channel | RX0 | 0 | 4 | RX1 | 1 | 5 | RX2 | 2 | 6 | RX3 | 3 | 7 |
|-----------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------|-----------|-----|---|---|-----|---|---|-----|---|---|-----|---|---|
| | I channel | Q channel | | | | | | | | | | | | | | | |
| RX0 | 0 | 4 | | | | | | | | | | | | | | | |
| RX1 | 1 | 5 | | | | | | | | | | | | | | | |
| RX2 | 2 | 6 | | | | | | | | | | | | | | | |
| RX3 | 3 | 7 | | | | | | | | | | | | | | | |
| LPF_CUTOFF_FREQ_ERROR_VALUE | 8 | <p>The deviations of RX IFA LPF cutoff frequency from the ideally expected values for all the enabled RX channels are reported here.</p> <p>LPF_CUTOFF_FREQ_ERROR = $100 \times (\text{Measured Cutoff Frequency} / \text{Expected Cutoff Frequency}) - 100$, for RX IF filter in the LPF region.</p> <p>Byte numbers corresponding to measured cutoff frequency error on different RX channels, in this field are here:</p> <table border="0"> <thead> <tr> <th></th> <th>I channel</th> <th>Q channel</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>0</td> <td>4</td> </tr> <tr> <td>RX1</td> <td>1</td> <td>5</td> </tr> <tr> <td>RX2</td> <td>2</td> <td>6</td> </tr> <tr> <td>RX3</td> <td>3</td> <td>7</td> </tr> </tbody> </table> <p>1 LSB = 1%, signed number Applicable only for the enabled channels.</p> | | I channel | Q channel | RX0 | 0 | 4 | RX1 | 1 | 5 | RX2 | 2 | 6 | RX3 | 3 | 7 |
| | I channel | Q channel | | | | | | | | | | | | | | | |
| RX0 | 0 | 4 | | | | | | | | | | | | | | | |
| RX1 | 1 | 5 | | | | | | | | | | | | | | | |
| RX2 | 2 | 6 | | | | | | | | | | | | | | | |
| RX3 | 3 | 7 | | | | | | | | | | | | | | | |

Continued on next page

Table 5.86 – continued from previous page

| RX_IFA_GAIN_ERROR_VALUE | 8 | <p>The deviations of RX IFA Gain from the ideally expected values for all the enabled RX channels are reported here.</p> <p>Byte numbers corresponding to measured cutoff frequency error on different RX channels and HPF/LPF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>I channel</th> <th>Q channel</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>0</td> <td>4</td> </tr> <tr> <td>RX1</td> <td>1</td> <td>5</td> </tr> <tr> <td>RX2</td> <td>2</td> <td>6</td> </tr> <tr> <td>RX3</td> <td>3</td> <td>7</td> </tr> </tbody> </table> <p>1 LSB = 0.1 dB, signed number Applicable only for the enabled channels.</p> | | I channel | Q channel | RX0 | 0 | 4 | RX1 | 1 | 5 | RX2 | 2 | 6 | RX3 | 3 | 7 |
|-------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------|-----------|-----|---|---|-----|---|---|-----|---|---|-----|---|---|
| | I channel | Q channel | | | | | | | | | | | | | | | |
| RX0 | 0 | 4 | | | | | | | | | | | | | | | |
| RX1 | 1 | 5 | | | | | | | | | | | | | | | |
| RX2 | 2 | 6 | | | | | | | | | | | | | | | |
| RX3 | 3 | 7 | | | | | | | | | | | | | | | |
| IFA_GAIN_EXP | 1 | Expected IFA gain 1 LSB = 1 dB | | | | | | | | | | | | | | | |
| RESERVED | 3 | 0x000000 | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> | | | | | | | | | | | | | | | |

5.13.29 Sub block 0x101B – AWR_MONITOR_TX0_POWER_REPORT_AE_SB

NOTE: The TX[0:2] power monitoring accuracy degrades at high TX back-offs and is unreliable for backoffs higher than 20dB.

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX power values during an explicit monitoring chirp. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.87: AWR_MONITOR_TX0_POWER_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x101B |
| SBLKLEN | 2 | Value = 24 |

Continued on next page

Table 5.87 – continued from previous page

| | | | | | | | | | | |
|----------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| STATUS_FLAGS | 2 | <p>Status flag indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_ABS_ERR</p> <p>b1 STATUS_FLATNESS_ERR</p> <p>b15:2 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> | | | | | | | | |
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> | | | | | | | | |
| PROFILE_INDX | 1 | <p>Profile Index for which this monitoring report applies</p> | | | | | | | | |
| RESERVED | 3 | <p>0x000000</p> | | | | | | | | |
| TX_POWER_VALUE | 6 | <p>The measured TX power for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Byte numbers corresponding to different TX and RF, in this field are here:</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td>RF1</td> <td>RF2</td> <td>RF3</td> </tr> <tr> <td>TX0</td> <td>1:0</td> <td>3:2</td> <td>5:4</td> </tr> </table> <p>(other bytes are reserved)</p> <p>1 LSB = 0.1 dBm, signed number</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are valid.</p> | | RF1 | RF2 | RF3 | TX0 | 1:0 | 3:2 | 5:4 |
| | RF1 | RF2 | RF3 | | | | | | | |
| TX0 | 1:0 | 3:2 | 5:4 | | | | | | | |
| RESERVED | 2 | <p>0x0000</p> | | | | | | | | |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> | | | | | | | | |

5.13.30 Sub block 0x101C – AWR_MONITOR_TX1_POWER_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX power values during an explicit monitoring chirp. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.88: AWR_MONITOR_TX1_POWER_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x101C |
| SBLKLEN | 2 | Value = 24 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_ABS_ERR b1 STATUS_FLATNESS_ERR b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TX_POWER_VALUE | 6 | The measured TX power for each enabled channel, at each enabled RF frequency is reported here. Byte numbers corresponding to different TX and RF, in this field are here: RF1 RF2 RF3 TX1 1:0 3:2 5:4 (other bytes are reserved) 1 LSB = 0.1 dBm, signed number Only the entries of enabled RF Frequencies and enabled RX channels are valid. |
| RESERVED | 2 | 0x0000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.13.31 Sub block 0x101D – AWR_MONITOR_TX2_POWER_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX power values during an explicit monitoring chirp. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.89: AWR_MONITOR_TX2_POWER_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x101D |
| SBLKLEN | 2 | Value = 24 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_ABS_ERR b1 STATUS_FLATNESS_ERR b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TX_POWER_VALUE | 6 | The measured TX power for each enabled channel, at each enabled RF frequency is reported here. Byte numbers corresponding to different TX and RF, in this field are here: RF1 RF2 RF3 TX2 1:0 3:2 5:4 (other bytes are reserved) 1 LSB = 0.1 dBm, signed number Only the entries of enabled RF Frequencies and enabled RX channels are valid. |
| RESERVED | 2 | 0x0000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.13.32 Sub block 0x101E – AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX reflection coefficient's magnitude values, meant for detecting TX ball break. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.90: AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|----------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x101E |
| SBLKLEN | 2 | Value = 20 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_TX0_BALLBREAK b15:1 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR.CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| TX_REFL_CO-EFF_VALUE | 2 | The TX reflection coefficient's magnitude for this channel is reported here. 1 LSB = 0.1 dB, signed number |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.13.33 Sub block 0x101F – AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX reflection coefficient's magnitude values, meant for detecting TX ball break. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.91: AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x101F |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.91 – continued from previous page

| | | |
|----------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_TX0.BALLBREAK b15:1 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR.CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| TX_REFL_CO-EFF_VALUE | 2 | The TX reflection coefficient's magnitude for this channel is reported here. 1 LSB = 0.1 dB, signed number |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14 Sub blocks related to AWR_RF_ASYNC_EVENT_MSG2

5.14.1 Sub block 0x1020 – AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX reflection coefficient's magnitude values, meant for detecting TX ball break. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.92: AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1020 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.92 – continued from previous page

| | | |
|----------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_TX0.BALLBREAK b15:1 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR.CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| TX_REFL_CO-EFF_VALUE | 2 | The TX reflection coefficient's magnitude for this channel is reported here. 1 LSB = 0.1 dB, signed number |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.2 Sub block 0x1021 – AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX gain and phase mismatch values during an explicit monitoring chirp. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.93: AWR_MONITOR_TX_GAIN_PHASE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1021 |
| SBLKLEN | 2 | Value = 60 |

Continued on next page

Table 5.93 – continued from previous page

| STATUS_FLAGS | 2 | <p>Status flag indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_TX_GAIN_MISMATCH</p> <p>b1 STATUS_TX_PHASE_MISMATCH</p> <p>b15:2 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> | | | | | | | | | | | | | | | | |
|----------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|-------|-----|-----|-------|-------|
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> | | | | | | | | | | | | | | | | |
| PROFILE_INDX | 1 | <p>Profile Index for which this monitoring report applies</p> | | | | | | | | | | | | | | | | |
| RESERVED | 3 | <p>0x000000</p> | | | | | | | | | | | | | | | | |
| TX_GAIN_VALUE | 18 | <p>The measured TX PA loopback tone power at the RX ADC input, for each enabled TX channel, at each enabled RF frequency is reported here.</p> <p>Byte numbers corresponding to different TX and RF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>TX0</td> <td>1:0</td> <td>7:6</td> <td>13:12</td> </tr> <tr> <td>TX1</td> <td>3:2</td> <td>9:8</td> <td>15:14</td> </tr> <tr> <td>TX2</td> <td>5:4</td> <td>11:10</td> <td>17:16</td> </tr> </tbody> </table> <p>1 LSB = 0.1dBm, signed number</p> <p>Only the entries of enabled RF Frequencies and enabled TX channels are valid.</p> | | RF1 | RF2 | RF3 | TX0 | 1:0 | 7:6 | 13:12 | TX1 | 3:2 | 9:8 | 15:14 | TX2 | 5:4 | 11:10 | 17:16 |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | |
| TX0 | 1:0 | 7:6 | 13:12 | | | | | | | | | | | | | | | |
| TX1 | 3:2 | 9:8 | 15:14 | | | | | | | | | | | | | | | |
| TX2 | 5:4 | 11:10 | 17:16 | | | | | | | | | | | | | | | |
| TX_PHASE_VALUE | 18 | <p>The measured TX phase for each enabled channel, at each enabled RF frequency is reported here.</p> <p>Byte numbers corresponding to different TX and RF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>TX0</td> <td>1:0</td> <td>7:6</td> <td>13:12</td> </tr> <tr> <td>TX1</td> <td>3:2</td> <td>9:8</td> <td>15:14</td> </tr> <tr> <td>TX2</td> <td>5:4</td> <td>11:10</td> <td>17:16</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ / 2^{16}$</p> <p>Only the entries of enabled RF Frequencies and enabled TX channels are valid.</p> <p>NOTE: these phases include an unknown bias common to all TX channels.</p> | | RF1 | RF2 | RF3 | TX0 | 1:0 | 7:6 | 13:12 | TX1 | 3:2 | 9:8 | 15:14 | TX2 | 5:4 | 11:10 | 17:16 |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | |
| TX0 | 1:0 | 7:6 | 13:12 | | | | | | | | | | | | | | | |
| TX1 | 3:2 | 9:8 | 15:14 | | | | | | | | | | | | | | | |
| TX2 | 5:4 | 11:10 | 17:16 | | | | | | | | | | | | | | | |
| RESERVED | 4 | <p>0x00000000</p> | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.93 – continued from previous page

| | | |
|-----------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RESERVED | 4 | 0x00000000 |
| TIMESTAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.3 Sub block 0x1022 – AWR_MONITOR_TX0_BPM_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX0 BPM error values. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.94: AWR_MONITOR_TX0_BPM_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|-------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1022 |
| SBLKLEN | 2 | Value = 20 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_TX0.BPM.PHASE b1 STATUS_TX0.BPM.AMPLITUDE b2 STATUS_TX0.PHASE.SHIFTER b15:3 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| PH.SHIFTER_MON_VAL2_MSB | 1 | MSB of the monitored phase shift for PH.SHIFTER.MON2 for TX0 When combined with PH.SHIFTER.MON_VAL2.LSB, this represents the phase shift value in 16 bits. $PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$ |
| PH.SHIFTER_MON_VAL1 | 2 | Monitored phase shift for PH.SHIFTER.MON1 for TX0 1 LSB = $360^\circ / 2^{16}$ |

Continued on next page

Table 5.94 – continued from previous page

| | | |
|-----------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX_BPM_PHASE_DIFF_VALUE | 2 | The TX output phase difference between the two BPM settings (phase for TX BPM setting 0 – phase for TX BPM setting 1) is reported here. 1 LSB = $360^\circ / 2^{16}$ |
| TX_BPM_AMPLITUDE_DIFF_VALUE | 1 | The deviation of the TX output amplitude difference between the two BPM settings (amplitude for TX BPM setting 0 – amplitude for TX BPM setting 1) from the ideal 0dB is reported here. 1 LSB = 0.1 dB, signed number |
| PH_SHIFTER_MON_VAL2_LSB | 1 | LSB of the monitored phase shift for PH_SHIFTER_MON2 for TX0 When combined with PH_SHIFTER_MON_VAL2_MSB, this represents the phase shift value in 16 bits. $PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$ 1 LSB = $360^\circ / 2^{16}$ |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.4 Sub block 0x1023 – AWR_MONITOR_TX1_BPM_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX1 BPM error values. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.95: AWR_MONITOR_TX1_BPM_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1023 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.95 – continued from previous page

| | | |
|-----------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | <p>Status flag indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_TX1_BPM_PHASE</p> <p>b1 STATUS_TX1_BPM_AMPLITUDE</p> <p>b2 STATUS_TX1_PHASE_SHIFTER</p> <p>b15:3 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> |
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> |
| PROFILE_INDX | 1 | <p>Profile Index for which this monitoring report applies</p> |
| PH_SHIFTER_MON_VAL2_MSB | 1 | <p>MSB of the monitored phase shift for PH_SHIFTER_MON2 for TX1</p> <p>When combined with PH_SHIFTER_MON_VAL2_LSB, this represents the phase shift value in 16 bits.</p> <p>$PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$</p> |
| PH_SHIFTER_MON_VAL1 | 2 | <p>Monitored phase shift for PH.SHIFTER.MON1 for TX1</p> <p>1 LSB = $360^\circ / 2^{16}$</p> |
| TX_BPM_PHASE_DIFF_VALUE | 2 | <p>The TX output phase difference between the two BPM settings (phase for TX BPM setting 0 – phase for TX BPM setting 1) is reported here.</p> <p>1 LSB = $360^\circ / 2^{16}$</p> |
| TX_BPM_AMPLITUDE_DIFF_VALUE | 1 | <p>The deviation of the TX output amplitude difference between the two BPM settings (amplitude for TX BPM setting 0 – amplitude for TX BPM setting 1) from the ideal 0dB is reported here.</p> <p>1 LSB = 0.1 dB, signed number</p> |
| PH_SHIFTER_MON_VAL2_LSB | 1 | <p>LSB of the monitored phase shift for PH_SHIFTER_MON2 for TX0</p> <p>When combined with PH_SHIFTER_MON_VAL2_MSB, this represents the phase shift value in 16 bits.</p> <p>$PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$</p> <p>1 LSB = $360^\circ / 2^{16}$</p> |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> |

5.14.5 Sub block 0x1024 – AWR_MONITOR_TX2_BPM_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured TX2 BPM error values. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.96: AWR_MONITOR_TX2_BPM_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|-----------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1024 |
| SBLKLEN | 2 | Value = 20 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_TX2.BPM.PHASE b1 STATUS_TX2.BPM.AMPLITUDE b2 STATUS_TX2.PHASE.SHIFTER b15:3 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE.INDX | 1 | Profile Index for which this monitoring report applies |
| PH.SHIFTER_MON_VAL2_MSB | 1 | MSB of the monitored phase shift for PH.SHIFTER_MON2 for TX2 When combined with PH.SHIFTER_MON_VAL2_LSB, this represents the phase shift value in 16 bits. $PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$ |
| PH.SHIFTER_MON_VAL1 | 2 | Monitored phase shift for PH.SHIFTER_MON1 for TX2 1 LSB = $360^\circ / 2^{16}$ |
| TX.BPM.PHASE_DIFF.VALUE | 2 | The TX output phase difference between the two BPM settings (phase for TX BPM setting 0 – phase for TX BPM setting 1) is reported here. 1 LSB = $360^\circ / 2^{16}$ |
| TX.BPM.AMPLITUDE_DIFF.VALUE | 1 | The deviation of the TX output amplitude difference between the two BPM settings (amplitude for TX BPM setting 0 – amplitude for TX BPM setting 1) from the ideal 0dB is reported here. 1 LSB = 0.1 dB, signed number |

Continued on next page

Table 5.96 – continued from previous page

| | | |
|-------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PH_SHIFTER_MON_VAL2_LSB | 1 | LSB of the monitored phase shift for PH_SHIFTER_MON2 for TX0 When combined with PH_SHIFTER_MON_VAL2_MSB, this represents the phase shift value in 16 bits. $PH_SHIFTER_MON_VAL2 = PH_SHIFTER_MON_VAL2_MSB \times 2^8 + PH_SHIFTER_MON_VAL2_LSB$ $1\text{ LSB} = 360^\circ / 2^{16}$ |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. $1\text{ LSB} = 1\text{ millisecond}$ (time stamp rolls over upon exceeding allotted bit width) |

5.14.6 Sub block 0x1025 – AWR_MONITOR_SYNTHESIZER_FREQUENCY_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information related to measured frequency error during the chirp. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.97: AWR_MONITOR_SYNTH_FREQUENCY_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1025 |
| SBLKLEN | 2 | Value = 32 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SYNTH_FREQ_ERR b15:1 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |

Continued on next page

Table 5.97 – continued from previous page

| | | |
|-------------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAX_FRE- QUENCY_ER- ROR_VALUE | 4 | This field indicates the maximum instantaneous frequency error measured during the chirps for which frequency monitoring has been enabled in the previous monitoring period. Bits Parameter b31:0 Maximum frequency error value, signed number. 1 LSB = 1 kHz. |
| FREQUENCY_ FAILURE_ COUNT | 4 | This field indicates the number of times during chirping in the previous monitoring period in which the measured frequency error violated the allowed threshold. Frequency error threshold violation is counted every 10 ns. Bits Parameter b31:19 RESERVED b18:0 Failure count, unsigned number |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.7 Sub block 0x1026 – AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the external signal voltage values measured using the GPADC. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.98: AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1026 |
| SBLKLEN | 2 | Value = 28 |

Continued on next page

Table 5.98 – continued from previous page

| | | |
|-------------------------------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | <p>Status flags indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit Definition</p> <p>b0 STATUS_ANALOGTEST1</p> <p>b1 STATUS_ANALOGTEST2</p> <p>b2 STATUS_ANALOGTEST3</p> <p>b3 STATUS_ANALOGTEST4</p> <p>b4 STATUS_ANAMUX</p> <p>b5 STATUS_VSENSE</p> <p>b15:6 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> |
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> |
| EXTERNAL_ANALOG_SIGNAL_VALUES | 12 | <p>MEASURED_VALUE</p> <p>Bytes SIGNAL</p> <p>1:0 ANALOGTEST1</p> <p>3:2 ANALOGTEST2</p> <p>5:4 ANALOGTEST3</p> <p>7:6 ANALOGTEST4</p> <p>9:8 ANAMUX</p> <p>11:10 VSENSE</p> <p>1 LSB = 1.8V/1024</p> |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> |

5.14.8 Sub block 0x1027 – AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about Internal TX0 internal analog signals. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.99: AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_
SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x1027 |
| SBLKLEN | 2 | Value = 16 |
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SUPPLY_TX0 b1 STATUS_DCBIAS_TX0 b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.9 Sub block 0x1028 – AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_

REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about Internal TX1 internal analog signals. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.100: AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_
AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1028 |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.100 – continued from previous page

| | | |
|--------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SUPPLY_TX1 b1 STATUS_DCBIAS_TX1 b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.10 Sub block 0x1029 – AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about Internal TX2 internal analog signals. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.101: AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1029 |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.101 – continued from previous page

| | | |
|--------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flag indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SUPPLY_TX2 b1 STATUS_DCBIAS_TX2 b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.11 Sub block 0x102A – AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about Internal RX internal analog signals. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.102: AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x102A |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.102 – continued from previous page

| | | |
|--------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flags indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SUPPLY_RX0 b1 STATUS_SUPPLY_RX1 b2 STATUS_SUPPLY_RX2 b3 STATUS_SUPPLY_RX3 b4 STATUS_DCBIAS_RX0 b5 STATUS_DCBIAS_RX1 b6 STATUS_DCBIAS_RX2 b7 STATUS_DCBIAS_RX3 b15:8 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| RESERVED | 3 | 0x000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.12 Sub block 0x102B – AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about Internal PM, CLK and LO subsystems' internal analog signals. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.103: AWR_MONITOR_PM_CLK_LO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x102B |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.103 – continued from previous page

| | | |
|----------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flags indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_SUPPLY_PMCLKLO b1 STATUS_DCBIAS_PMCLKLO b2 STATUS_LVDS_PMCLKLO (Use this status bit only if LVDS is used, else ignore this) b3 STATUS_SYNC_20G (Use this field only in cascade configuration) b15:4 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies |
| SYNC_20G_POWER | 1 | Monitored 20 GHz signal power, signed number Unit: 1 LSB = 0.5 dBm |
| RESERVED | 2 | 0x000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.13 Sub block 0x102C – AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing information about the measured value of the GPADC input DC signals whose measurements were enabled. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.104: AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x102C |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.104 – continued from previous page

| | | |
|------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS_FLAGS | 2 | Status flags indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_GPADC_REF1 b1 STATUS_GPADC_REF2 b15:2 RESERVED 0 – FAIL or check wasn't done 1 – PASS |
| ERROR_CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error |
| GPADC_REF1_VALUE | 2 | The measured GPADC outputs corresponding to internal DC signal (GPADC_REF1, expected level 0.45V) is reported here. 1 LSB = 1.8V/1024 |
| GPADC_REF2_VALUE | 2 | The measured GPADC outputs corresponding to internal DC signal (GPADC_REF2, expected level 1.2V) is reported here. 1 LSB = 1.8V/1024 |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.14 Sub block 0x102D – AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured PLL control voltage values during explicit monitoring chirps. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.105: AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x102D |
| SBLKLEN | 2 | Value = 32 |

Continued on next page

Table 5.105 – continued from previous page

| STATUS_FLAGS | 2 | <p>Status flags indicating pass fail results corresponding to various threshold checks under this monitor.</p> <p>Bit STATUS_FLAG for monitor</p> <p>b0 STATUS_APLL_VCTRL</p> <p>b1 STATUS_SYNTH_VCO1_VCTRL_MAX_FREQ</p> <p>b2 STATUS_SYNTH_VCO1_VCTRL_MIN_FREQ</p> <p>b3 STATUS_SYNTH_VCO1_SLOPE (Not supported in 1st gen devices, please refer latest DFP 1.x release note)</p> <p>b4 STATUS_SYNTH_VCO2_VCTRL_MAX_FREQ</p> <p>b5 STATUS_SYNTH_VCO2_VCTRL_MIN_FREQ</p> <p>b6 STATUS_SYNTH_VCO2_SLOPE (Not supported in 1st gen devices, please refer latest DFP 1.x release note)</p> <p>b15:7 RESERVED</p> <p>0 – FAIL or check wasn't done</p> <p>1 – PASS</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|-------|-----|------------|------|-----|---------------------------|------|-----|---------------------------|------|-----|------------------|---------|-----|---------------------------|------|-------|---------------------------|------|-------|------------------|---------|-------|----------|----------|
| ERROR_CODE | 2 | <p>Indicates any error reported during monitoring</p> <p>Value of 0 indicates no error</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PLL_CONTROL_VOLTAGE_VALUES | 16 | <p>The measured values of PLL control voltage levels and Synthesizer VCO slopes are reported here.</p> <p>Byte numbers corresponding to different control voltage values reported in this field are here:</p> <table border="1"> <thead> <tr> <th>Bytes</th> <th>SIGNAL</th> <th>1 LSB</th> </tr> </thead> <tbody> <tr> <td>1:0</td> <td>APLL_VCTRL</td> <td>1 mV</td> </tr> <tr> <td>3:2</td> <td>SYNTH_VCO1_VCTRL_MAX_FREQ</td> <td>1 mV</td> </tr> <tr> <td>5:4</td> <td>SYNTH_VCO1_VCTRL_MIN_FREQ</td> <td>1 mV</td> </tr> <tr> <td>7:6</td> <td>SYNTH_VCO1_SLOPE</td> <td>1 MHz/V</td> </tr> <tr> <td>9:8</td> <td>SYNTH_VCO2_VCTRL_MAX_FREQ</td> <td>1 mV</td> </tr> <tr> <td>11:10</td> <td>SYNTH_VCO2_VCTRL_MIN_FREQ</td> <td>1 mV</td> </tr> <tr> <td>13:12</td> <td>SYNTH_VCO2_SLOPE</td> <td>1 MHz/V</td> </tr> <tr> <td>15:14</td> <td>RESERVED</td> <td>RESERVED</td> </tr> </tbody> </table> | Bytes | SIGNAL | 1 LSB | 1:0 | APLL_VCTRL | 1 mV | 3:2 | SYNTH_VCO1_VCTRL_MAX_FREQ | 1 mV | 5:4 | SYNTH_VCO1_VCTRL_MIN_FREQ | 1 mV | 7:6 | SYNTH_VCO1_SLOPE | 1 MHz/V | 9:8 | SYNTH_VCO2_VCTRL_MAX_FREQ | 1 mV | 11:10 | SYNTH_VCO2_VCTRL_MIN_FREQ | 1 mV | 13:12 | SYNTH_VCO2_SLOPE | 1 MHz/V | 15:14 | RESERVED | RESERVED |
| Bytes | SIGNAL | 1 LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1:0 | APLL_VCTRL | 1 mV | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3:2 | SYNTH_VCO1_VCTRL_MAX_FREQ | 1 mV | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5:4 | SYNTH_VCO1_VCTRL_MIN_FREQ | 1 mV | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7:6 | SYNTH_VCO1_SLOPE | 1 MHz/V | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9:8 | SYNTH_VCO2_VCTRL_MAX_FREQ | 1 mV | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11:10 | SYNTH_VCO2_VCTRL_MIN_FREQ | 1 mV | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13:12 | SYNTH_VCO2_SLOPE | 1 MHz/V | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15:14 | RESERVED | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 5.105 – continued from previous page

| | | |
|------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Only the fields corresponding to the enabled monitors are valid. The failure thresholds are based on the following: Valid VCTRL values are [140 to 1400] mV. Valid VCO1_SLOPE values are [1760 to 2640] MHz/V. Valid VCO2_SLOPE values are [3520 to 5280] MHz/V. NOTE: The VCOx.SLOPE should be ignored when synth fault is injected. |
| RESERVED | 4 | 0x00000000 |
| TIME_STAMP | 4 | This field indicates when the last monitoring in the enabled set was performed. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) |

5.14.15 Sub block 0x102E – AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB

This API is a monitoring report API which the AWR device sends to the host, containing information about the relative frequency measurements. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.106: AWR_MONITOR_DUAL_CLOCK_COMP_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x102E |
| SBLKLEN | 2 | Value = 32 |
| STATUS_FLAGS | 2 | Status flags indicating pass fail results corresponding to various threshold checks under this monitor. Bit STATUS_FLAG for monitor b0 STATUS_CLK_PAIR0 b1 STATUS_CLK_PAIR1 b2 STATUS_CLK_PAIR2 b3 STATUS_CLK_PAIR3 b4 STATUS_CLK_PAIR4 b5 STATUS_CLK_PAIR5 (Not supported, refer latest release note) b15:6 RESERVED 0 – FAIL or check wasn't done 1 – PASS |

Continued on next page

Table 5.106 – continued from previous page

| ERROR.CODE | 2 | Indicates any error reported during monitoring Value of 0 indicates no error | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|--------------------------|-----|---|----------|-----|---|----------|-----|---|----------|-----|---|-----------|-----|---|-----------|-------|---|---------------------------------------------------------|-------|----------|----------|
| FREQ.MEAS. VALUES | 16 | <p>The measured clock frequencies from the enabled clock pair measurements are reported here.</p> <p>Byte numbers corresponding to different frequency measurement values reported in this field are here:</p> <table border="1"> <thead> <tr> <th>Bytes</th> <th>CLOCK PAIR</th> <th>MEASURED CLOCK FREQUENCY</th> </tr> </thead> <tbody> <tr> <td>1:0</td> <td>0</td> <td>BSS_600M</td> </tr> <tr> <td>3:2</td> <td>1</td> <td>BSS_200M</td> </tr> <tr> <td>5:4</td> <td>2</td> <td>BSS_100M</td> </tr> <tr> <td>7:6</td> <td>3</td> <td>GPADC_10M</td> </tr> <tr> <td>9:8</td> <td>4</td> <td>RCOSC_10M</td> </tr> <tr> <td>11:10</td> <td>5</td> <td>RAMPGEN_100M (Not supported, refer latest release note)</td> </tr> <tr> <td>15:12</td> <td>RESERVED</td> <td>RESERVED</td> </tr> </tbody> </table> <p>1 LSB = 0.1 MHz, unsigned number</p> | Bytes | CLOCK PAIR | MEASURED CLOCK FREQUENCY | 1:0 | 0 | BSS_600M | 3:2 | 1 | BSS_200M | 5:4 | 2 | BSS_100M | 7:6 | 3 | GPADC_10M | 9:8 | 4 | RCOSC_10M | 11:10 | 5 | RAMPGEN_100M (Not supported, refer latest release note) | 15:12 | RESERVED | RESERVED |
| Bytes | CLOCK PAIR | MEASURED CLOCK FREQUENCY | | | | | | | | | | | | | | | | | | | | | | | | |
| 1:0 | 0 | BSS_600M | | | | | | | | | | | | | | | | | | | | | | | | |
| 3:2 | 1 | BSS_200M | | | | | | | | | | | | | | | | | | | | | | | | |
| 5:4 | 2 | BSS_100M | | | | | | | | | | | | | | | | | | | | | | | | |
| 7:6 | 3 | GPADC_10M | | | | | | | | | | | | | | | | | | | | | | | | |
| 9:8 | 4 | RCOSC_10M | | | | | | | | | | | | | | | | | | | | | | | | |
| 11:10 | 5 | RAMPGEN_100M (Not supported, refer latest release note) | | | | | | | | | | | | | | | | | | | | | | | | |
| 15:12 | RESERVED | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | | | | | | | | | |
| TIME.STAMP | 4 | <p>This field indicates when the last monitoring in the enabled set was performed.</p> <p>1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width)</p> | | | | | | | | | | | | | | | | | | | | | | | | |

5.14.16 Sub block 0x1031 – AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB

This API is a Monitoring Report API which the AWR device sends to the host, containing the measured RX mixer input voltage swing values. The AWR device sends this to host at the programmed periodicity or when failure occurs, as programmed by the configuration API SB.

Table 5.107: AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x1031 |
| SBLKLEN | 2 | Value = 24 |

Continued on next page

Table 5.107 – continued from previous page

| STATUS_FLAGS | 2 | Bit STATUS_FLAG for monitor b0 STATUS_MIXER.IN.POWER.RX0 b1 STATUS_MIXER.IN.POWER.RX1 b2 STATUS_MIXER.IN.POWER.RX2 b3 STATUS_MIXER.IN.POWER.RX3 b15:4 RESERVED 0 – FAIL or check wasn't done 1 – PASS | | | | | | | | | | |
|---------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|-----|---|-----|---|-----|---|-----|---|
| ERROR_CODE | 2 | Internal sanity check violations are reported here. Value = 0: No error Other values: Error (see error code definition matrix) | | | | | | | | | | |
| PROFILE_INDX | 1 | Profile Index for which this monitoring report applies | | | | | | | | | | |
| RESERVED | 3 | 0x000000 | | | | | | | | | | |
| RX_MIXER_IN_VOLTAGE_VALUE | 4 | The measured RX mixer input voltage swing values are reported here. The byte location of the value for each receivers is tabulated here: <table border="1"> <thead> <tr> <th>Receiver</th> <th>Byte Location</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>0</td> </tr> <tr> <td>RX1</td> <td>1</td> </tr> <tr> <td>RX2</td> <td>2</td> </tr> <tr> <td>RX3</td> <td>3</td> </tr> </tbody> </table> 1 LSB = 1800 mV/256, unsigned number Only the entries of enabled RX channels are valid. | Receiver | Byte Location | RX0 | 0 | RX1 | 1 | RX2 | 2 | RX3 | 3 |
| Receiver | Byte Location | | | | | | | | | | | |
| RX0 | 0 | | | | | | | | | | | |
| RX1 | 1 | | | | | | | | | | | |
| RX2 | 2 | | | | | | | | | | | |
| RX3 | 3 | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | |
| TIME_STAMP | 4 | When this monitoring began is indicated here. 1 LSB = 1 millisecond (time stamp rolls over upon exceeding allotted bit width) | | | | | | | | | | |

5.15 Sub blocks related to AWR_DEV_RFPOWERUP_MSG

NOTE: All device config APIs having sub block ID \geq 0x4000 are applicable only for MSS in AWR1243 RF front end devices, for other xWR1443, xWR1642 and xWR1843 devices, these APIs are for reference only.

5.15.1 Sub block 0x4000 – AWR_DEV_RFPOWERUP_SB

This sub block is a command to power up the BSS

[5.108](#) describes the content of this sub block.

Table 5.108: AWR_DEV_POWERUP_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4000 |
| SBLKLEN | 2 | Value = 4 |

5.16 Sub blocks related to AWR_DEV_CONF_SET_MSG

5.16.1 Sub block 0x4040 – AWR_DEV_MCUCLOCK_CONF_SET_SB

This sub block contains the configurations to setup the desired frequency of the MCU Clock that is output from the device.

Table 5.109 describes the contents of this sub block.

Table 5.109: AWR_DEV_MCUCLOCK_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|---------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4040 |
| SBLKLEN | 2 | Value = 8 |
| MCUCLOCK_CTRL | 1 | This field controls the enable-disable of the MCU clock. Value Description 0x0 Disable MCU clock 0x1 Enable MCU clock |
| MCUCLOCK_SRC | 1 | This field specifies the source of the MCU clock. Applicable only in case of MCU clock enable. Else ignored. Value Description 0x0 XTAL (as connected to the device) 0x2 600MHz PLL divided clock |
| SRCCLOCK_DIV | 1 | This field specifies the division factor to be applied to source clock. Applicable only in case of MCU clock enable. Else ignored. Value Description 0x0 Divide by 1 0x1 Divide by 2 0xFF Divide by 256 |
| RESERVED | 1 | 0x00 |

5.16.2 Sub block 0x4041 – AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB

This sub block contains the configuration of the data format of the samples received over the receive chain to be transferred out to an external host over the configured data path (LVDS or CSI2).

Table 5.110 describes the content of this sub block.

Table 5.110: AWR_DEV_RX_DATA_FORMAT_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4041 |
| SBLKLEN | 2 | Value = 16 |
| RX_CHAN_EN | 2 | Bits Definition b0 RX_CHAN0_EN 0 Disable RX Channel 0 1 Enable RX Channel 0 b1 RX_CHAN0_EN 0 Disable RX Channel 1 1 Enable RX Channel 1 b2 RX_CHAN0_EN 0 Disable RX Channel 2 1 Enable RX Channel 2 b3 RX_CHAN0_EN 0 Disable RX Channel 3 1 Enable RX Channel 3 b15:4 RESERVED |
| NUM_ADC.BITS | 2 | Bits Definition b1:0 00 12 bits 01 14 bits 10 16 bits Other Reserved b15:2 RESERVED |
| ADC.OUT_FMT | 2 | Bits Definition b1:0 00 Real 01 Complex Other Reserved b15:2 RESERVED |

Continued on next page

Table 5.110 – continued from previous page

| | | | |
|------------------|---|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IQ_SWAP_SEL | 1 | Bits | Definition |
| | | b1:0 | To swap the IQ samples (if complex format) 00 Sample interleave mode – I first 01 Sample interleave mode – Q first Other Reserved |
| | | b7:2 | RESERVED |
| CHAN_INTER-LEAVE | 1 | Bits | Definition |
| | | b1:0 | Channel interleaving of the samples stored in the ADC buffer to be transferred out on the data path. 00 Interleaved mode of storage 01 Non-interleaved mode of storage Other Reserved |
| | | b7:2 | RESERVED |
| RESERVED | 4 | 0x00000000 | |

5.16.3 Sub block 0x4042 – AWR_DEV_RX_DATA_PATH_CONF_SET_SB

This sub block contains the configurations of the data path to transfer the captured ADC samples received over the receive chain to be transferred out to an external host.

Table 5.111 describes the content of this sub block.

Table 5.111: AWR_DEV_RX_DATA_PATH_CONF_SB contents

| Field Name | Number of bytes | Description |
|---------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4042 |
| SBLKLEN | 2 | Value = 12 |
| DATA_INTF_SEL | 1 | This field specifies the data path selected to transfer the Radar info. Value Description 0x0 CSI2 interface select 0x1 LVDS interface select 0x2 SPI interface selected (applicable when a large inter-frame time is provided to transfer the data over the SPI) |

Continued on next page

Table 5.111 – continued from previous page

| | | | |
|----------------------|---|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DATA_TRANS_ FMT_PKT0 | 1 | Bits | Description |
| | | b5:0 | Packet 0 content selection Value Definition 000001 ADC 000110 CP_ADC (See note at the bottom of this table) 001001 ADC_CP 110110 CP_ADC_CQ (See note at the bottom of this table) |
| | | b7:6 | Packet 0 virtual channel number (valid only for CSI2) Value Definition 00 Virtual channel number 0 (Default) 01 Virtual channel number 1 10 Virtual channel number 2 11 Virtual channel number 3 |
| DATA_TRANS_ FMT_PKT1 | 1 | Bits | Description |
| | | b5:0 | Packet 1 content selection Value Definition 000000 Suppress packet 1 transmission 001110 CP_CQ (See note at the bottom of this table) 001011 CQ_CP (See note at the bottom of this table) |
| | | b7:6 | Packet 1 virtual channel number (valid only for CSI2) Value Definition 00 Virtual channel number 0 (Default) 01 Virtual channel number 1 10 Virtual channel number 2 11 Virtual channel number 3 |

Continued on next page

Table 5.111 – continued from previous page

| CQ_CONFIG | 1 | <p>This specifies the data size of CQ samples on the lanes</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>b1:0 Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>00</td> <td>12 bit</td> </tr> <tr> <td></td> <td></td> <td>01</td> <td>14 bit</td> </tr> <tr> <td></td> <td></td> <td>10</td> <td>16 bit</td> </tr> <tr> <td></td> <td></td> <td>11</td> <td>RESERVED</td> </tr> </tbody> </table> <p>b7:2 RESERVED</p> <p>NOTE: The CQ size can be configured only if CQ and ADC data is sent in separate packets. When ADC and CQ is sent in the same packet, then CQ size will be same as ADC data size.</p> | Bits | Description | b1:0 Value | Definition | | | 00 | 12 bit | | | 01 | 14 bit | | | 10 | 16 bit | | | 11 | RESERVED |
|----------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------|------------|------------|--|--|----|--------|--|--|----|--------|--|--|----|--------|--|--|----|----------|
| Bits | Description | b1:0 Value | Definition | | | | | | | | | | | | | | | | | | | |
| | | 00 | 12 bit | | | | | | | | | | | | | | | | | | | |
| | | 01 | 14 bit | | | | | | | | | | | | | | | | | | | |
| | | 10 | 16 bit | | | | | | | | | | | | | | | | | | | |
| | | 11 | RESERVED | | | | | | | | | | | | | | | | | | | |
| CQ0_TRANS_SIZE | 1 | <p>Number of samples (in 16 bit halfwords) of CQ0 data to be transferred. Valid range [32 halfwords to 128 halfwords] Value 0 = Disabled.</p> <p>NOTE: Ensure that the number of halfwords specified are a multiple of the number of lanes selected.</p> | | | | | | | | | | | | | | | | | | | | |
| CQ1_TRANS_SIZE | 1 | <p>Number of samples (in 16 bit halfwords) of CQ1 data to be transferred. Valid range [32 halfwords to 128 halfwords] Value 0 = Disabled.</p> <p>NOTE: Ensure that the number of halfwords specified are a multiple of the number of lanes selected.</p> | | | | | | | | | | | | | | | | | | | | |
| CQ2_TRANS_SIZE | 1 | <p>Number of samples (in 16 bit halfwords) of CQ2 data to be transferred. Valid range [32 halfwords to 128 halfwords] Value 0 = Disabled.</p> <p>NOTE: Ensure that the number of halfwords specified are a multiple of the number of lanes selected.</p> | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | | | | |

| NOTE1: | <p>CP is Chirp Parameter information which is defined for each RX as follows</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>b11:0</td> <td> Chirp number In legacy frame configuration, chirp number for starts from 1 and increments for each chirp within the frame and resets to 0 for the next frame. In advanced frame configuration chirp number starts from 1 and increments for each chirp within the burst and resets to 0 for the next burst. </td> </tr> <tr> <td>b15:12</td> <td>RESERVED</td> </tr> <tr> <td>b17:16</td> <td> Channel number The receive channel number which is encoded as <table border="1"> <thead> <tr> <th>Bit</th> <th>RX</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>RX0</td> </tr> <tr> <td>01</td> <td>RX1</td> </tr> <tr> <td>10</td> <td>RX2</td> </tr> <tr> <td>11</td> <td>RX3</td> </tr> </tbody> </table> </td> </tr> <tr> <td>b21:18</td> <td> Profile number The profile number to which the chirp belongs </td> </tr> <tr> <td>b31:22</td> <td>RESERVED</td> </tr> </tbody> </table> | Bit | Description | b11:0 | Chirp number In legacy frame configuration, chirp number for starts from 1 and increments for each chirp within the frame and resets to 0 for the next frame. In advanced frame configuration chirp number starts from 1 and increments for each chirp within the burst and resets to 0 for the next burst. | b15:12 | RESERVED | b17:16 | Channel number The receive channel number which is encoded as <table border="1"> <thead> <tr> <th>Bit</th> <th>RX</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>RX0</td> </tr> <tr> <td>01</td> <td>RX1</td> </tr> <tr> <td>10</td> <td>RX2</td> </tr> <tr> <td>11</td> <td>RX3</td> </tr> </tbody> </table> | Bit | RX | 00 | RX0 | 01 | RX1 | 10 | RX2 | 11 | RX3 | b21:18 | Profile number The profile number to which the chirp belongs | b31:22 | RESERVED |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|----|-----|----|-----|----|-----|----|-----|--------|-----------------------------------------------------------------|--------|----------|
| Bit | Description | | | | | | | | | | | | | | | | | | | | | | |
| b11:0 | Chirp number In legacy frame configuration, chirp number for starts from 1 and increments for each chirp within the frame and resets to 0 for the next frame. In advanced frame configuration chirp number starts from 1 and increments for each chirp within the burst and resets to 0 for the next burst. | | | | | | | | | | | | | | | | | | | | | | |
| b15:12 | RESERVED | | | | | | | | | | | | | | | | | | | | | | |
| b17:16 | Channel number The receive channel number which is encoded as <table border="1"> <thead> <tr> <th>Bit</th> <th>RX</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>RX0</td> </tr> <tr> <td>01</td> <td>RX1</td> </tr> <tr> <td>10</td> <td>RX2</td> </tr> <tr> <td>11</td> <td>RX3</td> </tr> </tbody> </table> | Bit | RX | 00 | RX0 | 01 | RX1 | 10 | RX2 | 11 | RX3 | | | | | | | | | | | | |
| Bit | RX | | | | | | | | | | | | | | | | | | | | | | |
| 00 | RX0 | | | | | | | | | | | | | | | | | | | | | | |
| 01 | RX1 | | | | | | | | | | | | | | | | | | | | | | |
| 10 | RX2 | | | | | | | | | | | | | | | | | | | | | | |
| 11 | RX3 | | | | | | | | | | | | | | | | | | | | | | |
| b21:18 | Profile number The profile number to which the chirp belongs | | | | | | | | | | | | | | | | | | | | | | |
| b31:22 | RESERVED | | | | | | | | | | | | | | | | | | | | | | |
| NOTE2: | CQ is Chirp Quality information which is defined in Section 9 | | | | | | | | | | | | | | | | | | | | | | |

5.16.4 Sub block 0x4043 – AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB

This sub block contains the configurations to enables the lanes of the LVDS/CSI2 path to transfer Radar information to an external host.

Table 5.112 describes the content of this sub block.

Table 5.112: AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4043 |
| SBLKLEN | 2 | Value = 8 |

Continued on next page

Table 5.112 – continued from previous page

| | | | |
|----------|---|--------|-------------------------------------------------|
| LANE_EN | 2 | Bits | Description |
| | | b0 | LANE0_EN 0 Disable lane 0 1 Enable lane 0 |
| | | b1 | LANE1_EN 0 Disable lane 1 1 Enable lane 1 |
| | | b2 | LANE2_EN 0 Disable lane 2 1 Enable lane 2 |
| | | b3 | LANE3_EN 0 Disable lane 3 1 Enable lane 3 |
| | | b15:4 | RESERVED |
| RESERVED | 2 | 0x0000 | |

5.16.5 Sub block 0x4044 – AWR_DEV_RX_DATA_PATH_CLK_SET_SB

This sub block contains the clock configurations for data transfer on the LVDS/CSI2 lanes. Table 5.113 describes the content of this sub block.

Table 5.113: AWR_DEV_RX_DATA_PATH_CLK_SET_SB contents

| Field Name | Number of bytes | Description | |
|-------------------------------------------------------------------------------|-----------------|----------------|-----------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4044 | |
| SBLKLEN | 2 | Value = 8 | |
| LANE_CLK_CFG (Selection valid only for LVDS. For CSI2, DDR is used always) | 1 | Bits | Description |
| | | b0 | BIT_CLK_SEL 0 SDR clock 1 DDR clock (Only valid value for CSI2) |
| | | b7:1 | RESERVED |

Continued on next page

Table 5.113 – continued from previous page

| | | |
|-----------|---|---------------------------|
| DATA_RATE | 1 | Data rate selection |
| | | Value Description |
| | | 0000b 900 Mbps (DDR only) |
| | | 0001b 600 Mbps (DDR only) |
| | | 0010b 450 Mbps (SDR, DDR) |
| | | 0011b 400 Mbps (DDR only) |
| | | 0100b 300 Mbps (SDR, DDR) |
| | | 0101b 225 Mbps (DDR only) |
| | | 0110b 150 Mbps (DDR only) |
| | | Others RESERVED |
| RESERVED | 2 | 0x0000 |

5.16.6 Sub block 0x4045 – AWR_DEV_LVDS_CFG_SET_SB

This sub block contains the configurations of the LVDS lanes.

Table 5.114 describes the content of this sub block.

Table 5.114: AWR_DEV_LVDS_CFG_SET_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4045 |
| SBLKLEN | 2 | Value = 8 |
| LANE_FMT_MAP | 2 | LANE0 Format Map. The mapping of the data on the lanes is depicted in the figure below 0x0000 Format map 0 0x0001 Format map 1 |

Continued on next page

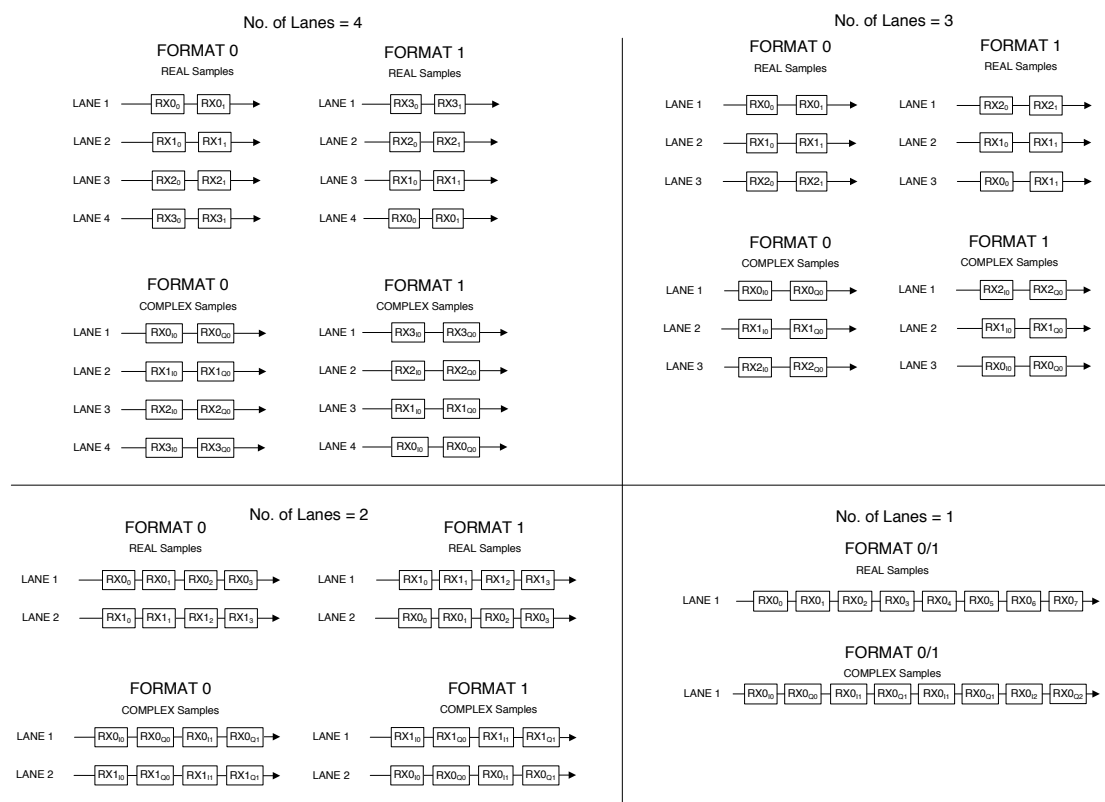
Table 5.114 – continued from previous page

| LANE_PARAM_CFG | 2 | Bit | Description |
|----------------|---|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | b0 | MSB_FIRST 0 Disable (LSB First) 1 Enable (MSB First) |
| | | b1 | Packet End Pulse Enable 0 Disable 1 Enable |
| | | b2 | CRC Enable 0 Disable 1 Enable |
| | | b7:3 | RESERVED |
| | | b8 | Configures LSB/MSB first for CRC 0 CRC value swapped wrt to MSB_FIRST setting 1 CRC value follows MSB_FIRST setting |
| | | b9 | Frame clock state during idle 0 Frame clock is held low 1 Frame clock is held high |
| | | b10 | Frame clock period for CRC(when CRC enabled - b2) 0 32-bit CRC is trasmitted as single sample with frame clock set to 16high, 16low configuration 1 32-bit CRC is trasmitted as single sample with frame clock set to 8high, 8low configuration |
| | | b11 | Bit clock state during idle 0 Bit clock toggles during idle when there are no transmission 1 Bit clock doesn't toggle during idle when there are no transmission, the value of bit clock is held low |
| | | b12 | CRC inversion control(when CRC enabled - b2) 0 The calculated value of 32-bit ethernet polynomial CRC is inverted and sent out 1 The calculated value of 32-bit ethernet polynomial CRC is sent without inversion |
| | | b15:13 | RESERVED |

Continued on next page

Table 5.114 – continued from previous page

The mapping of the 8 sample ($8 \times 16 = 128$ bit) information onto the serial interface lanes is determined by the LANE_FMT_MAP parameter. The choice of format map translating to the transfer of data on the lanes is depicted in the image below (the x axis represents time – hence the samples are as available on the lanes in time and the receiver will receive the samples in the reverse order as depicted below).


Figure 5.4: Lane formats and the order of receiving the data from the lanes

5.16.7 Sub block 0x4046 – AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB

This sub block contains the configurations of the data path to transfer the captured ADC samples continuously without any break to an external host.

Table 5.115 describes the content of this sub block.

Table 5.115: AWR_DEV_RX_CONTSTREAMING_MODE_CFG_SET_SB contents

| Field Name | Number of bytes | Description |
|---------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4046 |
| SBLKLEN | 2 | Value = 8 |
| CONT_STREAMING_MODE | 2 | Continuous streaming mode enable Value Description 0x0 Continuous streaming mode data transfer disable 0x1 Continuous streaming mode data transfer enable |
| RESERVED | 2 | 0x0000 |

5.16.8 Sub block 0x4047 – AWR_DEV_CSI2_CFG_SET_SB

This sub block contains the various configurations of the parameters of the CSI2 module. Table 5.116 describes the content of this sub block.

Table 5.116: AWR_DEV_CSI2_CFG_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4047 |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 5.116 – continued from previous page

| LANE_POS_POL_SEL | 4 | Bits | Definition |
|------------------|---|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | b2:0 | DATA_LANE0.POS Valid values (Should be a unique position if lane 0 is enabled, ignored if lane 0 is not enabled): 000b – Unused, 001b – Position 1 (default), 010b – Position 2, 011b – Position 3, 100b – Position 4, 101b – Position 5 |
| | | b3 | DATA_LANE0.POL 0b – PLUSMINUS pin order, 1b – MINUSPLUS pin order |
| | | b6:4 | DATA_LANE1.POS Valid values (Should be a unique position if lane 1 is enabled, ignored if lane 1 is not enabled): 000b – Unused, 001b – Position 1, 010b – Position 2 (default), 011b – Position 3, 100b – Position 4, 101b – Position 5 |
| | | b7 | DATA_LANE1.POL 0b – PLUSMINUS pin order, 1b – MINUSPLUS pin order |
| | | b10:8 | DATA_LANE2.POS Valid values (Should be a unique position if lane 2 is enabled, ignored if lane 2 is not enabled): 000b – Unused, 001b – Position 1, 010b – Position 2, 011b – Position 3, 100b – Position 4 (default), 101b – Position 5 |
| | | b11 | DATA_LANE2.POL 0b – PLUSMINUS pin order, 1b – MINUSPLUS pin order |
| | | b14:12 | DATA_LANE3.POS Valid values (Should be a unique position if lane 3 is enabled, ignored if lane 3 is not enabled): 000b – Unused, 001b – Position 1, 010b – Position 2, 011b – Position 3, 100b – Position 4, 101b – Position 5 (default) |
| | | b15 | DATA_LANE3.POL 0b – PLUSMINUS pin order, 1b – MINUSPLUS pin order |
| | | b18:16 | CLOCK.POS Valid values (Should be a unique position): 0000b – Unused, 001b – Unused, 010b – Position 2, 011b – Position 3 (default), 100b – Position 4 |
| | | b19 | CLOCK.POL 0b – PLUSMINUS pin order, 1b – MINUSPLUS pin order |
| | | b31:20 | RESERVED |

Continued on next page

Table 5.116 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
|----------|---|------------|

5.16.9 Sub block 0x4048 – AWR_DEV_PMICLOCK_CONF_SET_SB

This sub block contains the configurations to setup the desired frequency of the PMIC Clock that is output from the device. The configurations also allow setting up the dither values for the clock. Table 5.117 describes the contents of this sub block.

Table 5.117: AWR_DEV_PMICLOCK_CONF_SET_SB contents

| Field Name | Number of bytes | Description |
|---------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x4048 |
| SBLKLEN | 2 | Value = 16 |
| PMICLOCK_CTRL | 1 | This field controls the enable-disable of the PMIC clock. Value Description 0x0 Disable PMIC clock 0x1 Enable PMIC clock |
| PMICLOCK_SRC | 1 | This field specifies the source of the PMIC clock. Applicable only in case of PMIC clock enable. Else ignored. Value Description 0x0 XTAL (as connected to the device) 0x2 600 MHz PLL divided clock |
| SRCLOCK_DIV | 1 | This field specifies the division factor to be applied to source clock. Applicable only in case of PMIC clock enable. Else ignored. Value Description 0x0 Divide by 1 0x1 Divide by 2 0xFF Divide by 256 |

Continued on next page

Table 5.117 – continued from previous page

| | | |
|---------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MODE_SELECT | 1 | <p>This field specifies the mode of operation for the PMIC clock generation. Applicable only in case of PMIC clock enable. Else ignored.</p> <p>Value Description</p> <p>0x0 Continuous mode (free running mode where the frequency change/jump is triggered based on configured number of PMIC clock ticks)</p> <p>0x1 Chirp-to-Chirp staircase mode (frequency change/jump is triggered at every chirp boundary)</p> |
| FREQ_SLOPE | 4 | <p>Applicable only in case of PMIC clock enable. Else ignored.</p> <p>Bit Description</p> <p>b25:0 Frequency slope value to be applied in [7.18] unsigned format 1 LSB = $1/2^{18}$</p> <p>b31:26 RESERVED</p> <p>In continuous mode this value is accumulated every PMIC clock tick with the seed as MIN_NDIV_VAL till MAX_NDIV_VAL is reached</p> <p>In the stair case mode this value is accumulated every chirp with the seed as MIN_NDIV_VAL till MAX_NDIV_VAL is reached</p> |
| MIN_NDIV_VAL | 1 | <p>Applicable only in case of PMIC clock enable. Else ignored.</p> <p>Minimum allowed divider value (depends upon the highest desired clock frequency)</p> |
| MAX_NDIV_VAL | 1 | <p>Applicable only in case of PMIC clock enable. Else ignored.</p> <p>Maximum allowed divider value (depends upon the lowest desired clock frequency)</p> |
| CLK_DITHER_EN | 1 | <p>Applicable only in case of PMIC clock enable and frequency slope is non-zero. Else ignored.</p> <p>This field controls the enable-disable of the clock dithering. Adds a pseudo random real number (0 or 1) to the accumulated divide value. Hence it brings a random dithering of 1 LSB.</p> <p>Value Description</p> <p>0x0 Clock dithering disabled</p> <p>0x1 Clock dithering enabled</p> |
| RESERVED | 1 | 0x00 |

Example 1. PMIC clock with no slope in continuous mode

Objective: To configure the PMIC clock at frequency of 2 MHz with no slope.

Configurations:

1. PMICCLK_SRC = 0x2 (600 MHz PLL divided clock)
2. SRCLOCK_DIV = 29, Reference clock = $600 \text{ MHz} / (29 + 1) = 20 \text{ MHz}$
3. MIN_NDIV_VAL = MAX_NDIV_VAL = 10 (Computed as $20 \text{ MHz} / 2.0 \text{ MHz}$)
4. FREQ_SLOPE = 0

With the above configuration, the PMIC clock frequency would be $\text{PMIC clock} = (20 \text{ MHz} / 10) = 2 \text{ MHz}$

Example 2. Dithered PMIC clock with slope in chirp-to-chirp staircase mode

Objective: To configure a dithered PMIC clock at frequencies ranging from 2 MHz to 2.5 MHz over 32 chirps.

Configurations:

1. PMICCLK_SRC = 0x2 (600 MHz PLL divided clock)
2. SRCLOCK_DIV = 2, Reference clock = $600 \text{ MHz} / (2 + 1) = 200 \text{ MHz}$
3. MODE_SELECT = 1
4. FREQ_SLOPE = 169125 (Computed as $(\text{MAX_NDIV_VAL} - \text{MIN_NDIV_VAL}) \times 2^{18} / 31$)
5. MIN_NDIV_VAL = 80 (Computed as $200 \text{ MHz} / 2.5 \text{ MHz}$)
6. MAX_NDIV_VAL = 100 (Computed as $200 \text{ MHz} / 2.0 \text{ MHz}$)
7. CLK_DITHER_EN = 1

With the above configuration, the PMIC clock frequency would be vary between $(200 \text{ MHz} / 80)$ and $(200 \text{ MHz} / 100)$ in steps of $(200 \text{ MHz} / [(80 + (N \times \text{FREQ_SLOPE} / 2^{18} + X))])$ where

- N = Chirp number
- X = random fractional value in the range (0, 1) that adds the dither

The PMIC clock frequency is determined by the clock divider value which starts with a value of 100, providing a PMIC clock of 2 MHz for the 1st chirp, decrementing the divider by $\text{FREQ_SLOPE} / 2^{18} = 0.64516$ every chirp and finally reaching a value of 20 for the 32nd chirp providing a PMIC clock of 2.5 MHz.

Table 5.118: PMIC clock frequency across chirps in chirp-to-chirp staircase mode in an example when PMIC clock varies from 2 MHz to 2.5 MHz in 32 chirps

| Chirp Number | PMIC Clock Frequency (MHz) | Calculation |
|--------------|----------------------------|--------------------------------------|
| 1 | 2.50000 | $200/(80 + 0 \times 169125/2^{18})$ |
| 2 | 2.48000 | $200/(80 + 1 \times 169125/2^{18})$ |
| 3 | 2.46032 | $200/(80 + 2 \times 169125/2^{18})$ |
| 4 | 2.44094 | $200/(80 + 3 \times 169125/2^{18})$ |
| 5 | 2.42188 | $200/(80 + 4 \times 169125/2^{18})$ |
| 6 | 2.40310 | $200/(80 + 5 \times 169125/2^{18})$ |
| 7 | 2.38462 | $200/(80 + 6 \times 169125/2^{18})$ |
| 8 | 2.36641 | $200/(80 + 7 \times 169125/2^{18})$ |
| 9 | 2.34848 | $200/(80 + 8 \times 169125/2^{18})$ |
| 10 | 2.33083 | $200/(80 + 9 \times 169125/2^{18})$ |
| 11 | 2.31343 | $200/(80 + 10 \times 169125/2^{18})$ |
| 12 | 2.29630 | $200/(80 + 11 \times 169125/2^{18})$ |
| 13 | 2.27941 | $200/(80 + 12 \times 169125/2^{18})$ |
| 14 | 2.26277 | $200/(80 + 13 \times 169125/2^{18})$ |
| 15 | 2.24638 | $200/(80 + 14 \times 169125/2^{18})$ |
| 16 | 2.23022 | $200/(80 + 15 \times 169125/2^{18})$ |
| 17 | 2.21429 | $200/(80 + 16 \times 169125/2^{18})$ |
| 18 | 2.19858 | $200/(80 + 17 \times 169125/2^{18})$ |
| 19 | 2.18310 | $200/(80 + 18 \times 169125/2^{18})$ |
| 20 | 2.16783 | $200/(80 + 19 \times 169125/2^{18})$ |
| 21 | 2.15278 | $200/(80 + 20 \times 169125/2^{18})$ |
| 22 | 2.13793 | $200/(80 + 21 \times 169125/2^{18})$ |
| 23 | 2.12329 | $200/(80 + 22 \times 169125/2^{18})$ |
| 24 | 2.10884 | $200/(80 + 23 \times 169125/2^{18})$ |
| 25 | 2.09459 | $200/(80 + 24 \times 169125/2^{18})$ |
| 26 | 2.08054 | $200/(80 + 25 \times 169125/2^{18})$ |
| 27 | 2.06667 | $200/(80 + 26 \times 169125/2^{18})$ |
| 28 | 2.05298 | $200/(80 + 27 \times 169125/2^{18})$ |
| 29 | 2.03947 | $200/(80 + 28 \times 169125/2^{18})$ |
| 30 | 2.02614 | $200/(80 + 29 \times 169125/2^{18})$ |
| 31 | 2.01299 | $200/(80 + 30 \times 169125/2^{18})$ |
| 32 | 2.00000 | $200/(80 + 31 \times 169125/2^{18})$ |

Example 3. Dithered PMIC clock with slope in continuous mode

Objective: To configure a dithered PMIC clock at frequencies ranging from 2 MHz to 2.5 MHz over 100 μ s.

Configurations:

1. PMICCLK_SRC = 0x2 (600 MHz PLL divided clock)
2. SRCLOCK_DIV = 2, Reference clock = 600 MHz / (2 + 1) = 200 MHz
3. MODE_SELECT = 0
4. FREQ_SLOPE = 23302 (Computed as $(MAX_NDIV_VAL - MIN_NDIV_VAL) \cdot 2^{18} / (100 \mu s \cdot (2.5 \text{ MHz} + 2 \text{ MHz}) / 2)$)
5. MIN_NDIV_VAL = 80 (Computed as 200 MHz / 2.5 MHz)
6. MAX_NDIV_VAL = 100 (Computed as 200 MHz / 2.0 MHz)
7. CLK_DITHER_EN = 1

With the above configuration, the PMIC clock frequency would be PMIC clock would vary between = (200 MHz / 80) to (200 MHz / 100) in steps of $(200 \text{ MHz} / [(80 + (N \times 23302 / 2^{18} + X))])$ where

- N = Iteration count that ticks every PMIC clock. The average value of PMIC clock here is $\sim 2.25 \text{ MHz}$. Hence the iteration count ticks every $(1 / 2.25 \text{ MHz}) \sim 0.444 \mu s$.
- X = random fractional value in the range (0, 1) that adds the dither

The PMIC clock frequency is determined by the clock divider which starts with a value of 100, on the 1st PMIC clock period, providing a PMIC clock of 2 MHz, decrementing the divider value by $23303 / 2^{18} = 0.08889$ every PMIC clock period of $1 / 2.25 \text{ MHz} \sim 0.444 \mu s$, finally reaching a value of 80 on 225th PMIC clock period, providing a PMIC clock of 2.5 MHz. Hence, the frequency varies from [2 MHz, 2.5 MHz] over 225 PMIC clock periods or $225 \times 0.444 \mu s$ or $\sim 100 \mu s$.

5.16.10 Sub block 0x4049 – AWR_MSS_PERIODICTESTS_CONF_SB

This sub block is used to trigger the periodic tests in MSS.

Table 5.119 describes the content of this sub block.

Table 5.119: AWR_MSS_PERIODICTESTS_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4049 |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.119 – continued from previous page

| | | |
|----------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PERIODICITY | 4 | Periodicity at which tests need to be run 1 LSB = 1 ms Minimum value is 40 ms Maximum value is 150ms NOTE: MSS Windowed WDT period is set to this periodicity and WDT can not support period more than 150ms. |
| TEST_EN | 4 | 1 – Enable, 0 – Disable Bit Monitoring type b0 PERIODIC_CONFIG_REGISTER_READ_EN b1 ESM_MONITORING_EN b31:2 RESERVED |
| REPORTING_MODE | 1 | Controls when the AWR device sends the report corresponding to the periodic tests to the host. A report generically refers to both success/failure status flags. Value Definition 0 Report is sent every monitoring period 1 Report is sent only on a failure |
| RESERVED | 3 | 0x000000 |

NOTE: The MSS periodic monitor test run and latent tests are not recommended to run in parallel as latent tests are destructive tests which would cause periodic tests to fail.

5.16.11 Sub block 0x404A – AWR_MSS_LATENTFAULT_TEST_CONF_SB

This sub block is used to trigger the periodic latent fault tests in MSS, this API should not be issued when functional frames are running, these are destructive tests.

Table 5.120 describes the content of this sub block.

Table 5.120: AWR_MSS_LATENTFAULT_TEST_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x404A |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.120 – continued from previous page

| TEST_EN.1 | 4 | Bits | Definition |
|-----------|---|------|----------------------------------------------------------------------|
| | | b0 | MibSPI self-test |
| | | b1 | DMA self-test |
| | | b2 | RESERVED |
| | | b3 | RTI self-test |
| | | b4 | ESM self-test |
| | | b5 | EDMA self-test |
| | | b6 | CRC self-test |
| | | b7 | VIM self-test |
| | | b8 | RESERVED |
| | | b9 | Mailbox self-test |
| | | b10 | LVDS pattern generation test |
| | | b11 | CSI2 pattern generation test |
| | | b12 | Generating NERROR |
| | | b13 | MibSPI single bit error test |
| | | b14 | MibSPI double bit error test |
| | | b15 | DMA Parity error |
| | | b16 | TCMA RAM single bit errors (Not supported refer latest release note) |
| | | b17 | TCMB RAM single bit errors (Not supported refer latest release note) |
| | | b18 | TCMA RAM double bit errors (Not supported refer latest release note) |
| | | b19 | TCMB RAM double bit errors (Not supported refer latest release note) |
| | | b20 | TCMA RAM parity errors (Not supported refer latest release note) |
| | | b21 | TCMB RAM parity errors (Not supported refer latest release note) |
| | | b22 | RESERVED |
| | | b23 | RESERVED |
| | | b24 | DMA MPU Region tests |
| | | b25 | MSS Mailbox single bit errors |
| | | b26 | MSS Mailbox double bit errors |
| | | b27 | BSS Mailbox single bit errors |
| | | b28 | BSS Mailbox double bit errors |
| | | b29 | EDMA MPU test |
| | | b30 | EDMA parity test |
| | | b31 | CSI2 parity test |

Continued on next page

Table 5.120 – continued from previous page

| | | | |
|----------------|---|--------|--------------------------------------------------------------------------------------------------------------------|
| TEST_EN_2 | 4 | Bits | Definition |
| | | b0 | DCC self-test |
| | | b1 | DCC fault insertion test |
| | | b2 | RESERVED |
| | | b3 | VIM RAM parity test |
| | | b4 | SCI boot time test |
| | | b31:5 | RESERVED |
| REPORTING_MODE | 1 | Value | Definition |
| | | 0 | Report is sent after test completion |
| | | 1 | Report is send only upon a failure |
| TEST_MODE | 1 | Value | Definition |
| | | 0 | Production mode. Latent faults are tested and any failures are reported |
| | | 1 | Characterization mode. Faults are injected and failures are reported which allows testing of the failure reporting |
| RESERVED | 2 | 0x0000 | |

NOTE: The MSS latent self tests are destructive tests, which would cause corruption in ongoing SPI/mailbox transactions and may generate N-Error signals while performing ESM G2 error checks. It is recommended not to run these self tests in functional mode of operation.

5.16.12 Sub block 0x404B – AWR_DEV_TESTPATTERN_GEN_SET_SB

This sub block contains the configurations to setup the test pattern to be generated and transferred over the selected high speed interface (LVDS/CSI2). This command has to be issued after the data path configurations commands are issued. This can be used to perform a sanity test of the high speed interface connectivity and correct reception.

Table 5.121 describes the contents of this sub block.

Table 5.121: AWR_DEV_TESTPATTERN_GEN_SET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x404B |
| SBLKLEN | 2 | Value = 48 |

Continued on next page

Table 5.121 – continued from previous page

| TESTPATTERN_ GEN_CTRL | 1 | This field controls the enable-disable of the generation of the test pattern. Value Description 0x0 Disable test pattern generation 0x1 Enable test pattern generation | | | | | | | | | | | | | | | |
|----------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------|---------------------|---|---------|------|---|------|------|---|---------|------|---|------|------|
| TESTPATTERN_ GEN_TIMING | 1 | Number of system clocks (200 MHz) between successive samples for the test pattern gen. Applicable only in case of Test pattern enable. Else ignored. | | | | | | | | | | | | | | | |
| TESTPATTERN_ PKT_SIZE | 2 | Number of ADC samples to capture for each RX Valid range: 64 to MAX_NUM_SAMPLES, Where MAX_NUM_SAMPLES is such that all the enabled RX channels' data fits into 16 kB memory, with each sample consuming 2 bytes for real ADC output case and 4 bytes for complex 1x and complex 2x ADC output cases. For example in AWR1243/xWR1443 when the ADC buffer size is 16 kB <table border="1"> <thead> <tr> <th>Number of RX chains</th> <th>ADC format</th> <th>MAX_NUM_ SAMPLES</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>Complex</td> <td>1024</td> </tr> <tr> <td>4</td> <td>Real</td> <td>2048</td> </tr> <tr> <td>2</td> <td>Complex</td> <td>2048</td> </tr> <tr> <td>2</td> <td>Real</td> <td>4096</td> </tr> </tbody> </table> | Number of RX chains | ADC format | MAX_NUM_ SAMPLES | 4 | Complex | 1024 | 4 | Real | 2048 | 2 | Complex | 2048 | 2 | Real | 4096 |
| Number of RX chains | ADC format | MAX_NUM_ SAMPLES | | | | | | | | | | | | | | | |
| 4 | Complex | 1024 | | | | | | | | | | | | | | | |
| 4 | Real | 2048 | | | | | | | | | | | | | | | |
| 2 | Complex | 2048 | | | | | | | | | | | | | | | |
| 2 | Real | 4096 | | | | | | | | | | | | | | | |
| NUM_TESTPAT- TERN_PKTS | 4 | Number of test pattern packets to send For infinite packets set it to 0 | | | | | | | | | | | | | | | |
| TESTPATTERN_ RX0_ICFG | 4 | This field specifies the values for Rx0, I channel. Applicable only in case of test pattern enable. Else ignored. Bits Description b15:0 Start offset value to be used for the first sample for the test pattern data b31:16 Value to be added for each successive sample for the test pattern data | | | | | | | | | | | | | | | |
| TESTPATTERN_ RX0_QCFG | 4 | This field specifies the values for Rx0, Q channel. Applicable only in case of test pattern enable. Else ignored. Bits Description b15:0 Start offset value to be used for the first sample for the test pattern data b31:16 Value to be added for each successive sample for the test pattern data | | | | | | | | | | | | | | | |

Continued on next page

Table 5.121 – continued from previous page

| | | |
|--------------------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TESTPATTERN_ RX1_ICFG | 4 | <p>This field specifies the values for Rx1, I channel. Applicable only in case of test pattern enable. Else ignored.</p> <p>Bits Description</p> <p>b15:0 Start offset value to be used for the first sample for the test pattern data</p> <p>b31:16 Value to be added for each successive sample for the test pattern data</p> |
| TESTPATTERN_ RX1_QCFG | 4 | <p>This field specifies the values for Rx1, Q channel. Applicable only in case of test pattern enable. Else ignored.</p> <p>Bits Description</p> <p>b15:0 Start offset value to be used for the first sample for the test pattern data</p> <p>b31:16 Value to be added for each successive sample for the test pattern data</p> |
| TESTPATTERN_ RX2_ICFG | 4 | <p>This field specifies the values for Rx2, I channel. Applicable only in case of test pattern enable. Else ignored.</p> <p>Bits Description</p> <p>b15:0 Start offset value to be used for the first sample for the test pattern data</p> <p>b31:16 Value to be added for each successive sample for the test pattern data</p> |
| TESTPATTERN_ RX2_QCFG | 4 | <p>This field specifies the values for Rx2, Q channel. Applicable only in case of test pattern enable. Else ignored.</p> <p>Bits Description</p> <p>b15:0 Start offset value to be used for the first sample for the test pattern data</p> <p>b31:16 Value to be added for each successive sample for the test pattern data</p> |
| TESTPATTERN_ RX3_ICFG | 4 | <p>This field specifies the values for Rx3, I channel. Applicable only in case of test pattern enable. Else ignored.</p> <p>Bits Description</p> <p>b15:0 Start offset value to be used for the first sample for the test pattern data</p> <p>b31:16 Value to be added for each successive sample for the test pattern data</p> |

Continued on next page

Table 5.121 – continued from previous page

| | | |
|--------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TESTPATTERN_ RX3_QCFG | 4 | This field specifies the values for Rx3, Q channel. Applicable only in case of test pattern enable. Else ignored. Bits Description b15:0 Start offset value to be used for the first sample for the test pattern data b31:16 Value to be added for each successive sample for the test pattern data |
| RESERVED | 4 | 0x00000000 |

NOTE: This test pattern can be used only in LVDS testing and bring-up

5.16.13 Sub block 0x404C – AWR_DEV_CONFIGURATION_SET_SB

This API is used to configure the CRC type for the async events from MSS. The default is 16 bit CRC if this API is not issued. The first async event after MSS powerup will have a 16 bit CRC.

Table 5.122: AWR_DEV_CONFIGURATION_SET_SB contents

| Field Name | Number of bytes | Description |
|-------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x404C |
| SBLKLEN | 2 | Value = 16 |
| ASYNC_EVENT_ CRC_CFG | 1 | Value Description 0 16 bit CRC for MSS async events 1 32 bit CRC for MSS async events 2 64 bit CRC for MSS async events |
| RESERVED1 | 1 | 0x00 |
| RESERVED2 | 2 | 0x0000 |
| RESERVED3 | 4 | 0x00000000 |
| RESERVED4 | 4 | 0x00000000 |

5.17 Sub blocks related to AWR_DEV_CONF_GET_MSG

5.17.1 Sub block 0x4060 – AWR_DEV_MCUCLOCK_GET_SB

This API is used to read the MCU clock configuration. Response packet structure will be same as AWR_DEV_MCUCLOCK_SET_SB

Table 5.123: AWR_DEV_MCUCLOCK_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4060 |
| SBLKLEN | 2 | Value = 4 |

5.17.2 Sub block 0x4061 – AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB

This API is used to read the RX data format configuration. Response packet structure will be same as AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB

Table 5.124: AWR_DEV_RX_DATA_FORMAT_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4061 |
| SBLKLEN | 2 | Value = 4 |

5.17.3 Sub block 0x4062 – AWR_DEV_RX_DATA_PATH_CONF_GET_SB

This API is used to read the RX data path configuration. Response packet structure will be same as AWR_DEV_RX_DATA_PATH_CONF_SET_SB

Table 5.125: AWR_DEV_RX_DATA_PATH_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4062 |
| SBLKLEN | 2 | Value = 4 |

5.17.4 Sub block 0x4063 – AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB

This API is used to read the RX data path lane enable configuration. Response packet structure will be same as AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB

Table 5.126: AWR_DEV_RX_DATA_PATH_LANEEN_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4063 |
| SBLKLEN | 2 | Value = 4 |

5.17.5 Sub block 0x4064 – AWR_DEV_RX_DATA_PATH_CLK_GET_SB

This API is used to read the RX data path clock configuration. Response packet structure will be same as AWR_DEV_RX_DATA_PATH_CLK_SET_SB

Table 5.127: AWR_DEV_RX_DATA_PATH_CLK_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4064 |
| SBLKLEN | 2 | Value = 4 |

5.17.6 Sub block 0x4065 – AWR_DEV_LVDS_CFG_GET_SB

This API is used to read the LVDS configuration. Response packet structure will be same as AWR_DEV_LVDS_CFG_SET_SB

Table 5.128: AWR_DEV_LVDS_CFG_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4065 |
| SBLKLEN | 2 | Value = 4 |

5.17.7 Sub block 0x4066 – AWR_DEV_RX_CONTSTREAMING_MODE_CONF_GET_SB

This API is used to read the continuous streaming mode configuration. Response packet structure will be same as AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB

Table 5.129: AWR_DEV_RX_CONTSTREAMING_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4066 |
| SBLKLEN | 2 | Value = 4 |

5.17.8 Sub block 0x4067 – AWR_DEV_CSI2_CFG_GET_SB

This API is used to read the CSI2 configuration. Response packet structure will be same as AWR_DEV_CSI2_CFG_SET_SB

Table 5.130: AWR_DEV_CSI2_CFG_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4067 |
| SBLKLEN | 2 | Value = 4 |

5.17.9 Sub block 0x4068 – AWR_DEV_PMICCLOCK_CONF_GET_SB

This API is used to read the PMIC clock configuration. Response packet structure will be same as AWR_DEV_PMICCLOCK_CONF_SET_SB

Table 5.131: AWR_DEV_PMICCLOCK_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4068 |
| SBLKLEN | 2 | Value = 4 |

5.17.10 Sub block 0x4069 – AWR_MSS_LATENTFAULT_TEST_CONF_GET_SB

This API is used to read the MSS latent fault test configuration. Response packet structure will be same as AWR_MSS_LATENTFAULT_TEST_CONF_SET_SB

Table 5.132: AWR_MSS_LATENTFAULT_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x4069 |
| SBLKLEN | 2 | Value = 4 |

5.17.11 Sub block 0x406A – AWR_MSS_PERIODICTESTS_CONF_GET_SB

This API is used to read the MSS periodic tests configuration. Response packet structure will be same as AWR_MSS_PERIODICTESTS_CONF_SET_SB

Table 5.133: AWR_MSS_PERIODICTESTS_CONF_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|-------------|
|------------|-----------------|-------------|

| | | |
|---------|---|----------------|
| SBLKID | 2 | Value = 0x406A |
| SBLKLEN | 2 | Value = 4 |

5.17.12 Sub block 0x406B – AWR_DEV_TESTPATTERN_GEN_GET_SB

This API is used to read the test pattern generation configuration. Response packet structure will be same as AWR_DEV_TESTPATTERN_GEN_SET_SB

Table 5.134: AWR_DEV_TESTPATTERN_GEN_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x406B |
| SBLKLEN | 2 | Value = 4 |

5.18 Sub blocks related to AWR_DEV_FILE_DOWNLOAD_MSG

5.18.1 Sub block 0x4080 – AWR_DEV_FILE_DOWNLOAD_SB

This sub block is used to send the file in chunks/parts for download into RAM. Table 5.135 describes the content of this sub block.

Table 5.135: AWR_DEV_FILE_DOWNLOAD_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | | | | | | | |
|--------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------|-----|----------|-----|----------|-----|----------|-----|----------|-----|-------------|-----|-------------|-----|-------------|-----|-------------|
| SBLKID | 2 | Value = 0x4080 | | | | | | | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = Variable | | | | | | | | | | | | | | | | | | |
| FILE_TYPE | 4 | <table border="0"> <tr> <td>Value</td> <td>Description</td> </tr> <tr> <td>0x0</td> <td>RESERVED</td> </tr> <tr> <td>0x1</td> <td>RESERVED</td> </tr> <tr> <td>0x2</td> <td>RESERVED</td> </tr> <tr> <td>0x3</td> <td>RESERVED</td> </tr> <tr> <td>0x4</td> <td>META.IMAGE1</td> </tr> <tr> <td>0x5</td> <td>META.IMAGE2</td> </tr> <tr> <td>0x6</td> <td>META.IMAGE3</td> </tr> <tr> <td>0x7</td> <td>META.IMAGE4</td> </tr> </table> | Value | Description | 0x0 | RESERVED | 0x1 | RESERVED | 0x2 | RESERVED | 0x3 | RESERVED | 0x4 | META.IMAGE1 | 0x5 | META.IMAGE2 | 0x6 | META.IMAGE3 | 0x7 | META.IMAGE4 |
| Value | Description | | | | | | | | | | | | | | | | | | | |
| 0x0 | RESERVED | | | | | | | | | | | | | | | | | | | |
| 0x1 | RESERVED | | | | | | | | | | | | | | | | | | | |
| 0x2 | RESERVED | | | | | | | | | | | | | | | | | | | |
| 0x3 | RESERVED | | | | | | | | | | | | | | | | | | | |
| 0x4 | META.IMAGE1 | | | | | | | | | | | | | | | | | | | |
| 0x5 | META.IMAGE2 | | | | | | | | | | | | | | | | | | | |
| 0x6 | META.IMAGE3 | | | | | | | | | | | | | | | | | | | |
| 0x7 | META.IMAGE4 | | | | | | | | | | | | | | | | | | | |
| FILE_LENGTH | 4 | Length of File | | | | | | | | | | | | | | | | | | |
| FILE_CONTENT | Variable | Content of File, may split into multiple chunks. | | | | | | | | | | | | | | | | | | |

NOTE: In the first chunk of file, FILE_TYPE and FILE_LENGTH is available and then first chunk onward these two fields will not be part of SB content

5.19 Sub blocks related to AWR_DEV_FRAME_CONFIG_APPLY_MSG

5.19.1 Sub block 0x40C0 – AWR_DEV_FRAME_CONFIG_APPLY_SB

This sub block is used to indicate to MSS to apply all the device configurations in the hardware. This API should be used when lagacy frame config is used.

Table 5.136 describes the content of this sub block.

Table 5.136: AWR_DEV_FRAME_CONFIG_APPLY_SB contents

| Field Name | Number of bytes | Description |
|----------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x40C0 |
| SBLKLEN | 2 | Value = 12 |
| NUM_CHIRPS | 4 | Number of chirps per frame |
| HALF_WORDS_PER_CHIRP | 2 | Number of half words in ADC buffer per chirp Example 1: In real mode, if number of ADC samples per chirp is 256 then this value will be 256 Example 2: In complex1x or complex2x modes, if number of ADC samples per chirp is 256 then this value will be 512 |
| RESERVED | 2 | 0x0000 |

5.19.2 Sub block 0x40C1 – AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB

This sub block is used to indicate to MSS to apply all the advanced frame configuration settings in the hardware. This API should be used when advance frame config is used.

Table 5.137 describes the content of this sub block.

Table 5.137: AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|-------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x40C1 |
| SBLKLEN | 2 | Value = 40 |
| NUM_SUB-FRAMES | 1 | Number of sub frames enabled in this frame Valid range: 1 to 4 |
| RESERVED | 3 | 0x00 |

Continued on next page

Table 5.137 – continued from previous page

| | | |
|----------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SF1_TOT_NUM_CHIRPS | 4 | Number of chirps in sub frame 1 |
| SF1_NUM_ADC_SAMPLES_PER_DATA_PKT | 2 | <p>Number of half words (16 bits) of ADC samples per data packet in sub-frame 1</p> <p>Example 1: In real mode, if number of ADC samples per chirp in subframe1 is 256 then this value will be 256</p> <p>Example 2: In complex1x or complex2x modes, if number of ADC samples per chirp in subframe1 is 256 then this value will be 512</p> <p>In AWR1243/xWR1443: Program this as the same as number of ADC samples in each chirp of this sub frame (required to be the same)</p> <p>Exception: Can do #chirps based ping-pong as in xWR1642 (see below), if CP/CQ are not needed. Useful for chirp stitching use case.</p> <p>In xWR1642/xWR1843 (For reference Only): The ADC samples corresponding to one or more chirps can be grouped and sent to the DSP as a single packet. Program this as the number of half words of ADC samples per packet. Ensure that in one sub frame, there is integer number of such packets.</p> <p>Maximum size of a data packet: (16384 - 1) half words.</p> |
| SF1_PROC_NUM_CHIRPS_PER_DATA_PKT | 1 | <p>Number of chirps per data packet to process at a time in sub-frame 1.</p> <p>In AWR1243/xWR1443: Program this as 1.</p> <p>Exception: Can be > 1 as in 1642 if CP/CQ is not needed. Useful for chirp stitching use case.</p> <p>In xWR1642/xWR1843 (For reference Only): The ADC samples corresponding to one or more chirps can be grouped and sent to the DSP as a single packet. Program this as the corresponding number of chirps per packet.</p> <p>Maximum value = 8.</p> <p>Note on maximum size: 8 chirps for CP and BPM.</p> |
| RESERVED | 1 | 0x00 |
| SF2_TOT_NUM_CHIRPS | 4 | Number of chirps in sub-frame 2 |
| SF2_NUM_ADC_SAMPLES_PER_DATA_PKT | 2 | <p>Number of ADC Samples per data packet in sub-frame 2</p> <p>Same conditions apply as in sub-frame 1.</p> |
| SF2_PROC_NUM_CHIRPS_PER_DATA_PKT | 1 | <p>Number of chirps per data packet to process at a time in sub-frame 2</p> <p>Same conditions apply as in sub-frame 1.</p> |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 5.137 – continued from previous page

| | | |
|----------------------------------|---|------------------------------------------------------------------------------------------------------------------|
| SF3_TOT_NUM_CHIRPS | 4 | Number of chirps in sub-frame3 |
| SF3_NUM_ADC_SAMPLES_PER_DATA_PKT | 2 | Number of ADC samples per data packet in sub-frame 3 Same conditions apply as in sub-frame 1. |
| SF3_PROC_NUM_CHIRPS_PER_DATA_PKT | 1 | Number of chirps per data packet to process at a time in sub-frame 3 Same conditions apply as in sub-frame 1. |
| RESERVED | 1 | 0x00 |
| SF4_TOT_NUM_CHIRPS | 4 | Number of chirps in sub-frame4 |
| SF4_NUM_ADC_SAMPLES_PER_DATA_PKT | 2 | Number of ADC samples per data packet in sub-frame 4 Same conditions apply as in sub-frame 1. |
| SF4_PROC_NUM_CHIRPS_PER_DATA_PKT | 1 | Number of chirps per data packet to process at a time in sub-frame 4 Same conditions apply as in sub-frame 1. |
| RESERVED | 1 | 0x00 |

5.20 Sub blocks related to AWR_DEV_STATUS_GET_MSG

5.20.1 Sub block 0x40E0 – AWR_MSSVERSION_GET_SB

This sub block reads MSS FW version. The information returned by the device will be in the format as given in AWR_MSSVERSION_SB.

Table 5.138 describes the contents of the request sub block

Table 5.138: AWR_MSSVERSION_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x40E0 |
| SBLKLEN | 2 | Value = 4 |

Response to AWR_MSSVERSION_GET_SB

AWR_MSSVERSION_SB sub block is sent by the radar device in response to AWR_MSSVERSION_GET_SB. Note that SBLKID for both AWR_MSSVERSION_GET_SB and AWR_MSSVERSION_SB are same.

Table 5.139 describes the contents of the response sub block.

Table 5.139: AWR_MSSVERSION_SB contents

| Field Name | Number of bytes | Description |
|----------------------------|-----------------|-----------------------------------|
| SBLKID | 2 | Value = 0x40E0 |
| SBLKLEN | 2 | Value = 20 |
| HW_VARIANT | 1 | HW variant number |
| HW_VERSION_MAJOR | 1 | HW version major number |
| HW_VERSION_MINOR | 1 | HW version minor number |
| MSS_FW_VERSION_MAJOR | 1 | MSS FW version major number |
| MSS_FW_VERSION_MINOR | 1 | MSS FW version minor number |
| MSS_FW_VERSION_BUILD | 1 | MSS FW version build number |
| MSS_FW_VERSION_DEBUG | 1 | MSS FW version debug number |
| MSS_FW_VERSION_YEAR | 1 | Year of MSS FW version release |
| MSS_FW_VERSION_MONTH | 1 | Month of MSS FW version release |
| MSS_FW_VERSION_DAY | 1 | Day of MSS FW version release |
| MSS_FW_VERSION_PATCH_MAJOR | 1 | MSS FW version patch major number |
| MSS_FW_VERSION_PATCH_MINOR | 1 | MSS FW version patch minor number |
| MSS_FW_VERSION_PATCH_YEAR | 1 | Year of MSS FW patch release |
| MSS_FW_VERSION_PATCH_MONTH | 1 | Month of MSS FW patch release |
| MSS_FW_VERSION_PATCH_DAY | 1 | Day of MSS FW patch release |

Continued on next page

Table 5.139 – continued from previous page

| | | | |
|----------------------------------|---|------|----------------------|
| MSS_FW_PATCH_BUILD_DEBUG_VERSION | 1 | Bit | Definition |
| | | b3:0 | DEBUG version number |
| | | b7:4 | BUILD version number |

5.20.2 Sub block 0x40E1 – AWR_MSSCPUFAULT_STATUS_GET_SB

This sub block provides the MSS CPU fault information.

Table 5.140 describes the content of this sub block.

Table 5.140: AWR_MSSVERSION_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x40E1 |
| SBLKLEN | 2 | Value = 4 |

Response to AWR_MSSCPUFAULT_STATUS_GET_SB

AWR_MSSCPUFAULT_STATUS_SB is sent in response to AWR_MSSCPUFAULT_STATUS_GET_SB.

Table 5.141 describes the content of AWR_MSSCPUFAULT_STATUS_SB

Table 5.141: AWR_MSSCPUFAULT_STATUS_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|-------------------------------------------|---|-------------------------------------------|---|---------------------------------|---|------------------------------------|----------|----------|
| SBLKID | 2 | Value = 0x40E1 | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 36 | | | | | | | | | | | | |
| FAULT_TYPE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>MSS Processor Undefined Instruction Abort</td> </tr> <tr> <td>1</td> <td>MSS Processor Instruction pre-fetch Abort</td> </tr> <tr> <td>2</td> <td>MSS Processor Data Access Abort</td> </tr> <tr> <td>3</td> <td>MSS Processor Firmware Fatal Error</td> </tr> <tr> <td>0x4-0xFF</td> <td>Reserved</td> </tr> </table> | Value | Definition | 0 | MSS Processor Undefined Instruction Abort | 1 | MSS Processor Instruction pre-fetch Abort | 2 | MSS Processor Data Access Abort | 3 | MSS Processor Firmware Fatal Error | 0x4-0xFF | Reserved |
| Value | Definition | | | | | | | | | | | | | |
| 0 | MSS Processor Undefined Instruction Abort | | | | | | | | | | | | | |
| 1 | MSS Processor Instruction pre-fetch Abort | | | | | | | | | | | | | |
| 2 | MSS Processor Data Access Abort | | | | | | | | | | | | | |
| 3 | MSS Processor Firmware Fatal Error | | | | | | | | | | | | | |
| 0x4-0xFF | Reserved | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | |
| LINE_NUM | 2 | Valid only in case of FAULT type is 0x3, provides the firmware line number at which fatal error occurred. | | | | | | | | | | | | |
| FAULT_LR | 4 | The instruction PC address at which Fault occurred | | | | | | | | | | | | |

Continued on next page

Table 5.141 – continued from previous page

| | | |
|----------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FAULT_PREV_LR | 4 | The return address of the function from which fault function has been called (Call stack LR) |
| FAULT_SPSR | 4 | The CPSR register value at which fault occurred |
| FAULT_SP | 4 | The SP register value at which fault occurred |
| FAULT_CAUSE_ADDRESS | 4 | The address access at which Fault occurred (valid only for fault type 0x0 to 0x2) |
| FAULT_ERROR_STATUS | 2 | The status of Error (Error Cause type - valid only for fault type 0x0 to 0x2) 0x000 BACKGROUND_ERR 0x001 ALIGNMENT_ERR 0x002 DEBUG_EVENT 0x00D PERMISSION_ERR 0x008 SYNCH_EXTER_ERR 0x406 ASYNCH_EXTER_ERR 0x409 SYNCH_ECC_ERR 0x408 ASYNCH_ECC_ERR |
| FAULT_ERROR_SOURCE | 1 | The Source of the Error (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 ERR_SOURCE_AXI_MASTER 0x1 ERR_SOURCE_ATCM 0x2 ERR_SOURCE_BTCM |
| FAULT_AXI_ERROR_TYPE | 1 | The AXI Error type (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 AXI_DECOD_ERR 0x1 AXI_SLAVE_ERR |
| FAULT_ACCESS_TYPE | 1 | The Error Access type (Error Access type - valid only for fault type 0x0 to 0x2) 0x0 READ_ERR 0x1 WRITE_ERR |
| FAULT_RECOVERY_TYPE | 1 | The Error Recovery type (Error Recovery type - Valid only for fault type 0x0 to 0x2) 0x0 UNRECOVERY 0x1 RECOVERY |
| RESERVED | 2 | 0x0000 |

5.20.3 Sub block 0x40E2 – AWR_MSESFAULT_STATUS_GET_SB

This sub block provides the information regarding additional Master sub system faults. Table 5.142 describes the content of this sub block.

Table 5.142: AWR_MSSESMFAULT_STATUS_GET_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x40E2 |
| SBLKLEN | 2 | Value = 4 |

The Response to above request is given in the AWR_MSSESMFAULT_STATUS_SB.
 Table 5.143 describes the contents of AWR_MSSESMFAULT_STATUS_SB.

Table 5.143: AWR_MSSESMFAULT_STATUS_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x40E2 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.143 – continued from previous page

| ESM.GROUP1_ ERRORS | 4 | Bits | Definition |
|--------------------|---|------|----------------------------------------------------------------|
| | | b0 | NERROR in sync |
| | | b1 | RESERVED |
| | | b2 | DMA MPU Region tests |
| | | b3 | DMA Parity error |
| | | b4 | RESERVED |
| | | b5 | RESERVED |
| | | b6 | Access error interrupt from FFT ACC |
| | | b7 | CSI TX FIFO Parity Err |
| | | b8 | TPCC parity error |
| | | b9 | CBUF ECC single bit error |
| | | b10 | RESERVED |
| | | b11 | RESERVED |
| | | b12 | RESERVED |
| | | b13 | Error response from the Peripheral when a DMA transfer is done |
| | | b14 | RESERVED |
| | | b15 | VIM RAM double bit errors |
| | | b16 | RESERVED |
| | | b17 | MibSPI double bit error test |
| | | b18 | ECC error on CBUFF |
| | | b19 | RESERVED |
| | | b20 | VIM RAM single bit errors |
| | | b21 | RESERVED |
| | | b22 | FRC Lock Step Error |
| | | b23 | RESERVED |
| | | b24 | RESERVED |
| | | b25 | MibSPI single bit error test |
| | | b26 | TCMB RAM single bit errors |
| | | b27 | STC error |
| | | b28 | TCMB RAM single bit errors |
| | | b29 | TPTC0 read to protected memory |
| | | b30 | DCC compare error |
| | | b31 | CR4F self-test error.(test of error path by error forcing) |

Continued on next page

Table 5.143 – continued from previous page

| | | | |
|-------------------|----------|------------|---------------------------------------------|
| ESM_GROUP2_ERRORS | 4 | Bits | Definition |
| | | b0 | TCMA RAM single bit errors |
| | | b1 | RESERVED |
| | | b2 | RESERVED |
| | | b3 | TPTC0 write to protected memory |
| | | b4 | RESERVED |
| | | b5 | RESERVED |
| | | b6 | Access error interrupt from FFT ACC |
| | | b7 | VIM Self-Test Error |
| | | b8 | RESERVED |
| | | b9 | RESERVED |
| | | b10 | RESERVED |
| | | b11 | RESERVED |
| | | b12 | QSPI not able to perform the Write to FLASH |
| | | b13 | RESERVED |
| | | b14 | RESERVED |
| | | b15 | RESERVED |
| | | b16 | RESERVED |
| | | b17 | RESERVED |
| | | b18 | RESERVED |
| | | b19 | RESERVED |
| | | b20 | RESERVED |
| | | b21 | RESERVED |
| | | b22 | RESERVED |
| | | b23 | RESERVED |
| | | b24 | RESERVED |
| | | b25 | RESERVED |
| | | b26 | BSS Mailbox single bit errors |
| | | b27 | BSS Mailbox double bit errors |
| | | b28 | MSS Mailbox single bit errors |
| | | b29 | MSS Mailbox double bit errors |
| | | b30 | RESERVED |
| b31 | RESERVED | | |
| RESERVED | 4 | 0x00000000 | |
| RESERVED | 4 | 0x00000000 | |

5.21 Sub blocks related to AWR_DEV_ASYNC_EVENT_MSG

5.21.1 Sub block 0x5000 – AWR_AE_DEV_MSSPOWERUPDONE_SB

This sub block indicates that Master SS power up is now complete. It also indicates the status of boot up tests done by Master SS. This async event is sent when host IRQ is enabled.

Table 5.144 describes the contents of this sub block

Table 5.144: AWR_AE_DEV_MSSPOWERUPDONE_SB contents

| Field Name | Number of bytes | Description |
|--------------------|-----------------|-----------------------------------------|
| SBLKID | 2 | Value = 0x5000 |
| SBLKLEN | 2 | Value = 24 |
| MSS.POWERUP_TIME | 4 | Master SS power up time 1 LSB = 5 ns |
| MSS.POWERUP_STATUS | 8 | Refer to Table 6.3 for bit map details |

Continued on next page

Table 5.144 – continued from previous page

| | | |
|------------------|---|---------------------------------------------------------------|
| BOOTTEST_ STATUS | 8 | 0 – PASS, 1 – FAIL |
| | | Bit Definition |
| | | b0 MibSPI self-test |
| | | b1 DMA self-test |
| | | b2 Watchdog (WDT) self-test |
| | | b3 RTI self-test |
| | | b4 ESM self-test |
| | | b5 EDMA self-test |
| | | b6 CRC self-test |
| | | b7 VIM self-test |
| | | b8 MPU self-test |
| | | b9 Mailbox self-test |
| | | b10 LVDS pattern generation test |
| | | b11 CSI2 pattern generation test |
| | | b12 NERROR generation test |
| | | b13 MibSPI single bit error test |
| | | b14 MibSPI double bit error test |
| | | b15 DMA Parity error test |
| | | b16 TCMA RAM single bit error test |
| | | b17 TCMB RAM single bit error test |
| | | b18 TCMA RAM double bit error test |
| | | b19 TCMB RAM double bit error test |
| | | b20 TCMA RAM parity error test |
| | | b21 TCMB RAM parity error test |
| | | b22 VIM lockstep test |
| | | b23 CCM R4 lockstep test |
| | | b24 DMA MPU region test |
| | | b25 MSS Mailbox single bit error test |
| | | b26 MSS Mailbox double bit error test |
| | | b27 BSS Mailbox single bit error test |
| | | b28 BSS Mailbox double bit error test |
| | | b29 EDMA MPU test |
| | | b30 EDMA parity test |
| | | b31 CSI2 parity test |
| | | b32 PBIST (VIM RAM/TCM RAM/MibSPI SRAM/Mailbox/EDMA/DMA/CSI2) |
| | | b33 LBIST (VIM/CR4) |
| | | b34 PCR test |
| | | b63:35 RESERVED |

Continued on next page

Table 5.144 – continued from previous page

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------|
| NOTE: | The functional APIs shall be sent to radar device only after receiving AWR_AE_DEV_MSSPOWERUPDONE_SB Async-event after power cycle. |
|--------------|------------------------------------------------------------------------------------------------------------------------------------|

5.21.2 Sub block 0x5001 – AWR_AE_DEV_RFPOWERUPDONE_SB

This sub block indicates that BIST SS power up is now complete.

Table 5.145 describes the contents of this sub block

Table 5.145: AWR_AE_DEV_RFPOWERUPDONE_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x5001 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.145 – continued from previous page

| | | |
|---------------------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BSS_POWERUP_ BIST_STATUS_ FLAGS | 4 | 1 – PASS, 0 – FAIL Bit Status Information b0 ROM CRC check b1 CR4 and VIM lockstep test b2 RESERVED b3 VIM test (Not supported in 1st Gen devices, refer latest release note) b4 STC test of diagnostic b5 CR4 STC b6 CRC test b7 RAMPGEN memory ECC test b8 DFE Parity test b9 DFE memory ECC b10 RAMPGEN lockstep test b11 FRC lockstep test b12 DFE memory PBIST b13 RAMPGEN memory PBIST b14 PBIST test b15 WDT test b16 ESM test b17 DFE STC b18 RESERVED b19 ATCM, BTCM ECC test b20 ATCM, BTCM parity test b21 RESERVED b22 RESERVED b23 RESERVED b24 FFT test b25 RTI test b26 PCR test b31:27 RESERVED |
| POWERUP_TIME | 4 | RF BIST SS Power up time 1 LSB = 5 ns |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

5.21.3 Sub block 0x5002 – AWR_AE_MSS_CPUFAULT_SB

This sub block indicates CPU fault status of Master SS.

Table 5.146 describes the content of this sub block.

Table 5.146: AWR_AE_MSS_CPUFAULT_STATUS_SB contents

| Field Name | Number of bytes | Description |
|---------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x5002 |
| SBLKLEN | 2 | Value = 36 |
| FAULT_TYPE | 1 | 0 MSS Processor Undefined Instruction Abort 1 MSS Processor Instruction pre-fetch Abort 2 MSS Processor Data Access Abort 3 MSS Processor Firmware Fatal Error 0x4-0xFF Reserved |
| RESERVED | 1 | 0x00 |
| LINE_NUM | 2 | Valid only in case of FAULT type is 0x3, provides the firmware line number at which fatal error occurred. |
| FAULT_LR | 4 | The instruction PC address at which Fault occurred |
| FAULT_PREV_LR | 4 | The return address of the function from which fault function has been called (Call stack LR) |
| FAULT_SPSR | 4 | The CPSR register value at which fault occurred |
| FAULT_SP | 4 | The SP register value at which fault occurred |
| FAULT_CAUSE_ADDRESS | 4 | The address access at which Fault occurred (valid only for fault type 0x0 to 0x2) |
| FAULT_ERROR_STATUS | 2 | The status of Error (Error Cause type - valid only for fault type 0x0 to 0x2) 0x000 BACKGROUND_ERR 0x001 ALIGNMENT_ERR 0x002 DEBUG_EVENT 0x00D PERMISSION_ERR 0x008 SYNCH_EXTER_ERR 0x406 ASYNCH_EXTER_ERR 0x409 SYNCH_ECC_ERR 0x408 ASYNCH_ECC_ERR |

Continued on next page

Table 5.146 – continued from previous page

| | | |
|----------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FAULT_ERROR_SOURCE | 1 | The Source of the Error (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 ERR_SOURCE_AXI_MASTER 0x1 ERR_SOURCE_ATCM 0x2 ERR_SOURCE_BTCM |
| FAULT_AXI_ERROR_TYPE | 1 | The AXI Error type (Error Source type - valid only for fault type 0x0 to 0x2) 0x0 AXI_DECOD_ERR 0x1 AXI_SLAVE_ERR |
| FAULT_ACCESS_TYPE | 1 | The Error Access type (Error Access type - valid only for fault type 0x0 to 0x2) 0x0 READ_ERR 0x1 WRITE_ERR |
| FAULT_RECOVERY_TYPE | 1 | The Error Recovery type (Error Recovery type - Valid only for fault type 0x0 to 0x2) 0x0 UNRECOVERY 0x1 RECOVERY |
| RESERVED | 2 | 0x0000 |

5.21.4 Sub block 0x5003 – AWR_AE_MSS_ESMFAULT_STATUS_SB

This sub block indicates any other faults inside the MSS.

Table 5.147 describes the content of this sub block.

Table 5.147: AWR_AE_MSS_ESMFAULT_STATUS_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x5003 |
| SBLKLEN | 2 | Value = 20 |

Continued on next page

Table 5.147 – continued from previous page

| ESM.GROUP1_ ERRORS | 4 | Bits | Definition |
|-----------------------|---|------|----------------------------------------------------------------|
| | | b0 | NERROR in sync |
| | | b1 | RESERVED |
| | | b2 | DMA MPU Region tests |
| | | b3 | DMA Parity error |
| | | b4 | RESERVED |
| | | b5 | RESERVED |
| | | b6 | Access error interrupt from FFT ACC |
| | | b7 | CSI TX FIFO Parity Err |
| | | b8 | TPCC parity error |
| | | b9 | CBUF ECC single bit error |
| | | b10 | RESERVED |
| | | b11 | RESERVED |
| | | b12 | RESERVED |
| | | b13 | Error response from the Peripheral when a DMA transfer is done |
| | | b14 | RESERVED |
| | | b15 | VIM RAM double bit errors |
| | | b16 | RESERVED |
| | | b17 | MibSPI double bit error test |
| | | b18 | ECC error on CBUFF |
| | | b19 | RESERVED |
| | | b20 | VIM RAM single bit errors |
| | | b21 | RESERVED |
| | | b22 | FRC Lock Step Error |
| | | b23 | RESERVED |
| | | b24 | RESERVED |
| | | b25 | MibSPI single bit error test |
| | | b26 | TCMB RAM single bit errors |
| | | b27 | STC error |
| | | b28 | TCMB RAM single bit errors |
| | | b29 | TPTC0 read to protected memory |
| | | b30 | DCC compare error |
| | | b31 | CR4F self-test error.(test of error path by error forcing) |

Continued on next page

Table 5.147 – continued from previous page

| | | | |
|-------------------|----------|------------|---------------------------------------------|
| ESM_GROUP2_ERRORS | 4 | Bits | Definition |
| | | b0 | TCMA RAM single bit errors |
| | | b1 | RESERVED |
| | | b2 | RESERVED |
| | | b3 | TPTC0 write to protected memory |
| | | b4 | RESERVED |
| | | b5 | RESERVED |
| | | b6 | Access error interrupt from FFT ACC |
| | | b7 | VIM Self-Test Error |
| | | b8 | RESERVED |
| | | b9 | RESERVED |
| | | b10 | RESERVED |
| | | b11 | RESERVED |
| | | b12 | QSPI not able to perform the Write to FLASH |
| | | b13 | RESERVED |
| | | b14 | RESERVED |
| | | b15 | RESERVED |
| | | b16 | RESERVED |
| | | b17 | RESERVED |
| | | b18 | RESERVED |
| | | b19 | RESERVED |
| | | b20 | RESERVED |
| | | b21 | RESERVED |
| | | b22 | RESERVED |
| | | b23 | RESERVED |
| | | b24 | RESERVED |
| | | b25 | RESERVED |
| | | b26 | BSS Mailbox single bit errors |
| | | b27 | BSS Mailbox double bit errors |
| | | b28 | MSS Mailbox single bit errors |
| | | b29 | MSS Mailbox double bit errors |
| | | b30 | RESERVED |
| b31 | RESERVED | | |
| RESERVED | 4 | 0x00000000 | |
| RESERVED | 4 | 0x00000000 | |

NOTE: The FRC lockstep fatal error is connected to MSS ESM Group 1 lines, This fatal error must be handled in Host in AWR1243 device (MSS or DSS in xWR1642, xWR1843 and IWR6843 devices)

5.21.5 Sub block 0x5004 – RESERVED

5.21.6 Sub block 0x5005 – AWR_AE_MSS_BOOTERRORSTATUS_SB

This sub block indicates error status of MSS when booted over SPI. This async event is sent after the bootup over SPI is complete.

Table 5.148 describes the content of this sub block.

Table 5.148: AWR_AE_MSS_BOOTERRORSTATUS_SB contents

| Field Name | Number of bytes | Description |
|--------------------|-----------------|-----------------------------------------|
| SBLKID | 2 | Value = 0x5005 |
| SBLKLEN | 2 | Value = 24 |
| MSS.POWERUP_TIME | 4 | Master SS power up time 1 LSB = 5 ns |
| MSS.POWERUP_STATUS | 8 | Refer to Table 6.3 for bit map details |

Continued on next page

Table 5.148 – continued from previous page

| | | |
|------------------|---|---------------------------------------------------------------|
| BOOTTEST_ STATUS | 8 | 0 – PASS, 1 – FAIL |
| | | Bit Definition |
| | | b0 MibSPI self-test |
| | | b1 DMA self-test |
| | | b2 Watchdog (WDT) self-test |
| | | b3 RTI self-test |
| | | b4 ESM self-test |
| | | b5 EDMA self-test |
| | | b6 CRC self-test |
| | | b7 VIM self-test |
| | | b8 MPU self-test |
| | | b9 Mailbox self-test |
| | | b10 LVDS pattern generation test |
| | | b11 CSI2 pattern generation test |
| | | b12 NERROR generation test |
| | | b13 MibSPI single bit error test |
| | | b14 MibSPI double bit error test |
| | | b15 DMA Parity error test |
| | | b16 TCMA RAM single bit error test |
| | | b17 TCMB RAM single bit error test |
| | | b18 TCMA RAM double bit error test |
| | | b19 TCMB RAM double bit error test |
| | | b20 TCMA RAM parity error test |
| | | b21 TCMB RAM parity error test |
| | | b22 VIM lockstep test |
| | | b23 CCM R4 lockstep test |
| | | b24 DMA MPU region test |
| | | b25 MSS Mailbox single bit error test |
| | | b26 MSS Mailbox double bit error test |
| | | b27 BSS Mailbox single bit error test |
| | | b28 BSS Mailbox double bit error test |
| | | b29 EDMA MPU test |
| | | b30 EDMA parity test |
| | | b31 CSI2 parity test |
| | | b32 PBIST (VIM RAM/TCM RAM/MibSPI SRAM/Mailbox/EDMA/DMA/CSI2) |
| | | b33 LBIST (VIM/CR4) |
| | | b63:34 RESERVED |

Continued on next page

Table 5.148 – continued from previous page

| | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------|
| NOTE: | The functional APIs shall be sent to radar device only after receiving AWR_AE_MSS_BOOTERRORSTATUS_SB Async-event after power-cycle. |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------|

5.21.7 Sub block 0x5006 – AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB

This sub block indicates the test status report of the latent fault tests.

Table 5.149 describes the content of this sub block.

Table 5.149: AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x5006 |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 5.149 – continued from previous page

| | | |
|--------------|---|-----------------------------------|
| TEST_STATUS_ | 4 | 1 – PASS, 0 - FAIL |
| FLAG1 | | Bits Definition |
| | | b0 MibSPI self-test |
| | | b1 DMA self-test |
| | | b2 Reserved |
| | | b3 RTI self-test |
| | | b4 ESM self-test |
| | | b5 EDMA self-test |
| | | b6 CRC self-test |
| | | b7 VIM self-test |
| | | b8 Reserved |
| | | b9 Mailbox self-test |
| | | b10 LVDS pattern generation test |
| | | b11 CSI2 pattern generation test |
| | | b12 Generating NERROR |
| | | b13 MibSPI single bit error test |
| | | b14 MibSPI double bit error test |
| | | b15 DMA Parity error |
| | | b16 TCMA RAM single bit errors |
| | | b17 TCMB RAM single bit errors |
| | | b18 TCMA RAM double bit errors |
| | | b19 TCMB RAM double bit errors |
| | | b20 TCMA RAM parity errors. |
| | | b21 TCMB RAM parity errors. |
| | | b22 Reserved |
| | | b23 Reserved |
| | | b24 DMA MPU Region tests |
| | | b25 MSS Mailbox single bit errors |
| | | b26 MSS Mailbox double bit errors |
| | | b27 BSS Mailbox single bit errors |
| | | b28 BSS Mailbox double bit errors |
| | | b29 EDMA MPU test |
| | | b30 EDMA parity test |
| | | b31 CSI2 parity test |

Continued on next page

Table 5.149 – continued from previous page

| | | | |
|-------------------|---|------------|--------------------------|
| TEST_STATUS_FLAG2 | 4 | Bits | Definition |
| | | b0 | DCC self-test |
| | | b1 | DCC fault insertion test |
| | | b2 | RESERVED |
| | | b3 | VIM RAM parity test |
| | | b4 | SCI boot time test |
| | | b31:5 | RESERVED |
| RESERVED | 4 | 0x00000000 | |

5.21.8 Sub block 0x5007 – AWR_AE_MSS_PERIODICTEST_STATUS_SB

This sub block indicates test status of the periodic tests.

Table 5.150 describes the content of this sub block.

Table 5.150: AWR_AE_MSS_PERIODICTEST_STATUS_SB contents

| Field Name | Number of bytes | Description |
|------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x5007 |
| SBLKLEN | 2 | Value = 12 |
| TEST_STATUS_FLAG | 4 | 1 – PASS, 0 – FAIL Bits Definition b0 Periodic read back of static registers b1 ESM self-test b31:2 RESERVED |
| RESERVED | 4 | 0x00000000 |

5.21.9 Sub block 0x5008 – AWR_AE_MSS_RFERROR_STATUS_SB

This sub block indicates the RF error status.

Table 5.151 describes the content of this sub block.

Table 5.151: AWR_AE_MSS_RFERROR_STATUS_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x5008 |

Continued on next page

Table 5.151 – continued from previous page

| | | | | | | | | | | | | | | | | | | | | |
|-------------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------|---|---------------|---|--------------|---|----------------------|---|----------------------|-----|----------|---|----------------------------------------------|--------|----------|
| SBLKLEN | 2 | Value = 12 | | | | | | | | | | | | | | | | | | |
| ERROR_STATUS_FLAG | 4 | <table> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>No fault</td> </tr> <tr> <td>1</td> <td>BSS FW assert</td> </tr> <tr> <td>2</td> <td>BSS FW abort</td> </tr> <tr> <td>3</td> <td>BSS ESM GROUP1 ERROR</td> </tr> <tr> <td>4</td> <td>BSS ESM GROUP2 ERROR</td> </tr> <tr> <td>6-5</td> <td>RESERVED</td> </tr> <tr> <td>7</td> <td>BSS monitoring failure in Mode 1(Quiet mode)</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> | Value | Definition | 0 | No fault | 1 | BSS FW assert | 2 | BSS FW abort | 3 | BSS ESM GROUP1 ERROR | 4 | BSS ESM GROUP2 ERROR | 6-5 | RESERVED | 7 | BSS monitoring failure in Mode 1(Quiet mode) | Others | RESERVED |
| Value | Definition | | | | | | | | | | | | | | | | | | | |
| 0 | No fault | | | | | | | | | | | | | | | | | | | |
| 1 | BSS FW assert | | | | | | | | | | | | | | | | | | | |
| 2 | BSS FW abort | | | | | | | | | | | | | | | | | | | |
| 3 | BSS ESM GROUP1 ERROR | | | | | | | | | | | | | | | | | | | |
| 4 | BSS ESM GROUP2 ERROR | | | | | | | | | | | | | | | | | | | |
| 6-5 | RESERVED | | | | | | | | | | | | | | | | | | | |
| 7 | BSS monitoring failure in Mode 1(Quiet mode) | | | | | | | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | | | |

5.21.10 Sub block 0x5009 – AWR_AE_MSS_VMON_ERRORSTATUS_SB

This sub block indicates fault in analog supplies or LDO short circuit condition. Once a fault is detected the functionality cannot be resumed from then on and the sensor needs to be re-started (This API AE is not supported).

Table 5.152: AWR_AE_MSS_VMON_ERRORSTATUS_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------|---|---------------------|--------|----------|
| SBLKID | 2 | Value = 0x5009 | | | | | | | | |
| SBLKLEN | 2 | Value = 16 | | | | | | | | |
| FAULT_TYPE | 1 | <table> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>NO FAULT</td> </tr> <tr> <td>1</td> <td>ANALOG_SUPPLY_FAULT</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> | Value | Definition | 0 | NO FAULT | 1 | ANALOG_SUPPLY_FAULT | Others | RESERVED |
| Value | Definition | | | | | | | | | |
| 0 | NO FAULT | | | | | | | | | |
| 1 | ANALOG_SUPPLY_FAULT | | | | | | | | | |
| Others | RESERVED | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |

Continued on next page

Table 5.152 – continued from previous page

| | | | |
|-----------|---|------------|--------------------------------------------------------------|
| FAULT_SIG | 4 | Bit | Definition |
| | | b0 | VDDIN under voltage indication |
| | | b1 | VDDIN over voltage indication |
| | | b2 | VIN_18CLK supply fault |
| | | b3 | VIOIN supply fault (Unable to resolve between 1.8V and 3.3V) |
| | | b4 | VIN_SRAM under voltage indication |
| | | b5 | VIOIN_18 supply fault |
| | | b6 | APLL_VCO_LDO short circuit |
| | | b31:7 | RESERVED |
| RESERVED | 4 | 0x00000000 | |

5.21.11 Sub block 0x500A – AWR_AE_MSS_ADC_DATA_SB

This async event is in response to the command which indicates ADC data needs to be transferred over SPI. This async event contains the ADC data followed by more such async events for additional data.

Table 5.153: AWR_AE_MSS_ADC_DATA_SB contents

| Field Name | Number of bytes | Description |
|------------|--------------------------|---------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x500A |
| SBLKLEN | 2 | Value = Variable (max = 226) |
| REMCHUNKS | 2 | Number of remaining chunks expected. (Remaining length / 220 bytes) |
| ADC_DATA | Variable (4 - 220 bytes) | ADC data captured by the MMIC |

5.21.12 Sub block 0x500B – RESERVED

5.22 Brief notes on the order of issuing API SBs

5.22.1 Single device mode

This section briefly describes in which order to issue the various API SBs defined in this document for a single device.

1. Power up the device
2. Wait for AWR_AE_MSSPOWERUPDONE_SB
3. AWR_DEV_CONFIGURATION_SET_SB
4. AWR_DEV_RFPOWERUP_SB
5. Wait for AWR_AE_RFPOWERUPDONE_SB
6. AWR_RF_MISC_CONF_SET_MSG
7. AWR_RF_STATIC_CONF_SET_MSG
 - a. AWR_RF_DEVICE_CFG_SB
 - b. AWR_CHAN_CONF_SET_SB
 - c. AWR_ADCOUT_CONF_SET_SB
 - d. AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS.EN set to 1 if RF supply is 1.0 V
 - e. AWR_LOWPOWERMODE_CONF_SET_SB
 - f. AWR_DYNAMICPOWERSAVE_CONF_SET_SB
 - g. AWR_CAL_MON_FREQUENCY_TX_POWER_LIMITS_SB
8. Data path configurations
 - a. AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB
 - b. AWR_DEV_RX_DATA_PATH_CONF_SET_SB
 - c. AWR_DEV_RX_DATA_PATH_LANE_EN_SB
 - d. AWR_DEV_RX_DATA_PATH_CLK_SET_SB
 - e. AWR_HIGHSPEEDINTFCLK_CONF_SET_SB
 - f. AWR_DEV_LVDS_CFG_SET_SB / AWR_DEV_CSI2_CFG_SET_SB
9. AWR_RF_INIT_MSG
 - a. AWR_CAL_DATA_RESTORE_SB (To restore factory calibration data to avoid on field RF interference during calibration)
 - b. Wait for AWR_AE_RF_INITCALIBSTATUS_SB
 - c. AWR_RF_INIT_CALIBRATION_CONF_SB (Enable only required calibration to run)
 - d. AWR_RFINIT_SB: This triggers very basic calibrations and RF initializations
 - e. Wait for AWR_AE_RF_INITCALIBSTATUS_SB
10. AWR_RF_DYNAMIC_CONF_SET_MSG
 - a. AWR_PROG_FILT_COEFF_RAM_SET_SB (Applicable only in xWR1642/IWR6843/xWR1843)
 - b. AWR_PROG_FILT_CONF_SET_SB (Applicable only in xWR1642/IWR6843/xWR1843)

- c. AWR_PROFILE_CONF_SET_SB
- d. AWR_CHIRP_CONF_SET_SB
- e. AWR_LOOPBACK_BURST_CONF_SET_SB (if using loopback burst in advance frame config API)
- f. AWR_FRAME_CONF_SET_SB or AWR_ADVANCED_FRAME_CONF_SB
- g. AWR_CALIB_MON_TIME_UNIT_CONF_SB with CALIB_MON_TIME_UNIT value set to a value such that the total frame idle time across multiple CALIB_MON_TIME_UNITS is sufficient for all calibrations and monitoring. See Section 10 for details on calibration and monitoring durations. If any error AWR_CAL_MON_TIMING_FAIL_REPORT_AE_SB AE will be generated. The calibrations and monitors will not run properly if this error is generated.
- h. AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB (set all ONE_TIME_CALIB_ENABLE_MASK and set ENABLE_CAL_REPORT = 1)
- i. Wait for AWR_RUN_TIME_CALIBRATION_SUMMARY_REPORT_AE_SB
- j. AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB (set all RUN_TIME_CALIB_ENABLE_MASK and set ENABLE_CAL_REPORT = 0 to avoid receiving periodic async events)
- k. AWR_DEV_FRAME_CONFIG_APPLY_SB or AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB

11. MONITOR CONFIGURATIONS

- a. AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB : Wait for AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB AE. This API should not be issue when frames are running.
- b. AWR_MSS_LATENTFAULT_TEST_CONF_SB : Wait for AWR_AE_MSS_LATENTFAULT_TESTREPORT_SB AE.
- c. AWR_MSS_PERIODICTESTS_CONF_SB : Enable periodic digital monitors, the monitor starts immediately after enabling this API.
- d. AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB : Enable periodic digital monitors
- e. AWR_MONITOR_ANALOG_ENABLES_CONF_SB : Enable periodic analog monitors, The corresponding monitoring configuration APIs should be issued after issuing this API. Refer latest release note for all supported monitors.

12. AWR_RF_FRAME_TRIG_MSG

- a. AWR_FRAMESTARTSTOP_CONF_SB in Start mode: after this, frames get transmitted, wait for AWR_AE_RF_FRAME_TRIGGER_RDY_SB AE.

13. AWR_RF_FRAME_TRIG_MSG

- a. AWR_FRAMESTARTSTOP_CONF_SB in Stop mode: after this, frames are stopped. Wait for AWR_FRAME_END_AE_SB AE. The AWR_RF_FRAME_TRIG_MSG may be issued multiple times for multiple sets of frames

5.22.2 Cascaded device mode

This section briefly describes in which order to issue the various API SBs defined in this document for master and slave devices in a cascaded configuration.

When using cascaded devices, the reference clock is provided by master to slave. So unless master is powered-up and clock is available from master to slave, the slave device cannot be powered up.

Table 5.154: Sequence of APIs to be issued to master and slave devices in cascaded mode configuration for FMCW mode measurements

| Sl. No | Master device sequence | Slave device sequence |
|--------|----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1 | Power up master device | |
| 2 | Wait for AWR_AE_DEV_MSSPOWERUP-DONE_SB | |
| 3 | AWR_DEV_RFPOWERUP_SB | |
| 4 | Wait for AWR_AE_DEV_RFPOWERUP-DONE_SB | |
| 5 | AWR_CHAN_CONF_SET_SB with CAS-CADING_CFG = 0x0001. This will enable the reference clock for slave device | |
| 6 | | Power on slave device |
| 7 | | Wait for AWR_AE_DEV_MSSPOWERUP-DONE_SB |
| 8 | | AWR_DEV_RFPOWERUP_SB |
| 9 | | Wait for AWR_AE_DEV_RFPOWERUP-DONE_SB |
| 10 | | AWR_CHAN_CONF_SET_SB with CAS-CADING_CFG = 0x0002. |
| 11 | AWR_ADCOUT_CONF_SET_SB | AWR_ADCOUT_CONF_SET_SB |
| 12 | AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS_EN = 1 if RF supply is 1.0V | AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS_EN = 1 if RF supply is 1.0V |
| 13 | AWR_LOWPOWERMODE_CONF_SET_SB | AWR_LOWPOWERMODE_CONF_SET_SB |
| 14 | AWR_DYNAMICPOWERSAVE_CONF_SET_SB | AWR_DYNAMICPOWERSAVE_CONF_SET_SB |
| 15 | AWR_RF_INIT_SB | AWR_RF_INIT_SB |
| 16 | Wait for AWR_AE_RF_INITALIBSTATUS_SB | Wait for AWR_AE_RF_INITALIBSTATUS_SB |
| 17 | AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB | AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB |

Continued on next page

Table 5.154 – continued from previous page

| | | |
|----|-------------------------------------------------------|-------------------------------------------------------|
| 18 | AWR_DEV_RX_DATA_PATH_CONF_SET_SB | AWR_DEV_RX_DATA_PATH_CONF_SET_SB |
| 19 | AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB | AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB |
| 20 | AWR_HIGHSPEEDINTFCLK_CONF_SET_SB | AWR_HIGHSPEEDINTFCLK_CONF_SET_SB |
| 21 | AWR_DEV_RX_DATA_PATH_CLK_SET_SB | AWR_DEV_RX_DATA_PATH_CLK_SET_SB |
| 22 | AWR_DEV_LVDS_CFG_SET_SB/AWR_DEV_CSI2_CFG_SET_SB | AWR_DEV_LVDS_CFG_SET_SB/AWR_DEV_CSI2_CFG_SET_SB |
| 23 | AWR_PROFILE_CONF_SET_SB | AWR_PROFILE_CONF_SET_SB |
| 24 | AWR_CHIRP_CONF_SET_SB | AWR_CHIRP_CONF_SET_SB |
| 25 | AWR_FRAME_CONF_SET_SB with TRIGGER_SELECT = 0x0001 | AWR_FRAME_CONF_SET_SB with TRIGGER_SELECT = 0x0002 |
| 26 | AWR_DEV_FRAME_CONFIG_APPLY_MSG | AWR_DEV_FRAME_CONFIG_APPLY_MSG |
| 27 | | AWR_FRAMESTARTSOP_CONF_SB with STARTSTOP_CMD = 0x0001 |
| 28 | | Wait for AWR_AE_RF_FRAME_TRIGGER_RDY_SB |
| 29 | AWR_FRAMESTARTSOP_CONF_SB with STARTSTOP_CMD = 0x0001 | |
| 30 | Wait for AWR_AE_RF_FRAME_TRIGGER_RDY_SB | |

5.22.3 Continuous streaming mode (in single device case)

This section briefly describes in which order to issue the various API SBs defined in this document to enable continuous streaming mode on a single device

1. Power up the device
2. Wait for AWR_AE_MSSPOWERUPDONE_SB
3. AWR_DEV_RFPOWERUP_SB
4. Wait for AWR_AE_RFPOWERUPDONE_SB
5. AWR_RF_MISC_CONF_SET_MSG
6. AWR_RF_STATIC_CONF_SET_MSG
 - a. AWR_CHAN_CONF_SET_SB
 - b. AWR_ADCOUT_CONF_SET_SB

- c. AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS_EN set to 1 if RF supply is 1.0V
 - d. AWR_LOWPOWERMODE_CONF_SET_SB
 - e. AWR_DYNAMICPOWERSAVE_CONF_SET_SB
7. AWR_RF_STATIC_CONF_SET_MSG
8. Data path configurations
- a. AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB
 - b. AWR_DEV_RX_DATA_PATH_CONF_SET_SB
 - c. AWR_DEV_RX_DATA_PATH_LANE_EN_SB
 - d. AWR_DEV_RX_DATA_PATH_CLK_SET_SB
 - e. AWR_HIGHSPEEDINTFCLK_CONF_SET_SB
 - f. AWR_DEV_LVDS_CFG_SET_SB / AWR_DEV_CSI2_CFG_SET_SB
9. AWR_RF_INIT_MSG
- a. AWR_RFINIT_SB: This triggers very basic calibrations and RF initializations
 - b. Wait for AWR_AE_RF_INITCALIBSTATUS_SB
10. AWR_CONT_STREAMING_MODE_CONF_SET_SB
11. AWR_DEV_RX_CONTSTREAMING_MODE_CONF_SET_SB
12. AWR_CONT_STREAMING_MODE_EN_SB with CONT_STREAMING_EN = 0x0001 to start continuous streaming
13. AWR_CONT_STREAMING_MODE_EN_SB with CONT_STREAMING_EN = 0x0000 to stop continuous streaming
14. Repeat steps 9-11 for a different configuration

5.22.4 Continuous streaming (CW) mode (in cascaded device case)

Table 5.155: Sequence of APIs to be issued to master and slave devices in cascaded mode for CW mode measurements

| Sl. No | Master device sequence | Slave device sequence |
|--------|----------------------------------------|-----------------------|
| 1 | Power up master device | |
| 2 | Wait for AWR_AE_DEV_MSSPOWERUP-DONE_SB | |
| 3 | AWR_DEV_RFPOWERUP_SB | |
| 4 | Wait for AWR_AE_DEV_RFPOWERUP-DONE_SB | |

Continued on next page

Table 5.155 – continued from previous page

| | | |
|----|----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 5 | AWR_CHAN_CONF_SET_SB with CAS-CADING_CFG = 0x0001. This will enable the reference clock for slave device | |
| 6 | | Power on slave device |
| 7 | | Wait for AWR_AE_DEV_MSSPOWERUP_DONE_SB |
| 8 | | AWR_DEV_RFPOWERUP_SB |
| 9 | | Wait for AWR_AE_DEV_RFPOWERUP_DONE_SB |
| 10 | | AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS_EN = 1 if RF supply is 1.0V |
| 11 | | AWR_CHAN_CONF_SET_SB with CAS-CADING_CFG = 0x0002. |
| 12 | AWR_ADCOUT_CONF_SET_SB | AWR_ADCOUT_CONF_SET_SB |
| 13 | AWR_RF_LDO_BYPASS_SB with RFLDOBYPASS_EN = 1 if RF supply is 1.0V | |
| 14 | AWR_LOWPOWERMODE_CONF_SET_SB | AWR_LOWPOWERMODE_CONF_SET_SB |
| 15 | AWR_DYNAMICPOWERSAVE_CONF_SET_SB | AWR_DYNAMICPOWERSAVE_CONF_SET_SB |
| 16 | AWR_RF_INIT_SB | AWR_RF_INIT_SB |
| 17 | Wait for AWR_AE_RF_INITALIBSTATUS_SB | Wait for AWR_AE_RF_INITALIBSTATUS_SB |
| 18 | AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB | AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB |
| 19 | AWR_DEV_RX_DATA_PATH_CONF_SET_SB | AWR_DEV_RX_DATA_PATH_CONF_SET_SB |
| 20 | AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB | AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB |
| 21 | AWR_HIGHSPEEDINTFCLK_CONF_SET_SB | AWR_HIGHSPEEDINTFCLK_CONF_SET_SB |
| 22 | AWR_DEV_RX_DATA_PATH_CLK_SET_SB | AWR_DEV_RX_DATA_PATH_CLK_SET_SB |
| 23 | AWR_DEV_LVDS_CFG_SET_SB/AWR_DEV_CSI2_CFG_SET_SB | AWR_DEV_LVDS_CFG_SET_SB/AWR_DEV_CSI2_CFG_SET_SB |
| 24 | AWR_CONT_STREAMING_MODE_CONF_SET_SB | |

Continued on next page

Table 5.155 – continued from previous page

| | | |
|----|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| 25 | AWR_CONT_STREAMING_MODE_EN_ SB with CONT_STREAMING_EN = 0x0001 to start continuous streaming | |
| 26 | | AWR_CONT_STREAMING_MODE_CONF_SET_SB with the same RF frequency configuration as in master device |
| 27 | | AWR_CONT_STREAMING_MODE_EN_ SB with CONT_STREAMING_EN = 0x0001 to start continuous streaming |
| 28 | | AWR_CONT_STREAMING_MODE_EN_ SB with CONT_STREAMING_EN = 0x0000 to stop continuous streaming |
| 29 | AWR_CONT_STREAMING_MODE_EN_ SB with CONT_STREAMING_EN = 0x0000 to stop continuous streaming | |
| 30 | Repeat steps 24-29 for a different CW mode configuration | |

6 API Error Codes

Table 6.1: BSS API error codes

| | | |
|----------------------------------|-----|----------------------------------------------------------------------------------------------------------|
| Applicable to all API sub blocks | 1 | Incorrect API MSGID |
| | 2 | Sub block not found in the MSG |
| | 3 | Incorrect Sub block ID |
| | 4 | Incorrect Sub block length |
| | 5 | Incorrect Sub block data |
| | 6 | Error in processing the command |
| | 7 | Binary file CRC mismatch error |
| | 8 | Binary file type mismatch w.r.t. magic number |
| AWR_FRAMESTARTSTOP_CONF_SB | 20 | Frames are already started when the FRAME_START command was issued |
| | 21 | Frames are already stopped when the FRAME_STOP command was issued |
| | 22 | No valid frame configuration API was issued and frames are started |
| | 23 | START_STOP_CMD parameter is out of range |
| AWR_CHAN_CONF_SET_SB | 24 | RX_CHAN_EN parameter is out of range (Max range may vary based on device variant) |
| | 25 | TX_CHAN_EN parameter is out of range (Max range may vary based on device variant) |
| | 26 | CASCADING_CFG parameter is out of range [0, 2] |
| | 282 | Device variant does not allow cascading but API is issued to enable cascading mode |
| AWR_ADCOUT_CONF_SET_SB | 27 | NUM_ADC.BITS parameter is out of range [0, 2] |
| | 28 | ADC_OUT_FMT parameter is out of range [0, 3] |
| | 127 | FULL_SCALE_REDUCTION_FACTOR is > 0 for 16 bit ADC, or > 2 for 14 bit ADC mode or > 4 for 12 bit ADC mode |
| AWR_LOWPOWERMODE_CONF_SET_SB | 29 | LP_ADC_MODE parameter is out of range [0, 1] |
| | 156 | Regular ADC mode is used on a 5 MHz part variant device |
| AWR_DYNAMICPOWERSAVE_CONF_SET_SB | 30 | BLOCK_CFG parameter is out of range [0, 7] |
| AWR_HIGHSPEEDINTFCLK_CONF_SET_SB | 31 | HSICLKRATECODE[1:0] is 0 |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|-------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| | 32 | RESERVED |
| | 33 | HSICLKRATECODE[3:2] is 3 and HSICLKRATECODE[1:0] is 2 |
| | 34 | HSICLKRATECODE[3:2] is 3 and HSICLKRATECODE[1:0] is 2 |
| AWR_PROFILE_CONF_SET_SB | 35 | PF_INDIX is ≥ 4 |
| | 36 | PF_FREQ_START_CONST is not within [76, 81] GHz |
| | 37 | PF_IDLE_TIME_CONST > 5.24 ms |
| | 38 | Maximum DFE spill time > PF_IDLE_TIME_CONST |
| | 39 | PF_ADC_START_TIME_CONST > 4095 |
| | 40 | PF_RAMP_END_TIME > 524287 |
| | 41 | PF_RAMP_END_TIME < PF_ADC_START_TIME_CONST + ADC_SAMPLING_TIME (ADC_SAMPLING_TIME is time taken to sample NUM_ADC_SAMPLES) |
| | 42 | PF_TX_OUTPUT_POWER_BACKOFF for TX0 > 30 |
| | 43 | PF_TX_OUTPUT_POWER_BACKOFF for TX1 > 30 |
| | 44 | PF_TX_OUTPUT_POWER_BACKOFF for TX2 > 30 |
| | 45 | RESERVED |
| | 46 | Ramp end frequency is not within [76, 81] GHz |
| | 47 | Absolute value of TX_START_TIME is > 38.45 μ s |
| | 48 | Number of ADC samples is not within [2, 8192] |
| | 49 | Output sampling rate is not within [2, MaxSamplingRate] Msps. See Table 5.21 for the MaxSamplingRate. |
| | 50 | HPF1 corner frequency is > 700 kHz |
| | 51 | HPF2 corner frequency is > 2.8 MHz |
| 52 | PF_RX_GAIN is not within [24, 52] dB or PF_RX_GAIN is an odd number | |
| 53 | RESERVED | |
| 54 | RESERVED | |
| 55 | RESERVED | |
| 56 | RESERVED | |
| 57 | RESERVED | |
| 58 | RESERVED | |
| AWR_CHIRP_CONF_SET_SB | 59 | CHIRP_START_INDIX ≥ 512 |
| | 60 | CHIRP_END_INDIX ≥ 512 |
| | 61 | CHIRP_START_INDIX > CHIRP_END_INDIX |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|-----------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | 62 | PROFILE.INDX \geq 4 |
| | 63 | If the profile corresponding to PROFILE.INDX is not defined |
| | 64 | CHIRP_FREQ_START_VAR > 8388607 |
| | 65 | CHIRP_FREQ_SLOPE_VAR > 63 |
| | 66 | Chirp start frequency is outside [76, 78] GHz if the selected VCO is VCO1 or Chirp start frequency is outside [77, 81] GHz if the selected VCO is VCO2 or Chirp end frequency is outside [76, 78] GHz if the selected VCO is VCO1 or Chirp end frequency is outside [77, 81] GHz if the selected VCO is VCO2 or Chirp bandwidth is greater than maximum allowed as per device data sheet or Maximum chirp frequency is greater than maximum allowed as per device data sheet |
| | 67 | CHIRP_IDLE_TIME_VAR > 4095 |
| | 68 | CHIRP_ADC_START_TIME_VAR > 4095 |
| | 69 | RAMP_END_TIME < ADC_START_TIME + ADC_SAMPLING_TIME |
| | 70 | CHIRP_TX_EN > maximum simultaneous TX allowed as per device data sheet |
| | 71 | CHIRP_TX_EN indicates to enable a TX which is not enabled in AWR_CHAN_CONF_SET_SB |
| AWR_FRAME_CONF_SET_SB | 72 | CHIRP_START_INDX \geq 512 |
| | 73 | CHIRP_END_INDX \geq 512 |
| | 74 | CHIRP_START_INDX > CHIRP_END_INDX |
| | 75 | Chirp used in the frame is not configured by AWR_CHIRP_CONF_SET_SB |
| | 76 | One of the profiles used in the frame is not configured by AWR_PROF_CONF_SET_SB |
| | 77 | NUM_LOOPS is outside [1, 255] |
| | 78 | RESERVED |
| | 79 | FRAME_PERIODICITY is outside [100 μ s, 1.342 s] |
| | 80 | FRAME_ON_TIME > FRAME_PERIODICITY |
| | 81 | TRIGGER_SELECT is outside [1, 2] |
| | 82 | FRAME_TRIGGER_DELAY > 100 μ s |
| | 83 | API is issued when frames are ongoing |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|----------------------------------|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| AWR_ADVANCED_FRAME_CONF_SET_SB | 84 | NUM.SUBFRAMES is outside [1, 4] |
| | 85 | FORCE.SINGLE.PROFILE is outside [0, 1] |
| | 86 | FORCE.SINGLE.PROFILE ≥ 4 |
| | 87 | Profile defined by FORCE.SINGLE.PROFILE is not defined |
| | 88 | SF _x .CHIRP.START.INDX ≥ 512 |
| | 89 | SF _x .NUM.UNIQUE.CHIRPS.PER.BURST is outside the range [1, 512] |
| | 90 | Chirp used in the frame is not configured by AWR.CHIRP.CONF.SET_SB |
| | 91 | One of the profiles used in the frame is not configured by AWR.PROF.CONF.SET_SB |
| | 92 | SF _x .NUM.LOOPS.PER.BURST is outside the range [1, 255] |
| | 93 | SF _x .BURST.PERIOD is outside the range [100 μ s, 1.342 s] |
| | 94 | Burst ON time is > BURST.PERIOD |
| | 95 | SF _x .CHIRP.START.INDX.OFFSET ≥ 512 |
| | 96 | SF _x .CHIRP.START.INDX ≥ 512 or SF _x .CHIRP.START.INDX + SF _x .NUM.UNIQUE.CHIRPS.PER.BURST - 1 is ≥ 512 |
| | 97 | SF _x .NUM.BURSTS is outside the range [1, 512] |
| | 98 | SF _x .NUM.OUTER.LOOPS is outside the range [1, 64] |
| | 99 | SF _x .PERIOD is outside the range [100 μ s, 1.342 s] |
| 100 | Subframe on time > SF _x .PERIOD or when TESTSOURCE is enabled, SubFrame Idle time is < 150 μ s | |
| 101 | RESERVED | |
| 102 | TRIGGER.SELECT is outside the range [1, 2] | |
| 103 | FRAME.TRIGGER.DELAY is > 100 μ s | |
| 104 | API is issued when frames are on going | |
| AWR_RF_TEST_SOURCE_CONFIG_SET_SB | 105 | POSITION.VEC _x [y] < 0 |
| | 106 | RESERVED |
| | 107 | VELOCITY.VEC _x [x] > 5000 or VELOCITY.VEC _x [y] > 5000 or VELOCITY.VEC _x [z] > 5000 |
| | 108 | SIG.LEV.VEC _x > 950 |
| | 109 | RX.ANT.POS.XZ[Byte _x] > 120 |
| | 110 | RESERVED |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|----------------------------------------------|-----|------------------------------------------------------------------------------------------------------------|
| AWR_PROG_FILT_CONF_SET_SB | 111 | PROG_FILT_COEFF_START_INDEX is an odd number |
| | 112 | PROFILE_INDEX \geq 4 |
| | 126 | DFE mode is pseudo real |
| AWR_PROG_FILT_COEFF_RAM_SET_SB | 113 | API is issued for a non xWR1642/xWR1843 device |
| | 126 | DFE mode is pseudo real |
| AWR_RF_RADAR_MISC_CTL_SB | 114 | API is issued for an unsupported device |
| AWR_PERCHIRPPHASESHIFT_CONF_SB | 115 | CHIRP_START_INDEX \geq 512 |
| | 116 | CHIRP_END_INDEX \geq 512 |
| | 117 | CHIRP_START_INDEX > CHIRP_END_INDEX |
| AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB | 118 | Boot time calibrations are not done so cannot run runtime calibrations |
| AWR_CAL_MON_FREQUENCY_LIMITS_SB | 119 | FREQ_LIMIT_HIGH < 76 GHz or FREQ_LIMIT_HIGH > 81 GHz or FREQ_LIMIT_LOW > FREQ_LIMIT_HIGH |
| AWR_CALIB_MON_TIME_UNIT_CONF_SB | 120 | CALIB_MON_TIME_UNIT \leq 0 |
| AWR_RUN_TIME_CALIBRATION_CONF_AND_TRIGGER_SB | 121 | CALIBRATION_PERIODICITY = 0 |
| | 122 | API is issued when continuous streaming mode is on |
| | 123 | RX gain run time calibration was requested but boot time calibration was not performed |
| | 124 | LO distribution run time calibration was requested but boot time calibration was not performed |
| | 125 | TX power run time calibration was requested but boot time calibration was not performed |
| AWR_LOOPBACK_BURST_CONF_SET_SB | 132 | LOOPBACK_SEL is > 3 |
| | 133 | BURST_INDEX \geq 16 |
| | 134 | Burst is not valid but loopback is enabled for this burst |
| AWR_DYN_CHIRP_CONF_SET_SB | 135 | CHIRP_SEGMENT_SELECT > 31 if CHIRP_ROW_SELECT = 0 or CHIRP_SEGMENT_SELECT > 11 if CHIRP_ROW_SELECT != 0 |
| | 159 | CHIRP_ROW_SELECT > 3 |
| AWR_DYN_PER_CHIRP_PHASESHIFTER_CONF_SB | 136 | CHIRP_SEGMENT_SELECT > 31 |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------------------------------------------|
| AWR_CAL_DATA_RE- STORE_SB | 137 | CHUNK.ID \geq NUM.CHUNKS |
| | 138 | CAL.DATA is invalid |
| AWR_INTERCHIRP_ BLOCKCONTROLS_SB | 139 | RX02_RF_TURN_OFF_TIME is not within the range [-1024, 1023] |
| | 140 | RX13_RF_TURN_OFF_TIME is not within the range [-1024, 1023] |
| | 141 | RX02_BB_TURN_OFF_TIME is not within the range [-1024, 1023] |
| | 142 | RX13_BB_TURN_OFF_TIME is not within the range [-1024, 1023] |
| | 143 | RX02_RF_PREENABLE_TIME is not within the range [-1024, 1023] |
| | 144 | RX13_RF_PREENABLE_TIME is not within the range [-1024, 1023] |
| | 145 | RX02_BB_PREENABLE_TIME is not within the range [-1024, 1023] |
| | 146 | RX13_BB_PREENABLE_TIME is not within the range [-1024, 1023] |
| | 147 | RX02_RF_TURN_ON_TIME is not within the range [-1024, 1023] |
| | 148 | RX13_RF_TURN_ON_TIME is not within the range [-1024, 1023] |
| | 149 | RX02_BB_TURN_ON_TIME is not within the range [-1024, 1023] |
| | 150 | RX13_BB_TURN_ON_TIME is not within the range [-1024, 1023] |
| | 151 | RX_LO_TURN_OFF_TIME is not within the range [-1024, 1023] |
| | 152 | TX_LO_TURN_OFF_TIME is not within the range [-1024, 1023] |
| 153 | RX_LO_TURN_ON_TIME is not within the range [-1024, 1023] | |
| 154 | TX_LO_TURN_ON_TIME is not within the range [-1024, 1023] | |
| AWR_SUBFRAME_ START_CONF_SB | 155 | Sub-frame start command is issued but the frame is not configured for sub frame trigger mode |
| Common to all monitoring configuration APIs | 250 | Device type is not ASILB |
| | 251 | Fault injection API or Digital latent fault API is issued when frames are ongoing |
| | 252 | Invalid reporting mode |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|-------------------------------------------------|-----|-------------------------------------------------------------------------------------------------|
| | 253 | Configured profile ID is not within [0, 3] |
| | 254 | Monitoring profile ID is not configured yet |
| | 260 | Invalid RF bit mask |
| | 281 | Analog monitoring is not supported |
| | 290 | Monitoring chirp error |
| AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB | 251 | API is issued when frames are on-going |
| AWR_MONITORING_EXTERNAL_ANALOG_SIGNALS_CONF_SB | 255 | Settling time is configured is more than 12 μ s |
| AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB | 256 | None of the RXs are enabled |
| AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB | 257 | TX0 is not enabled |
| AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB | 258 | TX1 is not enabled |
| AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB | 259 | TX2 is not enabled |
| - | 261 | RESERVED |
| - | 262 | RESERVED |
| AWR_MONITOR_TXn_BALLBREAK_CONF_SB | 263 | Monitored TX channel is not enabled |
| AWR_MONITOR_RX_GAIN_PHASE_CONF_SB | 264 | Monitored RX channel is not enabled |
| AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB | | |
| AWR_MONITOR_RX_GAIN_PHASE_CONF_SB | 265 | TX selected for RX gain phase monitor is TX2 (Only TX0 or TX1 is allowed) |
| | 291 | PD power level is less than -40dBm (Used for RX Gain Monitor) |
| | 295 | PGA Gain used for monitoring is incorrect |
| AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB | 266 | SAT_MON_SEL is not in [0, 3] |
| | 267 | SAT_MON_PRIMARY_TIME_SLICE_DURATION is less than 0.64 μ s or greater than ADC sampling time |
| | 268 | SAT_MON_NUM_SLICES is 0 or greater than 127 |

Continued on next page

Table 6.1 – continued from previous page

| | | |
|----------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------|
| | 283 | RX saturation monitor is not supported |
| AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB | 269 | SIG_IMG_MON_NUM_SLICES is 0 or greater than 127 |
| | 270 | NUM_SAMPLES_PER_PRIMARY_TIME_SLICE is odd, or less than 4 in Complex1x mode or less than 8 in non-Complex1x modes or greater than NUM_ADC_SAMPLES |
| | 280 | Signal and image band monitor is not supported |
| AWR_ANALOG_FAULT_INJECTION_CONF_SB | 279 | LDO fault inject is requested but LDOs are bypassed |
| AWR_MONITOR_TXn_POWER_CONF_SB AWR_MONITOR_TXn_BALLBREAK_CONF_SB AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB | 294 | PD Reading incorrect (RF OFF reading higher than RF ON reading) |
| AWR_MONITOR_RX_GAIN_PHASE_CONF_SB AWR_MONITOR_TX_GAIN_PHASE_CONF_SB AWR_MONITOR_TXn_BPM_CONF_SB | 292 | ADC power level higher than +7 dBm or lower than -9.5 dBm |
| AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB | 293 | Low RX noise figure (Noise Figure is less than 0 dB) |

Table 6.2: MSS API error codes (Applicable only in AWR1243)

| | | |
|------------------------------------|------|-----------------------------------------------|
| Applicable to all API sub blocks | 1 | Incorrect API MSGID |
| | 2 | Sub block not found in the MSG |
| | 3 | Incorrect Sub block ID |
| | 4 | Incorrect Sub block length |
| | 5 | Incorrect Sub block data |
| | 6 | Error in processing the command |
| | 7 | Binary file CRC mismatch error |
| | 8 | Binary file type mismatch w.r.t. magic number |
| AWR_DEV_RX_DATA_FORMAT_CONF_SET_SB | 1001 | RX_CHAN_EN > 0xF |
| | 1002 | NUM_ADC_BITS > 2 |
| | 1003 | ADC_OUT_FMT > 1 |

Continued on next page

Table 6.2 – continued from previous page

| | | |
|--------------------------------------------|------|------------------------------------------------------------------------------|
| | 1004 | IQ_SWAP_SEL > 1 |
| | 1005 | CHAN_INTERLEAVE > 1 |
| AWR_DEV_RX_DATA_PATH_CONF_SET_SB | 1006 | DATA_INTF_SEL > 1 |
| | 1007 | DATA_TRANS_FMT_PKT0 [5:0] not a valid value. Valid set {0x1, 0x6, 0x9, 0x36} |
| | 1008 | DATA_TRANS_FMT_PKT1 [5:0] not a valid value. Valid set {0x0, 0xD, 0xB} |
| | 1050 | CQ_CONFIG is out of range |
| AWR_DEV_RX_DATA_PATH_LANEEN_SET_SB | 1009 | LANE_EN > 0xF |
| | 1010 | Reserved |
| AWR_DEV_RX_DATA_PATH_CLK_SET_SB | 1011 | LANE_CLK_CFG > 1 |
| | 1012 | LANE_CLK_CFG != 1 for CSI2 |
| | 1013 | DATA_RATE - Invalid combination of data rate and DDR or SDR operation |
| AWR_DEV_LVDS_CFG_SET_SB | 1014 | LANE_FMT_MAP > 1 |
| | 1015 | LANE_PARAM_CFG > 7 |
| AWR_DEV_RX_CON_TSTREAMING_MODE_CONF_SET_SB | 1016 | CONT_STREAMING_MODE > 1 |
| | 1017 | CONT_STREAMING_MODE already in requested mode |
| AWR_DEV_CSI2_CFG_SET_SB | 1018 | LANE_POS_POL_SEL [DATA_LANE0_POS] > 5 |
| | 1019 | LANE_POS_POL_SEL [DATA_LANE1_POS] > 5 |
| | 1020 | LANE_POS_POL_SEL [DATA_LANE2_POS] > 5 |
| | 1021 | LANE_POS_POL_SEL [DATA_LANE3_POS] > 5 |
| | 1022 | LANE_POS_POL_SEL [CLOCK_POS] is outside the range [2,4] |
| AWR_DEV_FRAME_CONFIG_APPLY_SB | 1023 | HALF_WORDS_PER_CHIRP is outside the range [64, 8192] |
| AWR_DEV_ADV_FRAME_CONFIG_APPLY_SB | 1024 | NUM_SUBFRAMES is outside the range [1,4] |
| | 1025 | SF1_TOT_NUM_CHIRPS is outside the range [1, 0xFFFF] |
| | 1026 | SF1_NUM_ADC_SAMPLES_PER_DATA_PKT is outside the range [64, 8192] |
| | 1027 | SF1_PROC_NUM_CHIRPS_PER_DATA_PKT != 1 |

Continued on next page

Table 6.2 – continued from previous page

| | | |
|-----------------------------------|------|-----------------------------------------------------------------------------------------------------------------------|
| | 1028 | SF2_TOT_NUM_CHIRPS is outside the range [1, 0xFFFF], if NUM.SUBFRAMES \geq 2 |
| | 1029 | SF2_NUM_ADC_SAMPLES_PER_DATA_PKT is outside the range [64, 8192], if NUM.SUBFRAMES \geq 2 |
| | 1030 | SF2_PROC_NUM_CHIRPS_PER_DATA_PKT \neq 1, if NUM.SUBFRAMES \geq 2 |
| | 1031 | SF3_TOT_NUM_CHIRPS is outside the range [1, 0xFFFF], if NUM.SUBFRAMES \geq 3 |
| | 1032 | SF3_NUM_ADC_SAMPLES_PER_DATA_PKT is outside the range [64, 8192], if NUM.SUBFRAMES \geq 3 |
| | 1033 | SF3_PROC_NUM_CHIRPS_PER_DATA_PKT \neq 1, if NUM.SUBFRAMES \geq 3 |
| | 1034 | SF4_TOT_NUM_CHIRPS is outside the range [1, 0xFFFF], if NUM.SUBFRAMES == 4 |
| | 1035 | SF4_NUM_ADC_SAMPLES_PER_DATA_PKT is outside the range [64, 8192], if NUM.SUBFRAMES == 4 |
| | 1036 | SF4_PROC_NUM_CHIRPS_PER_DATA_PKT \neq 1, if NUM.SUBFRAMES == 4 |
| AWR_DEV_MCUCLOCK_CONF_SET_SB | 1040 | MCUCLOCK_CTRL is out of range |
| | 1041 | MCUCLOCK_SRC is out of range |
| AWR_DEV_PMICCLOCK_CONF_SET_SB | 1042 | PMICCLOCK_CTRL is out of range |
| | 1043 | PMICCLOCK_SRC is out of range |
| | 1044 | MODE_SELECT is out of range |
| | 1045 | FREQ_SLOPE is out of range |
| | 1046 | CLK_DITHER_EN is out of range |
| AWR_DEV_TESTPATTERN_GEN_SET_SB | 1047 | TESTPATTERN_GEN_CTRL is out of range |
| | 1048 | DATA_INTF_SEL (Data interface selected in AWR_DEV_RX_DATA_PATH_CONF_SET_SB) is SPI |
| AWR_MSS_LATENT_FAULT_TEST_CONF_SB | 1051 | RL_API_NRESP_LFAULTTEST_UNSUPPORTED_OOR (Unsupported Latent Fault test selected in AWR_MSS_LATENT_FAULT_TEST_CONF_SB) |

6.1 Error codes for boot on SPI

Table 6.3: Bit field describing the error status during boot on SPI

| Error description | Error code | Error code bit position |
|----------------------------------------------|------------|-------------------------|
| CERT_AUTH_FAILURE | 0x00000001 | BIT0 |
| CERT_PARSER_FAILURE | 0x00000002 | BIT1 |
| RPRC_IMG1_AUTH_FAILURE | 0x00000004 | BIT2 |
| RPRC_IMG2_AUTH_FAILURE | 0x00000008 | BIT3 |
| RPRC_IMG3_AUTH_FAILURE | 0x00000010 | BIT4 |
| RPRC_HDR_NOT_FOUND | 0x00000020 | BIT5 |
| METAHEADER_NOT_FOUND | 0x00000040 | BIT6 |
| SW_ANTIROLLBACK_CHK_FAILURE | 0x00000080 | BIT7 |
| EFUSE_INTEGRITY_FAILURE | 0x00000100 | BIT8 |
| CERT_FIELD_VALIDITY_FAILURE | 0x00000200 | BIT9 |
| CERT_FIELD_INVALID_AUTH_KEY_INDEX | 0x00000400 | BIT10 |
| CERT_FIELD_INVALID_HASH_TYPE | 0x00000800 | BIT11 |
| CERT_FIELD_INVALID_SUBSYSTEM | 0x00001000 | BIT12 |
| CERT_FIELD_INVALID_DECRYPT_KEY_INDEX | 0x00002000 | BIT13 |
| CERT_FIELD_CEK_EFUSE_MISMATCH | 0x00004000 | BIT14 |
| CERT_FIELD_CEK1_EFUSE_MISMATCH | 0x00008000 | BIT15 |
| CERT_FIELD_CEK2_EFUSE_MISMATCH | 0x00010000 | BIT16 |
| CERT_FIELD_INVALID_SUBSYSTEM_BANK_ALLOCATION | 0x00020000 | BIT17 |
| CERT_FIELD_INVALID_TOTAL_BANKS_ALLOCATION | 0x00040000 | BIT18 |
| RPRC_PARSER_FILE_LENGTH_MISMATCH | 0x00080000 | BIT19 |
| RPRC_PARSER_MSS_FILE_OFFSET_MISMATCH | 0x00100000 | BIT20 |
| RPRC_PARSER_BSS_FILE_OFFSET_MISMATCH | 0x00200000 | BIT21 |
| RPRC_PARSER_DSS_FILE_OFFSET_MISMATCH | 0x00400000 | BIT22 |
| CERT_FIELD_INVALID_DECRYPT_KEY | 0x00800000 | BIT23 |
| CERT_FIELD_INVALID_AUTH_KEY | 0x01000000 | BIT24 |
| HS_DEVICE_CERT_NOT_PRESENT | 0x02000000 | BIT25 |
| TEST_PORT_ENABLING_FAILED | 0x04000000 | BIT26 |
| SHARED_MEM_ALLOC_FAILED | 0x08000000 | BIT27 |
| MSSIMAGE_NOT_FOUND | 0x10000000 | BIT28 |
| METAHEADER_NUMFILES_ERROR | 0x20000000 | BIT29 |
| METAHEADER_CRC_FAILURE | 0x40000000 | BIT30 |

7 Radar Monitoring APIs

sec:RadarMonApis

AWR monitoring can be configured through a set of API sub blocks defined in this section. Note that these APIs cover the RF/Analog related monitoring mechanisms. There are separate monitoring mechanisms for the digital logic (including the processor, memory, etc.) which are internal to the device and not explicitly enabled through these APIs.

The monitoring APIs are structured as follows. There are common configuration APIs that control the overall periodicity of monitoring, as well as, enable/disable control for each monitoring mechanism. Then, for each monitoring mechanism there is an individual API to allow the customer to set an appropriate threshold for declaring failure from that monitoring. Also, for each monitoring mechanism, there is an individual API to report soft (raw) values from that monitoring.

NOTE1: Each monitor can perform monitoring on only one profile at a time. Though it is possible that different monitors can monitor different profiles simultaneously.

NOTE2: None of the Safety Monitoring supported in QM devices (IWR6843, xWR1443, IWR1642 and IWR1843), The monitoring configurations defined below from sub-block ID 0x01C0 to 0x01DF are not valid in QM devices.

NOTE3: None of the Safety Monitoring supported in xWR1443 ES3.0 devices except Rx saturation and signal image monitor defined in page 319, only these two monitors are supported in QM devices.

NOTE4: All Monitoring configurations and enable control APIs shall be issues before triggering the frames. The run time programming or configuration update for monitors are not supported while frames are running.

7.1 Common Configurations and Reports

This section covers the APIs corresponding to the common configurations and reports.

7.1.1 Sub block 0x01C0 – AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB

This API SB contains the consolidated configuration of all digital monitoring. This is issued by the host to the AWR device.

The enabled monitoring functions are executed when the API is issued, this API should be issued only when frames are not running, these are destructive tests. The scheduling of these monitoring should be handled in the external application. Report of these monitoring will be available in the async event [AWR_MONITOR_RF_DIG_LATENTFAULT_REPORT_AE_SB](#).

Table 7.1: AWR_MONITOR_RF_DIG_LATENTFAULT_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x01C0 |
| SBLKLEN | 2 | Value = 16 |

Continued on next page

Table 7.1 – continued from previous page

| | | |
|------------------------|---|-------------------------------------------------------------------------------------|
| DIG_MONITORING_ENABLES | 4 | 1 – Enable, 0 – Disabled |
| | | Bit Definition |
| | | b0 RESERVED |
| | | b1 CR4 and VIM lockstep test |
| | | b2 RESERVED |
| | | b3 VIM test (Not supported in 1st Gen devices, refer latest release note) |
| | | b4 RESERVED |
| | | b5 RESERVED |
| | | b6 CRC test |
| | | b7 RAMPGEN memory ECC (Not supported in 1st Gen devices, refer latest release note) |
| | | b8 DFE Parity test |
| | | b9 DFE memory ECC |
| | | b10 RAMPGEN lockstep test |
| | | b11 FRC lockstep test of diagnostic |
| | | b12 RESERVED |
| | | b13 RESERVED |
| | | b14 RESERVED |
| | | b15 RESERVED |
| | | b16 ESM test |
| | | b17 DFE STC |
| | | b18 RESERVED |
| | | b19 ATCM, BTCM ECC test |
| | | b20 ATCM, BTCM parity test |
| | | b21 RESERVED |
| | | b22 RESERVED |
| | | b23 RESERVED |
| | | b24 FFT test |
| | | b25 RTI test |
| b26 RESERVED | | |
| b31:27 RESERVED | | |

Continued on next page

Table 7.1 – continued from previous page

| | | | |
|-----------|---|------------|-------------------------------------------------------------------------------------------------------------------------|
| TEST_MODE | 1 | Value | Definition |
| | | 0 | Production mode. Latent faults are tested and any failures are reported |
| | | 1 | Characterization mode. Faults are injected and failures are reported which allows testing of the failure reporting path |
| RESERVED | 3 | 0x000000 | |
| RESERVED | 4 | 0x00000000 | |

NOTE: The Characterization TEST_MODE is supported only for debug, in production or run time this test mode is not supported. The device reset is required after entering this mode.

7.1.2 Sub block 0x01C1 – AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB

This API SB contains the consolidated configuration of all periodic digital monitoring within radar sub-system. This is issued by the host to the AWR device.

The enabled monitoring functions are executed periodically and reports are sent based on reporting mode. Report of these monitoring will be available in the async event [AWR_MONITOR_RF_DIG_PERIODIC_REPORT_AE_SB](#).

Table 7.2: AWR_MONITOR_RF_DIG_PERIODIC_CONF_SB contents

| Field Name | Number of bytes | Description | |
|---------------------|-----------------|-------------------------|----------------------------------------|
| SBLKID | 2 | Value = 0x01C1 | |
| SBLKLEN | 2 | Value = 16 | |
| REPORTING_MODE | 1 | Value | Definition |
| | | 0 | Report is sent every monitoring period |
| | | 1 | Report is sent only on a failure |
| | | 2 | RESERVED |
| RESERVED | 3 | 0x000000 | |
| PERIODIC.DIG.MON.EN | 4 | 1 – Enable, 0 – Disable | |
| | | Bit | Monitoring type |
| | | b0 | PERIODIC.CONFG.REGISTER.READ.EN |
| | | b1 | RESERVED |
| | | b2 | DFE.STC.EN |
| | | b3 | FRAME.TIMING.MONITORING.EN |
| b31:4 | RESERVED | | |

Continued on next page

Table 7.2 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
|----------|---|------------|

7.1.3 Sub block 0x01C2 – AWR_MONITOR_ANALOG_ENABLES_CONF_SB

This API SB contains the consolidated configuration of all analog monitoring. This is issued by the host to the AWR device.

The enabled monitoring functions are executed with a periodicity of CAL_MON.TIME_UNITS number of logical frames. The host should ensure that all the enabled monitors can be completed in the available inter-frame times, based on the monitoring durations (to be provided separately).

Table 7.3: AWR_MONITOR_ANALOG_ENABLES_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x01C2 |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 7.3 – continued from previous page

| ANA_MONITORING_ENABLES | 4 | <p>If any bit in this field is set to 1, the associate monitors are enabled. The configurations and reports of each monitors are described in respective sub sections.</p> <table border="0"> <thead> <tr> <th data-bbox="662 436 694 459">Bit</th> <th data-bbox="758 436 869 459">Definition</th> </tr> </thead> <tbody> <tr><td>b0</td><td>TEMPERATURE_MONITOR</td></tr> <tr><td>b1</td><td>RX_GAIN_PHASE_MONITOR</td></tr> <tr><td>b2</td><td>RX_NOISE_FIGURE_MONITOR</td></tr> <tr><td>b3</td><td>RX_IFSTAGE_MONITOR</td></tr> <tr><td>b4</td><td>TX0_POWER_MONITOR</td></tr> <tr><td>b5</td><td>TX1_POWER_MONITOR</td></tr> <tr><td>b6</td><td>TX2_POWER_MONITOR</td></tr> <tr><td>b7</td><td>TX0_BALLBREAK_MONITOR</td></tr> <tr><td>b8</td><td>TX1_BALLBREAK_MONITOR</td></tr> <tr><td>b9</td><td>TX2_BALLBREAK_MONITOR</td></tr> <tr><td>b10</td><td>TX_GAIN_PHASE_MISMATCH_MONITOR</td></tr> <tr><td>b11</td><td>TX0_BPM_MONITOR</td></tr> <tr><td>b12</td><td>TX1_BPM_MONITOR</td></tr> <tr><td>b13</td><td>TX2_BPM_MONITOR</td></tr> <tr><td>b14</td><td>SYNTH_FREQ_MONITOR</td></tr> <tr><td>b15</td><td>EXTERNAL_ANALOG_SIGNALS_MONITOR</td></tr> <tr><td>b16</td><td>INTERNAL_TX0_SIGNALS_MONITOR</td></tr> <tr><td>b17</td><td>INTERNAL_TX1_SIGNALS_MONITOR</td></tr> <tr><td>b18</td><td>INTERNAL_TX2_SIGNALS_MONITOR</td></tr> <tr><td>b19</td><td>INTERNAL_RX_SIGNALS_MONITOR</td></tr> <tr><td>b20</td><td>INTERNAL_PMCLKLO_SIGNALS_MONITOR</td></tr> <tr><td>b21</td><td>INTERNAL_GPADC_SIGNALS_MONITOR</td></tr> <tr><td>b22</td><td>PLL_CONTROL_VOLTAGE_MONITOR</td></tr> <tr><td>b23</td><td>DCC_CLOCK_FREQ_MONITOR</td></tr> <tr><td>b24</td><td>RX_SATURATION_DETECTOR_MONITOR</td></tr> <tr><td>b25</td><td>RX_SIG_IMG_BAND_MONITOR</td></tr> <tr><td>b26</td><td>RX_MIXER_INPUT_POWER_MONITOR</td></tr> <tr><td>b31:27</td><td>RESERVED</td></tr> </tbody> </table> | Bit | Definition | b0 | TEMPERATURE_MONITOR | b1 | RX_GAIN_PHASE_MONITOR | b2 | RX_NOISE_FIGURE_MONITOR | b3 | RX_IFSTAGE_MONITOR | b4 | TX0_POWER_MONITOR | b5 | TX1_POWER_MONITOR | b6 | TX2_POWER_MONITOR | b7 | TX0_BALLBREAK_MONITOR | b8 | TX1_BALLBREAK_MONITOR | b9 | TX2_BALLBREAK_MONITOR | b10 | TX_GAIN_PHASE_MISMATCH_MONITOR | b11 | TX0_BPM_MONITOR | b12 | TX1_BPM_MONITOR | b13 | TX2_BPM_MONITOR | b14 | SYNTH_FREQ_MONITOR | b15 | EXTERNAL_ANALOG_SIGNALS_MONITOR | b16 | INTERNAL_TX0_SIGNALS_MONITOR | b17 | INTERNAL_TX1_SIGNALS_MONITOR | b18 | INTERNAL_TX2_SIGNALS_MONITOR | b19 | INTERNAL_RX_SIGNALS_MONITOR | b20 | INTERNAL_PMCLKLO_SIGNALS_MONITOR | b21 | INTERNAL_GPADC_SIGNALS_MONITOR | b22 | PLL_CONTROL_VOLTAGE_MONITOR | b23 | DCC_CLOCK_FREQ_MONITOR | b24 | RX_SATURATION_DETECTOR_MONITOR | b25 | RX_SIG_IMG_BAND_MONITOR | b26 | RX_MIXER_INPUT_POWER_MONITOR | b31:27 | RESERVED |
|------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|----|---------------------|----|-----------------------|----|-------------------------|----|--------------------|----|-------------------|----|-------------------|----|-------------------|----|-----------------------|----|-----------------------|----|-----------------------|-----|--------------------------------|-----|-----------------|-----|-----------------|-----|-----------------|-----|--------------------|-----|---------------------------------|-----|------------------------------|-----|------------------------------|-----|------------------------------|-----|-----------------------------|-----|----------------------------------|-----|--------------------------------|-----|-----------------------------|-----|------------------------|-----|--------------------------------|-----|-------------------------|-----|------------------------------|--------|----------|
| Bit | Definition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b0 | TEMPERATURE_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b1 | RX_GAIN_PHASE_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b2 | RX_NOISE_FIGURE_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3 | RX_IFSTAGE_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b4 | TX0_POWER_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b5 | TX1_POWER_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b6 | TX2_POWER_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | TX0_BALLBREAK_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b8 | TX1_BALLBREAK_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b9 | TX2_BALLBREAK_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b10 | TX_GAIN_PHASE_MISMATCH_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b11 | TX0_BPM_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b12 | TX1_BPM_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b13 | TX2_BPM_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b14 | SYNTH_FREQ_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b15 | EXTERNAL_ANALOG_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b16 | INTERNAL_TX0_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b17 | INTERNAL_TX1_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b18 | INTERNAL_TX2_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b19 | INTERNAL_RX_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b20 | INTERNAL_PMCLKLO_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b21 | INTERNAL_GPADC_SIGNALS_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b22 | PLL_CONTROL_VOLTAGE_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b23 | DCC_CLOCK_FREQ_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b24 | RX_SATURATION_DETECTOR_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b25 | RX_SIG_IMG_BAND_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b26 | RX_MIXER_INPUT_POWER_MONITOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b31:27 | RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Continued on next page

Table 7.3 – continued from previous page

| | | |
|----------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LDO_SC_MONITORING_EN | 4 | <p>If any bit in this field is set to 1, the associated monitors are enabled. There are no reports for these monitors. If there is any fault, the async event AWR_ANALOGFAULT_AE_SB will be sent.</p> <p>Bit Description</p> <p>b0 APLL LDO short circuit monitoring enable 0 – disable, 1 – enable</p> <p>b1 SYNTH VCO LDO short circuit monitoring enable 0 – disable, 1 – enable</p> <p>b2 PA LDO short circuit monitoring enable 0 – disable, 1 – enable</p> <p>b31:3 RESERVED</p> <p>Note: This feature is not supported in DFP 1.x (1st generation devices). Please refer latest DFP release note for more details</p> |
|----------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

7.2 Temperature Monitor

This section contains API SBs that configure the on chip temperature monitors and report the soft results from the monitor. The corresponding monitors are collectively named TEMPERATURE_MONITOR. These monitors observe the temperature near various RF analog and digital modules using temperature sensors and GPADC and compare them against configurable thresholds. The report is sent as an async event [AWR_MONITOR_TEMPERATURE_REPORT_AE_SB](#).

7.2.1 Sub block 0x01C3 – AWR_MONITOR_TEMPERATURE_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to temperature monitoring. Report of this monitoring will be available in the async event [AWR_MONITOR_TEMPERATURE_REPORT_AE_SB](#).

Table 7.4: AWR_MONITOR_TEMPERATURE_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x01C3 |
| SBLKLEN | 2 | Value = 24 |

Continued on next page

Table 7.4 – continued from previous page

| REPORTING_ MODE | 1 | Value | Definition |
|----------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| | | 0 | Report is sent every monitoring period without threshold check |
| | | 1 | Report is send only upon a failure (after checking for thresholds) |
| | | 2 | Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 | |
| ANA_TEMP_ THRESH.MIN | 2 | The temperatures read from near the sensors near the RF analog modules are compared against a minimum threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is outside this (minimum, maximum) range). 1 LSB = 1°C, signed number Valid range: -99°C to 199°C | |
| ANA_TEMP_ THRESH.MAX | 2 | The temperatures read from near the sensors near the RF analog modules are compared against a maximum threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is outside this (minimum, maximum) range). 1 LSB = 1°C, signed number Valid range: -99°C to 199°C | |
| DIG_TEMP_ THRESH.MIN | 2 | The temperatures read from near the sensor near the digital module are compared against a minimum threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is outside this (minimum, maximum) range). 1 LSB = 1°C, signed number Valid range: -99°C to 199°C | |
| DIG_TEMP_ THRESH.MAX | 2 | The temperatures read from near the sensor near the digital module are compared against a maximum threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is outside this (minimum, maximum) range). 1 LSB = 1°C, signed number Valid range: -99°C to 199°C | |
| TEMP_DIFF_ THRESH | 2 | The maximum difference across temperatures read from all the enabled sensors is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if the measured difference exceeds this field). 1 LSB = 1°C, unsigned number Valid range: 0°C to 100°C | |

Continued on next page

Table 7.4 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.3 RX Gain and Phase Monitor

This section contains API SBs that configure the monitors of receiver gain and phase. The corresponding monitors are collectively named `RX_GAIN_PHASE_MONITOR`. The report is sent as an async event [AWR_MONITOR_RX_GAIN_PHASE_REPORT_AE_SB](#).

NOTE:

It is recommended for the user to configure this monitor in verbose mode (Mode 0), so that Host can compute actual RX gain through temperature compensation and detect presence of interference using Noise Power.

In quiet mode, the user may consider programming broad thresholds for Absolute Gain Error, taking into account the temperature variation of reported `RX_GAIN_VALUE`.

7.3.1 Sub block 0x01C4 – AWR_MONITOR_RX_GAIN_PHASE_CONF_SB

This is a monitoring configuration API which the host sends to the AWR device, containing information related to RX gain and phase monitoring.

Table 7.5: AWR_MONITOR_RX_GAIN_PHASE_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01C4 |
| SBLKLEN | 2 | Value = 72 |
| PROFILE_INDX | 1 | This field indicates the profile Index for which this monitoring configuration applies. |

Continued on next page

Table 7.5 – continued from previous page

| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|--------------------------------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------------------------------------------|-----|----|---------------------------------------------------|-----|
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| REPORTING_MODE | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds). It is recommended not to use quiet mode, as Host has to compute actual RX gain and need to monitor Noise power to detect presence of interference.</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds). It is recommended not to use quiet mode, as Host has to compute actual RX gain and need to monitor Noise power to detect presence of interference. | 2 | Report is sent every monitoring period with threshold check | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds). It is recommended not to use quiet mode, as Host has to compute actual RX gain and need to monitor Noise power to detect presence of interference. | | | | | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | |
| TX_SEL | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>TX0 is used for generating loopback signal for RX gain measurement</td> </tr> <tr> <td>1</td> <td>TX1 is used for generating loopback signal for RX gain measurement</td> </tr> </tbody> </table> | Value | Definition | 0 | TX0 is used for generating loopback signal for RX gain measurement | 1 | TX1 is used for generating loopback signal for RX gain measurement | | | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | TX0 is used for generating loopback signal for RX gain measurement | | | | | | | | | | | | | |
| 1 | TX1 is used for generating loopback signal for RX gain measurement | | | | | | | | | | | | | |
| RX_GAIN_ABS_ERROR_THRESH | 2 | <p>The magnitude of difference between the programmed and measured RX gain for each enabled channel at each enabled RF frequency, is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>Before the comparison, the measured gains for each RF and RX are adjusted by subtracting the offset given in the RX_GAIN_MISMATCH_OFFSET_VALUE field</p> <p>1 LSB = 0.1 dB</p> <p>Valid range: 0 to 65535 (0 to 6553dB)</p> | | | | | | | | | | | | |

Continued on next page

Table 7.5 – continued from previous page

| | | |
|-------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RX_GAIN_MISMATCH_THRESH | 2 | <p>The magnitude of difference between measured RX gains across the enabled channels at each enabled RF frequency is compared against this threshold.</p> <p>The comparison result is part of the monitoring report message (Error bit is set if the measurement is above this threshold).</p> <p>Before the comparison, the measured gains for each RF and RX are adjusted by subtracting the offset given in the RX_GAIN_MISMATCH_OFFSET_VALUE field.</p> <p>1 LSB = 0.1 dB</p> <p>Valid range: 0 to 65535 (0 to 6553dB)</p> |
| RX_GAIN_FLATNESS_ERROR_THRESH | 2 | <p>The magnitude of measured RX gain flatness error, for each enabled channel, is compared against this threshold. The flatness error for a channel is defined as the peak to peak variation across RF frequencies. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>Before the comparison, the measured gains for each RF and RX are adjusted by subtracting the offset given in the RX_GAIN_MISMATCH_OFFSET_VALUE field.</p> <p>1 LSB = 0.1 dB</p> <p>Valid range: 0 to 65535 (0 to 6553dB)</p> <p>This flatness check is applicable only if multiple RF Frequencies are enabled, i.e., RF_FREQ_BITMASK has bit numbers 0,1,2 set.</p> |
| RX_PHASE_MISMATCH_THRESH | 2 | <p>The magnitude of measured RX phase mismatch across the enabled channels at each enabled RF frequency is compared against this threshold.</p> <p>The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>Before the comparison, the measured phases for each RF and RX are adjusted by subtracting the offset given in the RX_PHASE_MISMATCH_OFFSET_VALUE field.</p> <p>1 LSB = $360^\circ / 2^{16}$.</p> <p>Valid range: corresponding to 0° to 359.9°.</p> |

Continued on next page

Table 7.5 – continued from previous page

| RX_GAIN_MISMATCH_OFFSET_VALUE | 24 | <p>The offsets to be subtracted from the measured RX gain for each RX and RF before the relevant threshold comparisons are given here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>1:0</td> <td>9:8</td> <td>17:16</td> </tr> <tr> <td>RX1</td> <td>3:2</td> <td>11:10</td> <td>19:18</td> </tr> <tr> <td>RX2</td> <td>5:4</td> <td>13:12</td> <td>21:20</td> </tr> <tr> <td>RX3</td> <td>7:6</td> <td>15:14</td> <td>23:22</td> </tr> </tbody> </table> <p>1 LSB = 0.1 dB, signed number</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are considered.</p> | | RF1 | RF2 | RF3 | RX0 | 1:0 | 9:8 | 17:16 | RX1 | 3:2 | 11:10 | 19:18 | RX2 | 5:4 | 13:12 | 21:20 | RX3 | 7:6 | 15:14 | 23:22 |
|--------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-------|-------|-----|-----|-------|-------|-----|-----|-------|-------|
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | 1:0 | 9:8 | 17:16 | | | | | | | | | | | | | | | | | | | |
| RX1 | 3:2 | 11:10 | 19:18 | | | | | | | | | | | | | | | | | | | |
| RX2 | 5:4 | 13:12 | 21:20 | | | | | | | | | | | | | | | | | | | |
| RX3 | 7:6 | 15:14 | 23:22 | | | | | | | | | | | | | | | | | | | |
| RX_PHASE_MISMATCH_OFFSET_VALUE | 24 | <p>The offsets to be subtracted from the measured RX phase for each RX and RF before the relevant threshold comparisons are given here.</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>RX0</td> <td>1:0</td> <td>9:8</td> <td>17:16</td> </tr> <tr> <td>RX1</td> <td>3:2</td> <td>11:10</td> <td>19:18</td> </tr> <tr> <td>RX2</td> <td>5:4</td> <td>13:12</td> <td>21:20</td> </tr> <tr> <td>RX3</td> <td>7:6</td> <td>15:14</td> <td>23:22</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ / 2^{16}$, unsigned number</p> <p>Only the entries of enabled RF Frequencies and enabled RX channels are considered.</p> | | RF1 | RF2 | RF3 | RX0 | 1:0 | 9:8 | 17:16 | RX1 | 3:2 | 11:10 | 19:18 | RX2 | 5:4 | 13:12 | 21:20 | RX3 | 7:6 | 15:14 | 23:22 |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | | | | | |
| RX0 | 1:0 | 9:8 | 17:16 | | | | | | | | | | | | | | | | | | | |
| RX1 | 3:2 | 11:10 | 19:18 | | | | | | | | | | | | | | | | | | | |
| RX2 | 5:4 | 13:12 | 21:20 | | | | | | | | | | | | | | | | | | | |
| RX3 | 7:6 | 15:14 | 23:22 | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | | | | | |

7.4 RX Noise Monitor

This section contains API SBs that configure the monitor of receiver noise, and report the soft results from the monitor. The corresponding monitor is named RX_NOISE_FIGURE_MONITOR. The report is sent as an async event [AWR_MONITOR_RX_NOISE_FIGURE_REPORT_AE_SB](#).

7.4.1 Sub block 0x01C5 – AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB

This is a monitoring configuration API which the host sends to the AWR device, containing information related to RX noise monitoring of a profile.

NOTE: The RX Noise figure monitor API is not supported in this release. Please refer latest DFP release note for more info.

Table 7.6: AWR_MONITOR_RX_NOISE_FIGURE_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|---------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|----------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------|----|-------------------------------------------------------------|-----|----|---------------------------------------------------|-----|
| SBLKID | 2 | Value = 0x01C5 | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 16 | | | | | | | | | | | | |
| PROFILE.INDX | 1 | This field indicates the profile Index for which this monitoring configuration applies. | | | | | | | | | | | | |
| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the exact RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="0"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | |
| RX_NOISE_FIGURE_THRESHOLD | 2 | <p>The measured RX input referred noise figure at the enabled RF frequencies, for all channels, is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB)</p> | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | |

NOTE: The Rx gain and phase monitoring shall be enabled when enabling Rx noise figure Monitoring. This monitors only baseband noise figure.

7.5 RX IF Stage Monitor

This section contains API SBs that configure the monitors of receiver IF filter attenuation, and report the soft results from the monitor. The corresponding monitor is named RX_IFSTAGE_MONITOR. The report is sent as an async event [AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB](#).

7.5.1 Sub block 0x01C6 – AWR_MONITOR_RX_IFSTAGE_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to RX IF filter attenuation monitoring. The report is sent as as an async event [AWR_MONITOR_RX_IFSTAGE_REPORT_AE_SB](#).

Table 7.7: AWR_MONITOR_RX_IFSTAGE_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|------------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------------------------------------------------------------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01C6 | | | | | | | | |
| SBLKLEN | 2 | Value = 20 | | | | | | | | |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |
| HPF_CUTOFF_FREQ_ERROR_THRESH | 2 | <p>The absolute values of RX IF HPF cutoff percentage frequency errors are compared against the corresponding thresholds given in this field. The comparison results are part of the monitoring report message (Error bit is set if the absolute value of the errors exceeds respective thresholds).</p> <p>1 LSB = 1%, unsigned number Valid range: 1% to 128%</p> | | | | | | | | |

Continued on next page

Table 7.7 – continued from previous page

| | | |
|------------------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LPF_CUTOFF_FREQ_ERROR_THRESH | 2 | The absolute values of RX IF LPF cutoff percentage frequency errors are compared against the corresponding thresholds given in this field. The comparison results are part of the monitoring report message (Error bit is set if the absolute value of the errors exceeds respective thresholds). 1 LSB = 1%, unsigned number Valid range: 1% to 128% Note: This feature is not supported in this release. Please refer latest DFP release note for more details. |
| IFA_GAIN_ERROR_THRESH | 2 | The absolute deviation of RX IFA Gain from the expected gain for each enabled RX channel is compared against the thresholds given in this field. The comparison result is part of the monitoring report message (Error bit is set if the absolute value of the errors exceeds respective thresholds). 1 LSB = 0.1 dB, unsigned number Valid range: 0 to 65535 (0 to 6553dB) |
| RESERVED | 4 | 0x00000000 |

7.6 TX Power Monitor

This section contains API SBs that configure the monitors of transmitter output power, and report the soft results from the monitor. The corresponding monitors are collectively named TXn_POWER_MONITOR where n is the TX channel number.

7.6.1 Sub block 0x01C7 – AWR_MONITOR_TX0_POWER_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX0 power monitoring. Absolute TX power and flatness across RF frequencies are monitored here. The report is sent as an async event [AWR_MONITOR_TX0_POWER_REPORT_AE_SB](#).

Table 7.8: AWR_MONITOR_TX0_POWER_CONF_SB contents

| Field Name | Number of bytes | Description |
|---------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01C7 |
| SBLKLEN | 2 | Value = 20 |
| PROFILE_INDXX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. |

Continued on next page

Table 7.8 – continued from previous page

| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the exact RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF Name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
|--------------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|----------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------|----|-------------------------------------------------------------|-----|----|---------------------------------------------------|-----|
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |
| REPORTING_MODE | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | |
| TX_POWER_ABSOLUTE_ERROR.THRESH | 2 | <p>The magnitude of difference between the programmed and measured TX power for each enabled channel at each enabled RF frequency, is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB)</p> | | | | | | | | | | | | |
| TX_POWER_FLATNESS_ERROR.THRESH | 2 | <p>The magnitude of measured TX power flatness error, for each enabled channel, is compared against this threshold. The flatness error for a channel is defined as the peak to peak variation across RF frequencies. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB)</p> <p>This flatness check is applicable only if multiple RF Frequencies are enabled.</p> | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |

Continued on next page

Table 7.8 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
|----------|---|------------|

7.6.2 Sub block 0x01C8 – AWR_MONITOR_TX1_POWER_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX1 power monitoring. Absolute TX power and flatness across RF frequencies are monitored here. The report is sent as an async event [AWR_MONITOR_TX1_POWER_REPORT_AE_SB](#).

Table 7.9: AWR_MONITOR_TX1_POWER_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|----------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------|----|-------------------------------------------------------------|-----|----|---------------------------------------------------|-----|
| SBLKID | 2 | Value = 0x01C8 | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 20 | | | | | | | | | | | | |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. | | | | | | | | | | | | |
| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the exact RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF Name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |
| REPORTING_MODE | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | |

Continued on next page

Table 7.9 – continued from previous page

| | | |
|-----------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RESERVED | 1 | 0x00 |
| TX_POWER_ABSOLUTE_ERROR_THRESHOLD | 2 | The magnitude of difference between the programmed and measured TX power for each enabled channel at each enabled RF frequency, is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold). 1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB) |
| TX_POWER_FLATNESS_ERROR_THRESHOLD | 2 | The magnitude of measured TX power flatness error, for each enabled channel, is compared against this threshold. The flatness error for a channel is defined as the peak to peak variation across RF frequencies. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold). 1 LSB = 0.1 dB Valid range:0 to 65535 (0 to 6553dB) This flatness check is applicable only if multiple RF Frequencies are enabled. |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |

7.6.3 Sub block 0x01C9 – AWR_MONITOR_TX2_POWER_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX2 power monitoring. Absolute TX power and flatness across RF frequencies are monitored here. The report is sent as an async event [AWR_MONITOR_TX2_POWER_REPORT_AE_SB](#).

Table 7.10: AWR_MONITOR_TX2_POWER_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01C9 |
| SBLKLEN | 2 | Value = 20 |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. |

Continued on next page

Table 7.10 – continued from previous page

| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the exact RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF Name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
|--------------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|----------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------|----|-------------------------------------------------------------|-----|----|---------------------------------------------------|-----|
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |
| REPORTING_MODE | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check | | | | |
| Value | Definition | | | | | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | |
| TX_POWER_ABSOLUTE_ERROR.THRESH | 2 | <p>The magnitude of difference between the programmed and measured TX power for each enabled channel at each enabled RF frequency, is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB)</p> | | | | | | | | | | | | |
| TX_POWER_FLATNESS_ERROR.THRESH | 2 | <p>The magnitude of measured TX power flatness error, for each enabled channel, is compared against this threshold. The flatness error for a channel is defined as the peak to peak variation across RF frequencies. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB)</p> <p>This flatness check is applicable only if multiple RF Frequencies are enabled.</p> | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | |

Continued on next page

Table 7.10 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
|----------|---|------------|

7.7 TX Ball Break Monitor

This section contains API SBs that configure the monitors of transmitter balls and impedance matching. The corresponding monitors are collectively named TXn_BALLBREAK_MONITOR where n is the TX channel number.

TX ball break detection is performed through measurement of TX reflection coefficient's magnitude. The breakage of a TX ball is detected by observing high reflection magnitude.

7.7.1 Sub block 0x01CA – AWR_MONITOR_TX0_BALLBREAK_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX ball break detection.

This API SB controls the thresholds for the reflection coefficient magnitude check and the parameters for the reflection coefficient error distance check (variation from values at the time of factory calibration). The report is sent as an async event [AWR_MONITOR_TX0_BALLBREAK_REPORT_AE_SB](#).

Table 7.11: AWR_MONITOR_TX0_BALLBREAK_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01CA |
| SBLKLEN | 2 | Value = 16 |
| REPORTING_MODE | 1 | Value Definition 0 Report is sent every monitoring period without threshold check 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 |
| TX_REFL_COEFF_THRESH | 2 | The TX reflection coefficient's magnitude for each enabled channel is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = 0.1 dB, signed number Valid range: -32767 to +32767 (-3276dB to +3276dB) |

Continued on next page

Table 7.11 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.7.2 Sub block 0x01CB – AWR_MONITOR_TX1_BALLBREAK_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX ball break detection.

This API SB controls the thresholds for the reflection coefficient magnitude check and the parameters for the reflection coefficient error distance check (variation from values at the time of factory calibration). The report is sent as an async event [AWR_MONITOR_TX1_BALLBREAK_REPORT_AE_SB](#).

Table 7.12: AWR_MONITOR_TX1_BALLBREAK_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01CB |
| SBLKLEN | 2 | Value = 16 |
| REPORTING_MODE | 1 | Value Definition 0 Report is sent every monitoring period without threshold check 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 |
| TX_REFL_COEFF_THRESH | 2 | The TX reflection coefficient's magnitude for each enabled channel is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = 0.1 dB, signed number Valid range: -32767 to +32767 (-3276dB to +3276dB) |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.7.3 Sub block 0x01CC – AWR_MONITOR_TX2_BALLBREAK_CONF_SB

This API is a monitoring monfiguration API which the host sends to the AWR device, containing information related to TX ball break detection.

This API SB controls the thresholds for the reflection coefficient magnitude check and the parameters for the reflection coefficient error distance check (variation from values at the time of factory calibration). The report is sent as an async event [AWR_MONITOR_TX2_BALLBREAK_REPORT_AE_SB](#).

Table 7.13: AWR_MONITOR_TX2_BALLBREAK_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01CC |
| SBLKLEN | 2 | Value = 16 |
| REPORTING_MODE | 1 | Value Definition 0 Report is sent every monitoring period without threshold check 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 |
| TX_REFL_COEFF_THRESH | 2 | The TX reflection coefficient's magnitude for each enabled channel is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher or equal to this threshold, with the units of both quantities being the same). 1 LSB = 0.1 dB, signed number Valid range: -32767 to +32767 (-3276dB to +3276dB) |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.8 TX Gain and Phase Mismatch Monitoring

This section contains API SBs that configure the monitors of transmitter gain and phase mismatches, and report the soft results from the monitor. The corresponding monitors are collectively named TX_GAIN_PHASE_MISMATCH_MONITOR.

This monitor needs the operation of at least one RX channel. It also needs to use the RX in complex mode. Therefore, if all channels are disabled as per AWR_CHAN_CONF_SET_SB, this monitor automatically enables one RX channel. Further, this monitor automatically uses both I and Q channels of the receiver, irrespective of the ADC settings given by AWR_ADCOUT_CONF_SET_SB.

NOTE: The TX gain and phase mismatch monitor API is not supported in this release. Please refer latest DFP release note for more info.

7.8.1 Sub block 0x01CD – AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX gain and phase mismatch monitoring. The report is sent as an async event [AWR_MONITOR_TX_GAIN_PHASE_REPORT_AE_SB](#).

Table 7.14: AWR_MONITOR_TX_GAIN_PHASE_MISMATCH_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | | | | | |
|------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------|---------|-----|--------------------------------------------------|-----|----|--------------------------------------------------|-----|----|---------------------------------------------------|-----|
| SBLKID | 2 | Value = 0x01CD | | | | | | | | | | | | |
| SBLKLEN | 2 | Value = 56 | | | | | | | | | | | | |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. | | | | | | | | | | | | |
| RF_FREQ_BIT-MASK | 1 | <p>This field indicates the exact RF frequencies inside the profile's RF band at which to measure the required parameters. When each bit in this field is set, the measurement at the corresponding RF frequency is enabled w.r.t. the profile's RF band.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>RF frequency</th> <th>RF name</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>Lowest RF frequency in profile's sweep bandwidth</td> <td>RF1</td> </tr> <tr> <td>b1</td> <td>Center RF frequency in profile's sweep bandwidth</td> <td>RF2</td> </tr> <tr> <td>b2</td> <td>Highest RF frequency in profile's sweep bandwidth</td> <td>RF3</td> </tr> </tbody> </table> <p>The RF Name column is mentioned here to set the convention for the purpose of reporting and describing many monitoring packets.</p> | Bit number | RF frequency | RF name | b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | b1 | Center RF frequency in profile's sweep bandwidth | RF2 | b2 | Highest RF frequency in profile's sweep bandwidth | RF3 |
| Bit number | RF frequency | RF name | | | | | | | | | | | | |
| b0 | Lowest RF frequency in profile's sweep bandwidth | RF1 | | | | | | | | | | | | |
| b1 | Center RF frequency in profile's sweep bandwidth | RF2 | | | | | | | | | | | | |
| b2 | Highest RF frequency in profile's sweep bandwidth | RF3 | | | | | | | | | | | | |
| TX_EN | 1 | <p>This field indicates the TX channels that should be compared for gain and phase balance. Setting the corresponding bit to 1 enables that channel for imbalance measurement.</p> <table border="1"> <thead> <tr> <th>Bit number</th> <th>TX Channel</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>TX0</td> </tr> <tr> <td>b1</td> <td>TX1</td> </tr> <tr> <td>b2</td> <td>TX2</td> </tr> </tbody> </table> | Bit number | TX Channel | b0 | TX0 | b1 | TX1 | b2 | TX2 | | | | |
| Bit number | TX Channel | | | | | | | | | | | | | |
| b0 | TX0 | | | | | | | | | | | | | |
| b1 | TX1 | | | | | | | | | | | | | |
| b2 | TX2 | | | | | | | | | | | | | |

Continued on next page

Table 7.14 – continued from previous page

| RX_EN | 1 | <p>This field indicates the RX channels that should be enabled for TX to RX loopback measurement. Setting the corresponding bit to 1 enables that channel for imbalance measurement.</p> <p>Bit number RX Channel</p> <p>b0 RX0</p> <p>b1 RX1</p> <p>b2 RX2</p> <p>b3 RX3</p> | | | | | | | | |
|--------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------------------------------------------------------------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| REPORTING_MODE | 1 | <table border="1"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | |
| TX_GAIN_MISMATCH_THRESH | 2 | <p>The magnitude of difference between measured TX powers across the enabled channels at each enabled RF frequency is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if the measurement is above this threshold).</p> <p>Before the comparison, the measured gains for each RF and RX are adjusted by subtracting the offset given in the TX_GAIN_MISMATCH_OFFSET_VALUE field.</p> <p>1 LSB = 0.1 dB, unsigned number Valid range: 0 to 65535 (0 to 6553dB)</p> | | | | | | | | |
| TX_PHASE_MISMATCH_THRESH | 2 | <p>The magnitude of measured TX phase mismatch across the enabled channels at each enabled RF frequency is compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if any measurement is above this threshold).</p> <p>Before the comparison, the measured gains for each RF and RX are adjusted by subtracting the offset given in the TX_PHASE_MISMATCH_OFFSET_VALUE field.</p> <p>1 LSB = $360^\circ / 2^{16}$, unsigned number Valid range: corresponding to 0° to 359.9°.</p> | | | | | | | | |

Continued on next page

Table 7.14 – continued from previous page

| TX_GAIN_MISMATCH_OFFSET_VALUE | 18 | <p>The offsets to be subtracted from the measured TX gain for each TX and RF before the relevant threshold comparisons are given here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>TX0</td> <td>1:0</td> <td>7:6</td> <td>13:12</td> </tr> <tr> <td>TX1</td> <td>3:2</td> <td>9:8</td> <td>15:14</td> </tr> <tr> <td>TX2</td> <td>5:4</td> <td>11:10</td> <td>17:16</td> </tr> </tbody> </table> <p>Only the entries of enabled RF Frequencies and enabled TX channels are considered.</p> | | RF1 | RF2 | RF3 | TX0 | 1:0 | 7:6 | 13:12 | TX1 | 3:2 | 9:8 | 15:14 | TX2 | 5:4 | 11:10 | 17:16 |
|--------------------------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|-------|-----|-----|-------|-------|
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | |
| TX0 | 1:0 | 7:6 | 13:12 | | | | | | | | | | | | | | | |
| TX1 | 3:2 | 9:8 | 15:14 | | | | | | | | | | | | | | | |
| TX2 | 5:4 | 11:10 | 17:16 | | | | | | | | | | | | | | | |
| TX_PHASE_MISMATCH_OFFSET_VALUE | 18 | <p>The offsets to be subtracted from the measured TX phase for each TX and RF before the relevant threshold comparisons are given here.</p> <p>Byte numbers corresponding to different RX and RF, in this field are here:</p> <table border="1"> <thead> <tr> <th></th> <th>RF1</th> <th>RF2</th> <th>RF3</th> </tr> </thead> <tbody> <tr> <td>TX0</td> <td>1:0</td> <td>7:6</td> <td>13:12</td> </tr> <tr> <td>TX1</td> <td>3:2</td> <td>9:8</td> <td>15:14</td> </tr> <tr> <td>TX2</td> <td>5:4</td> <td>11:10</td> <td>17:16</td> </tr> </tbody> </table> <p>1 LSB = $360^\circ / 2^{16}$.</p> <p>Only the entries of enabled RF Frequencies and enabled TX channels are considered.</p> | | RF1 | RF2 | RF3 | TX0 | 1:0 | 7:6 | 13:12 | TX1 | 3:2 | 9:8 | 15:14 | TX2 | 5:4 | 11:10 | 17:16 |
| | RF1 | RF2 | RF3 | | | | | | | | | | | | | | | |
| TX0 | 1:0 | 7:6 | 13:12 | | | | | | | | | | | | | | | |
| TX1 | 3:2 | 9:8 | 15:14 | | | | | | | | | | | | | | | |
| TX2 | 5:4 | 11:10 | 17:16 | | | | | | | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | |

NOTE: Even when the TXs are matched, TX3 loopback path has gain and phase offsets wrt TX1 (and TX2), which get reported as mismatches in this API. These deterministic offsets can be compensated either through the OFFSET_VALUE fields (quiet mode) or through post processing by the host (verbose mode). Nominally, when the TXs are matched, TX3 - TX1 gain (i.e. loopback amplitude) is reported as -8dB. Nominally, when the TXs are matched, the reported TX3 - TX1 phase difference varies linearly with RF and it is reported as -5degree (76GHz) and 15degree (81GHz).

7.9 TX BPM Phase Monitor

This section contains API SBs that configure the monitors of transmitter binary phase modulation, and report the soft results from the monitor, for various TX channels. The corresponding monitors are collectively named TX0_BPM_MONITOR, TX1_BPM_MONITOR and TX2_BPM_MONITOR for

the respective TX channels.

NOTE: The TX BPM monitor APIs are not supported in this release. Please refer latest DFP release note for more info.

7.9.1 Sub block 0x01CE – AWR_MONITOR_TX0_BPM_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX0 BPM monitoring.

The report is sent as an async event [AWR_MONITOR_TX0_BPM_REPORT_AE_SB](#).

Table 7.15: AWR_MONITOR_TX0_BPM_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|--------------------|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|----|----------------------------------------------------------------|----|--------------------------------------------------------------------|------|-------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01CE | | | | | | | | |
| SBLKLEN | 2 | Value = 20 | | | | | | | | |
| PROFILE_INDXX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. | | | | | | | | |
| PH_SHIFTER_MON_CFG | 1 | <table border="0"> <tr> <td>Bit</td> <td>Definition</td> </tr> <tr> <td>b7</td> <td>Phase shifter monitoring enabled</td> </tr> <tr> <td>b6</td> <td>Phase shifter monitoring increment enabled</td> </tr> <tr> <td>b5:0</td> <td>Phase shifter monitoring increment value 1 LSB = 5.625°</td> </tr> </table> <p>NOTE: Phase shifter monitoring control is applicable only in devices which support phase shifter (refer data sheet). For other devices, this is a RESERVED bit and should be set to 0.</p> | Bit | Definition | b7 | Phase shifter monitoring enabled | b6 | Phase shifter monitoring increment enabled | b5:0 | Phase shifter monitoring increment value 1 LSB = 5.625° |
| Bit | Definition | | | | | | | | | |
| b7 | Phase shifter monitoring enabled | | | | | | | | | |
| b6 | Phase shifter monitoring increment enabled | | | | | | | | | |
| b5:0 | Phase shifter monitoring increment value 1 LSB = 5.625° | | | | | | | | | |
| PH_SHIFTER_MON1 | 1 | Phase1 of the phase shifter of TX0 which needs to be monitored 1 LSB = 5.625° | | | | | | | | |
| PH_SHIFTER_MON2 | 1 | Phase2 of the phase shifter of TX0 which needs to be monitored 1 LSB = 5.625° | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |

Continued on next page

Table 7.15 – continued from previous page

| | | |
|-------------------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RX_EN | 1 | <p>This field indicates the RX channels that should be enabled for TX to RX loopback measurement. Setting the corresponding bit to 1 enables that channel for imbalance measurement.</p> <p>Bit number RX Channel</p> <p>b0 RX0</p> <p>b1 RX1</p> <p>b2 RX2</p> <p>b3 RX3</p> |
| TX_BPM_PHASE_ERROR_THRESH | 2 | <p>The deviation of the TX output phase difference between the two BPM settings from the ideal 180° is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same).</p> <p>1 LSB = $360^\circ / 2^{16}$.</p> <p>Valid range: corresponding to 0° to 359.9°.</p> |
| TX_BPM_AMPLITUDE_ERROR_THRESH | 2 | <p>The deviation of the TX output amplitude difference between the two BPM settings is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same).</p> <p>1 LSB = 0.1 dB</p> <p>Valid range: 0 to 65535 (0 to 6553dB)</p> |
| PH_SHIFTER_THRESH.MAX | 2 | <p>Maximum threshold for the difference in the 2 configured phase shift values</p> <p>1 LSB = 5.625°</p> |
| PH_SHIFTER_THRESH.MIN | 2 | <p>Minimum threshold for the difference in the 2 configured phase shift values</p> <p>1 LSB = 5.625°</p> |
| RESERVED | 2 | 0x0000 |

7.9.2 Sub block 0x01CF – AWR_MONITOR_TX1_BPM_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX1 BPM monitoring. The report is sent as an async event [AWR_MONITOR_TX1_BPM_REPORT_AE_SB](#).

Table 7.16: AWR_MONITOR_TX1_BPM_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01CF |
| SBLKLEN | 2 | Value = 20 |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. |
| PH_SHIFTER_MON_CFG | 1 | Bit Definition b7 Phase shifter monitoring enabled b6 Phase shifter monitoring increment enabled b5:0 Phase shifter monitoring increment value 1 LSB = 5.625° NOTE: Phase shifter monitoring control is applicable only in devices which support phase shifter (refer data sheet). For other devices, this is a RESERVED bit and should be set to 0. |
| PH_SHIFTER_MON1 | 1 | Phase1 of the phase shifter of TX1 which needs to be monitored 1 LSB = 5.625° |
| PH_SHIFTER_MON2 | 1 | Phase2 of the phase shifter of TX1 which needs to be monitored 1 LSB = 5.625° |
| REPORTING_MODE | 1 | Value Definition 0 Report is sent every monitoring period without threshold check 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RX_EN | 1 | This field indicates the RX channels that should be enabled for TX to RX loopback measurement. Setting the corresponding bit to 1 enables that channel for imbalance measurement. Bit number RX Channel b0 RX0 b1 RX1 b2 RX2 b3 RX3 |

Continued on next page

Table 7.16 – continued from previous page

| | | |
|-------------------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX_BPM_PHASE_ERROR_THRESH | 2 | The deviation of the TX output phase difference between the two BPM settings from the ideal 180° is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = $360^\circ / 2^{16}$. Valid range: corresponding to 0° to 359.9°. |
| TX_BPM_AMPLITUDE_ERROR_THRESH | 2 | The deviation of the TX output amplitude difference between the two BPM settings is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB) |
| PH_SHIFTER_THRESH.MAX | 2 | Maximum threshold for the difference in the 2 configured phase shift values 1 LSB = 5.625° |
| PH_SHIFTER_THRESH.MIN | 2 | Minimum threshold for the difference in the 2 configured phase shift values 1 LSB = 5.625° |
| RESERVED | 2 | 0x0000 |

7.9.3 Sub block 0x01D0 – AWR_MONITOR_TX2_BPM_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX2 BPM monitoring. The report is sent as an async event [AWR_MONITOR_TX2_BPM_REPORT_AE_SB](#).

Table 7.17: AWR_MONITOR_TX2_BPM_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D0 |
| SBLKLEN | 2 | Value = 20 |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. |

Continued on next page

Table 7.17 – continued from previous page

| PH_SHIFTER_MON_CFG | 1 | Bit Definition b7 Phase shifter monitoring enabled b6 Phase shifter monitoring increment enabled b5:0 Phase shifter monitoring increment value 1 LSB = 5.625° NOTE: Phase shifter monitoring control is applicable only in devices which support phase shifter (refer data sheet). For other devices, this is a RESERVED bit and should be set to 0. | | | | | | | | |
|---------------------------|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------------------------------------------------------------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| PH_SHIFTER_MON1 | 1 | Phase1 of the phase shifter of TX2 which needs to be monitored 1 LSB = 5.625° | | | | | | | | |
| PH_SHIFTER_MON2 | 1 | Phase2 of the phase shifter of TX2 which needs to be monitored 1 LSB = 5.625° | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <thead> <tr> <th>Value</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </tbody> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| RX_EN | 1 | This field indicates the RX channels that should be enabled for TX to RX loopback measurement. Setting the corresponding bit to 1 enables that channel for imbalance measurement. Bit number RX Channel b0 RX0 b1 RX1 b2 RX2 b3 RX3 | | | | | | | | |
| TX_BPM_PHASE_ERROR_THRESH | 2 | The deviation of the TX output phase difference between the two BPM settings from the ideal 180° is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = $360^\circ / 2^{16}$. Valid range: corresponding to 0° to 359.9°. | | | | | | | | |

Continued on next page

Table 7.17 – continued from previous page

| | | |
|-------------------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TX_BPM_AMPLITUDE_ERROR_THRESH | 2 | The deviation of the TX output amplitude difference between the two BPM settings is compared against the threshold given here. The comparison result is part of the monitoring report message (Error bit is set if the measurement is higher than this threshold, with the units of both quantities being the same). 1 LSB = 0.1 dB Valid range: 0 to 65535 (0 to 6553dB) |
| PH_SHIFTER_THRESH_MAX | 2 | Maximum threshold for the difference in the 2 configured phase shift values 1 LSB = 5.625° |
| PH_SHIFTER_THRESH_MIN | 2 | Minimum threshold for the difference in the 2 configured phase shift values 1 LSB = 5.625° |
| RESERVED | 2 | 0x0000 |

7.10 Synthesizer Frequency Monitoring

This section contains API SBs that configure the monitors of synthesizer chirp frequency, and report the soft results from the monitor. The corresponding monitor is named SYNTH_FREQ_MONITOR.

7.10.1 Sub block 0x01D1 – AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to synthesizer frequency monitoring during chirping. The report is sent as an async event [AWR_MONITOR_SYNTH_FREQ_REPORT_AE_SB](#).

Table 7.18: AWR_MONITOR_SYNTHESIZER_FREQUENCY_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D1 |
| SBLKLEN | 2 | Value = 16 |
| PROFILE.INDX | 1 | This field indicates the Profile Index for which this monitoring configuration applies. |

Continued on next page

Table 7.18 – continued from previous page

| REPORTING_ MODE | 1 | Value | Definition |
|---------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| | | 0 | Report is sent every monitoring period without threshold check |
| | | 1 | Report is send only upon a failure (after checking for thresholds) |
| | | 2 | Report is sent every monitoring period with threshold check |
| FREQ.ERROR_ THRESH | 2 | During the chirp, the error of the measured instantaneous chirp frequency w.r.t. the desired value is continuously compared against this threshold. The comparison result is part of the monitoring report message (Error bit is set if the measurement is above this threshold, ever during the previous monitoring period). 1 LSB = 10 kHz Valid range: 0 to 65535 (0 to 655MHz) | |
| MONITOR_ START_TIME | 1 | This field determines when the monitoring starts in each chirp relative to the start of the ramp. 1 LSB = 0.2 μ s, signed number Valid range: -25 to 25 μ s | |
| RESERVED | 3 | 0x000000 | |
| RESERVED | 4 | 0x00000000 | |

NOTE:

It is recommended to re-issue this configuration API each time before enabling this monitor and frame trigger. The right sequence is as below:

1. Issue Synth frequency monitor configuration API.
2. Enable Synth frequency monitor.
3. Frame start.
4. Frame stop.
5. Frame start. (Optional in case of multiple frames)
6. Frame stop. (Optional in case of multiple frames)
7. Disable Synth frequency monitor (in case disabled for some reason)
8. Issue Synth frequency monitor configuration API.
9. Enable Synth frequency monitor.
10. Frame start.

7.11 External Analog Signals Monitor

This section contains API SBs that configure the monitors of external analog signals which are input to the device through pins ANALOGTEST1-4, ANAMUX and VSENSE (also called ADC1-6)

and report the soft results from the monitor. The corresponding monitors are collectively named EXTERNAL_ANALOG_SIGNALS_MONITOR. These monitors observe various analog signals input on the pins ADC1-6 using a GPADC and compare them against internally fixed thresholds.

7.11.1 Sub block 0x01D2 – AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to external DC signals monitoring (available only in xWR1642/IWR6843/xWR1843). The report is sent as an async event [AWR_MONITOR_EXTERNAL_ANALOG_SIGNALSREPORT_AE_SB](#).

Table 7.19 describes the content of this sub block.

Table 7.19: AWR_MONITOR_EXTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D2 |
| SBLKLEN | 2 | Value = 36 |
| REPORTING_MODE | 1 | Value Definition 0 Report is sent every monitoring period without threshold check 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 7.19 – continued from previous page

| | | |
|-----------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SIGNAL_INPUT_ENABLES | 1 | <p>This field indicates the sets of externally fed DC signals which are to be monitored using GPADC. When each bit in this field is set, the corresponding signal is monitored. The monitored signals are compared against programmed limits. The comparison result is part of the monitoring report message.</p> <p>Bit Location SIGNAL</p> <p>b0 ANALOGTEST1</p> <p>b1 ANALOGTEST2</p> <p>b2 ANALOGTEST3</p> <p>b3 ANALOGTEST4</p> <p>b4 ANAMUX</p> <p>b5 VSENSE</p> <p>Others RESERVED</p> |
| SIGNAL_BUFFER_ENABLES | 1 | <p>This field indicates the sets of externally fed DC signals which are to be buffered before being fed to the GPADC. When each bit in this field is set, the corresponding signal is buffered before the GPADC. The monitored signals are compared against programmed limits. The comparison result is part of the monitoring report message.</p> <p>Bit SIGNAL</p> <p>b0 ANALOGTEST1</p> <p>b1 ANALOGTEST2</p> <p>b2 ANALOGTEST3</p> <p>b3 ANALOGTEST4</p> <p>b4 ANAMUX</p> <p>Others RESERVED</p> |

Continued on next page

Table 7.19 – continued from previous page

| SIGNAL_SETTLING_TIME | 6 | <p>After connecting an external signal to the GPADC, the amount of time to wait for it to settle before taking GPADC samples is programmed in this field. For each signal, after that settling time, GPADC measurements take place for 6.4 μs (averaging 4 samples of the GPADC output). The byte locations of the settling times for each signal are tabulated here:</p> <table data-bbox="662 582 941 929"> <thead> <tr> <th>Byte Location</th> <th>SIGNAL</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ANALOGTEST1</td> </tr> <tr> <td>1</td> <td>ANALOGTEST2</td> </tr> <tr> <td>2</td> <td>ANALOGTEST3</td> </tr> <tr> <td>3</td> <td>ANALOGTEST4</td> </tr> <tr> <td>4</td> <td>ANAMUX</td> </tr> <tr> <td>5</td> <td>VSENSE</td> </tr> </tbody> </table> <p>1 LSB = 0.8 μs Valid range: 0 to 12 μs Valid programming condition: all the signals that are enabled should take a total of < 100 μs, including the programmed settling times and a fixed 6.4 μs of measurement time per enabled signal.</p> | Byte Location | SIGNAL | 0 | ANALOGTEST1 | 1 | ANALOGTEST2 | 2 | ANALOGTEST3 | 3 | ANALOGTEST4 | 4 | ANAMUX | 5 | VSENSE |
|----------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------|---|-------------|---|-------------|---|-------------|---|-------------|---|--------|---|--------|
| Byte Location | SIGNAL | | | | | | | | | | | | | | | |
| 0 | ANALOGTEST1 | | | | | | | | | | | | | | | |
| 1 | ANALOGTEST2 | | | | | | | | | | | | | | | |
| 2 | ANALOGTEST3 | | | | | | | | | | | | | | | |
| 3 | ANALOGTEST4 | | | | | | | | | | | | | | | |
| 4 | ANAMUX | | | | | | | | | | | | | | | |
| 5 | VSENSE | | | | | | | | | | | | | | | |

Continued on next page

Table 7.19 – continued from previous page

| | | |
|---------------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SIGNAL_THRESH | 12 | <p>The external DC signals measured on GPADC are compared against these minimum and maximum thresholds. The comparison result is part of the monitoring report message (Error bit is set if any measurement is outside this (minimum, maximum) range).</p> <p>Byte Location Threshold SIGNAL</p> <p>0 Minimum ANALOGTEST1</p> <p>1 Minimum ANALOGTEST2</p> <p>2 Minimum ANALOGTEST3</p> <p>3 Minimum ANALOGTEST4</p> <p>4 Minimum ANAMUX</p> <p>5 Minimum VSENSE</p> <p>6 Maximum ANALOGTEST1</p> <p>7 Maximum ANALOGTEST2</p> <p>8 Maximum ANALOGTEST3</p> <p>9 Maximum ANALOGTEST4</p> <p>10 Maximum ANAMUX</p> <p>11 Maximum VSENSE</p> <p>1 LSB = 1.8V/256</p> <p>Valid range: 0 to 255</p> |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.12 Internal Analog Signals Monitor

This section contains API SBs that configure the monitors of internal analog signals in the RF analog modules and report the soft results from the monitor. The corresponding monitors are collectively named INTERNAL_ANALOG_SIGNALS_MONITOR. These monitors observe various analog nodes in the RF and analog modules using a GPADC and compare them against internally fixed thresholds.

The configuration API SBs are organized to address various analog circuits as follows:

1. TX0 Internal Analog Signals Monitoring
 - a. This monitor is called INTERNAL_TX0_SIGNALS_MONITOR.
 - b. Signal sets that are monitored: (SUPPLY_TX, PWRDET_TX)
2. TX1 Internal Analog Signals Monitoring

- a. This monitor is called INTERNAL_TX1_SIGNALS_MONITOR
 - b. Signal sets that are monitored: (SUPPLY_TX, PWRDET_TX)
3. TX2 Internal Analog Signals Monitoring
 - a. This monitor is called INTERNAL_TX2_SIGNALS_MONITOR
 - b. Signal sets that are monitored: (SUPPLY_TX, PWRDET_TX)
 4. RX Internal Analog Signals Monitoring
 - a. This monitor is called INTERNAL_RX_SIGNALS_MONITOR
 - b. Signal sets that are monitored: (SUPPLY_RX, PWRDET_RX, DCBIAS_RX)
 5. PM CLK LO Internal Analog Signals Monitoring
 - a. This monitor is called INTERNAL_PMCLKLO_SIGNALS_MONITOR
 - b. Signal sets that are monitored: (SUPPLY_PMCLKLO, PWRDET_PMCLKLO, DCBIAS_PMCLKLO)
 6. GPADC Internal Analog Signals Monitoring
 - a. This monitor is called INTERNAL_GPADC_SIGNALS_MONITOR
 - b. Signal sets that are monitored: (GPADC_REF1, GPADC_REF2)

The results are reported in the corresponding REPORT API SBs in this section.

7.12.1 Sub block 0x01D3 – AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX0 Internal Analog Signals monitoring. The report is sent as an async event [AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.20: AWR_MONITOR_TX0_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description |
|---------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D3 |
| SBLKLEN | 2 | Value = 12 |
| PROFILE.INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring the enabled signals, using test chirps (static frequency, at the center of the profile's RF frequency band). |

Continued on next page

Table 7.20 – continued from previous page

| | | | |
|----------------|---|------------|--------------------------------------------------------------------|
| REPORTING_MODE | 1 | Value | Definition |
| | | 0 | RESERVED |
| | | 1 | Report is send only upon a failure (after checking for thresholds) |
| | | 2 | Report is sent every monitoring period with threshold check |
| RESERVED | 2 | 0x0000 | |
| RESERVED | 4 | 0x00000000 | |

7.12.2 Sub block 0x01D4 – AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX1 Internal Analog Signals monitoring. The report is sent as an async event [AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.21: AWR_MONITOR_TX1_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D4 |
| SBLKLEN | 2 | Value = 12 |
| PROFILE_INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring the enabled signals, using test chirps (static frequency, at the center of the profile's RF frequency band). |
| REPORTING_MODE | 1 | Value Definition 0 RESERVED 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 2 | 0x0000 |
| RESERVED | 4 | 0x00000000 |

7.12.3 Sub block 0x01D5 – AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to TX2 Internal Analog Signals monitoring. The report is sent as an async event [AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.22: AWR_MONITOR_TX2_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|----------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D5 | | | | | | | | |
| SBLKLEN | 2 | Value = 12 | | | | | | | | |
| PROFILE.INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring the enabled signals, using test chirps (static frequency, at the center of the profile's RF frequency band). | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>RESERVED</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | RESERVED | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | RESERVED | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | |

7.12.4 Sub block 0x01D6 – AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to RX Internal Analog Signals monitoring. The report is sent as an async event [AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.23: AWR_MONITOR_RX_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x01D6 |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 7.23 – continued from previous page

| | | | | | | | | | | |
|----------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| PROFILE_INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring the enabled signals, using test chirps (static frequency, at the center of the profile's RF frequency band). | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>RESERVED</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | RESERVED | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | RESERVED | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| RESERVED | 2 | 0x0000 | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | |

7.12.5 Sub block 0x01D7 – AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to Power Management, Clock generation and LO distribution circuits' Internal Analog Signals monitoring. The report is sent as an async event [AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.24: AWR_MONITOR_PMCLKLO_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|----------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|---|----------|---|--------------------------------------------------------------------|---|-------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D7 | | | | | | | | |
| SBLKLEN | 2 | Value = 12 | | | | | | | | |
| PROFILE_INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring the enabled signals, using test chirps (static frequency, at the center of the profile's RF frequency band). | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>RESERVED</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | RESERVED | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | RESERVED | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |

Continued on next page

Table 7.24 – continued from previous page

| | | |
|---------------------|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYNC_20G_SIG_SEL | 1 | This field is relevant only in cascade configuration and not applicable in single chip case Value Definition 0x00 20GHz SYNC monitoring disabled 0x01 FMCW_SYNC_IN monitoring enabled 0x02 FMCW_SYNC_OUT monitoring enabled 0x03 FMCW_CLK_OUT monitoring enabled |
| SYNC_20G_MIN_THRESH | 1 | The minimum threshold value of monitoring, signed number Unit: 1 LSB = 1 dBm |
| SYNC_20G_MAX_THRESH | 1 | The maximum threshold value of monitoring, signed number Unit: 1 LSB = 1 dBm |
| RESERVED | 3 | 0x000000 |

7.12.6 Sub block 0x01D8 – AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to GPADC Internal Analog Signals monitoring. During this monitor, only the relevant circuits are ensured to be ON. The monitored signals are compared against internally chosen valid limits. The comparison result is part of the consolidated monitoring report message (Error bit for any signal set is set to 1 if any measurement in that signal set is beyond valid limits). The report is sent as an async event [AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_REPORT_AE_SB](#).

Table 7.25: AWR_MONITOR_GPADC_INTERNAL_ANALOG_SIGNALS_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D8 |
| SBLKLEN | 2 | Value = 12 |
| REPORTING_MODE | 1 | Value Definition 0 RESERVED 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 3 | 0x000000 |

Continued on next page

Table 7.25 – continued from previous page

| | | |
|----------|---|------------|
| RESERVED | 4 | 0x00000000 |
|----------|---|------------|

7.13 PLL Control Voltage Monitor

This section contains API SBs that configure the monitors of APLL and Synthesizer VCO control voltages and report the soft results from the monitor. The corresponding monitors are collectively named PLL_CONTROL_VOLTAGE_MONITOR. These monitors observe the VCO control voltages under various conditions using the GPADC and compare them against internally fixed thresholds. The transmitters are kept in OFF state during these measurements to avoid external emission.

7.13.1 Sub block 0x01D9 – AWR_MONITOR_PLL_CONTROL_VOLTAGE_SIGNALS_CONF_SB

This is a monitoring configuration API which the host sends to the AWR device, containing information related to APLL and Synthesizer's control voltage signals monitoring. The report is sent as an async event [AWR_MONITOR_PLL_CONTROL_VOLTAGE_REPORT_AE_SB](#).

Table 7.26: AWR_MONITOR_PLL_CONTROL_VOLTAGE_CONF_SB contents

| Field Name | Number of bytes | Description |
|----------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01D9 |
| SBLKLEN | 2 | Value = 12 |
| REPORTING_MODE | 1 | Value Definition 0 Reserved 1 Report is send only upon a failure (after checking for thresholds) 2 Report is sent every monitoring period with threshold check |
| RESERVED | 1 | 0x00 |

Continued on next page

Table 7.26 – continued from previous page

| | | |
|---------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SIGNAL_EN- ABLES | 2 | <p>This field indicates the sets of signals which are to be monitored. When each bit in this field is set, the corresponding signal set is monitored using test chirps. Rest of the RF analog may not be ON during these test chirps. The APLL VCO control voltage can be monitored. The Synthesizer VCO control voltage for both VCO1 and VCO2 can be monitored, while operating at their respective minimum and maximum frequencies, and their respective VCO slope (Hz/V) can be monitored if both frequencies are enabled for that VCO. The monitored signals are compared against internally chosen valid limits. The comparison results are part of the monitoring report message.</p> <p>Bit Location SIGNAL b0 APLL_VCTRL b1 SYNTH_VCO1_VCTRL b2 SYNTH_VCO2_VCTRL b15:3 RESERVED The synthesizer VCO extreme frequencies are: Synthesizer VCO Frequency Limits (Min, Max) VCO1 (76GHz, 78GHz) VCO2 (77GHz, 81GHz) Synthesizer measurements are done with TX switched off to avoid emissions.</p> |
| RESERVED | 4 | 0x00000000 |

7.14 Dual Clock Comparator Based Clock Frequency Monitor

This section contains API SBs that configure the Dual Clock Comparator based monitors of clocks in the BSS digital modules and report the soft results from the monitor. The corresponding monitors are collectively named DCC_CLOCK_FREQ_MONITOR. These monitors observe the relative frequency of various clock pairs and compare the measured relative frequency errors against internally fixed thresholds.

The various clock pairs that are monitored are defined here:

| CLOCK PAIR | REFERENCE CLOCK | MEASURED CLOCK | ERROR THRESH- OLD (Tentative) |
|------------|-----------------|----------------|----------------------------------|
|------------|-----------------|----------------|----------------------------------|

| | | | |
|------|----------|-----------|--------|
| 0 | XTAL | BSS_600M | ±1.0% |
| 1 | BSS_600M | BSS_200M | ±1.0% |
| 2 | BSS_600M | BSS_100M | ±1.0% |
| 3 | BSS_600M | GPADC_10M | ±2.5% |
| 4 | BSS_600M | RCOSC_10M | ±30.0% |
| RSVD | RSVD | RSVD | RSVD |

The ideal frequencies of clocks involved in this monitor are given here:

Table 7.27: DCC Clock monitor pairs

| CLOCK NAME | CLOCK FREQUENCY (MHz) | COMMENTS |
|--------------|-----------------------|------------------------------------------------------------------------------------------------------------|
| XTAL | 40 | Crystal clock |
| BSS_600M | 600 | BSS root clock |
| BSS_200M | 200 | BSS processor clock |
| BSS_100M | 100 | BSS internal clock |
| GPADC_10M | 10 | GPADC clock used in monitoring and calibrations |
| RCOSC_10M | 10 (±10%) | RC Oscillator clock |
| RAMPGEN_100M | 100 | Clock for Ramp Generator (timing engine) and Digital Front End. (Not supported, refer latest release note) |

7.14.1 Sub block 0x01DA – AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to the DCC based clock frequency monitoring. The report is sent as an async event [AWR_MONITOR_DCC_DUAL_CLOCK_COMP_REPORT_AE_SB](#).

Table 7.28: AWR_MONITOR_DUAL_CLOCK_COMP_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------|-----------------|----------------|
| SBLKID | 2 | Value = 0x01DA |
| SBLKLEN | 2 | Value = 12 |

Continued on next page

Table 7.28 – continued from previous page

| REPORTING_MODE | 1 | Value | Definition | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|-----|------------|----|------------|----|------------|----|------------|----|-------------|----|-------------|----|--------------------------------|-------|----------|
| | | 0 | RESERVED check | | | | | | | | | | | | | | | | |
| | | 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | | | | | | | | |
| | | 2 | Report is sent every monitoring period with threshold check | | | | | | | | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | | | | | | | | | | |
| DCC_PAIR_ENABLEDS | 2 | <p>This field indicates which pairs of clocks to monitor. When a bit in the field is set to 1, the firmware monitors the corresponding clock pair by deploying the hardware's Dual Clock Comparator in the corresponding DCC mode.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>CLOCK PAIR</th> </tr> </thead> <tbody> <tr> <td>b0</td> <td>0 BSS_600M</td> </tr> <tr> <td>b1</td> <td>1 BSS_200M</td> </tr> <tr> <td>b2</td> <td>2 BSS_100M</td> </tr> <tr> <td>b3</td> <td>3 GPADC_10M</td> </tr> <tr> <td>b4</td> <td>4 RCOSC_10M</td> </tr> <tr> <td>b5</td> <td>5 RAMPGEN_100M (Not supported)</td> </tr> <tr> <td>b15:6</td> <td>RESERVED</td> </tr> </tbody> </table> <p>The comparison results are part of the monitoring report message. The definition of the clock pairs and their error thresholds for failure reporting are given in the table below the message definition.</p> | | Bit | CLOCK PAIR | b0 | 0 BSS_600M | b1 | 1 BSS_200M | b2 | 2 BSS_100M | b3 | 3 GPADC_10M | b4 | 4 RCOSC_10M | b5 | 5 RAMPGEN_100M (Not supported) | b15:6 | RESERVED |
| Bit | CLOCK PAIR | | | | | | | | | | | | | | | | | | |
| b0 | 0 BSS_600M | | | | | | | | | | | | | | | | | | |
| b1 | 1 BSS_200M | | | | | | | | | | | | | | | | | | |
| b2 | 2 BSS_100M | | | | | | | | | | | | | | | | | | |
| b3 | 3 GPADC_10M | | | | | | | | | | | | | | | | | | |
| b4 | 4 RCOSC_10M | | | | | | | | | | | | | | | | | | |
| b5 | 5 RAMPGEN_100M (Not supported) | | | | | | | | | | | | | | | | | | |
| b15:6 | RESERVED | | | | | | | | | | | | | | | | | | |
| RESERVED | 4 | 0x00000000 | | | | | | | | | | | | | | | | | |

7.15 RX Saturation Detection Monitor

This section contains API SBs that configure the monitoring of RX analog saturation detectors, and report the results from the monitor. The corresponding monitors are collectively named RX_SATURATION_DETECTOR_MONITOR and RX_SIG_IMG_BAND_MONITOR. The report is available in CQ RAM.

NOTE: The RX saturation detection monitor APIs are not supported in DFP 1.x (1st generation devices). Please refer latest DFP release note for more info.

7.15.1 Sub block 0x01DB – AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB

This API is a monitoring configuration API which the host sends to the mmWave device, containing information related to RX saturation detector monitoring. The report is available as CQ2 (part of CQ) in CQ RAM every chirp. The application should transfer the report from CQ RAM every chirp.

Table 7.29: AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB contents

| Field Name | Number of bytes | Description |
|-------------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01DB |
| SBLKLEN | 2 | Value = 24 |
| PROFILE.INDX | 1 | This field indicates the profile index for which this monitoring configuration applies. |
| SAT_MON_SELECT | 1 | 01 – Enable only the ADC saturation monitor 11 – Enable both the ADC and IFA1 saturation monitors |
| RESERVED | 1 | 0x00 |
| RESERVED | 1 | 0x00 |
| SAT_MON_PRIMARY_TIME_SLICE_DURATION | 2 | It specifies the duration of each (primary) time slice. 1 LSB = 0.16 μ s. Valid range: 4 to floor(ADC sampling time us/0.16 μ s) NOTES: The minimum allowed duration of each (primary) time slice is 4 LSBs = 0.64 μ s. Also, the maximum number of (primary) time slices that will be monitored in a chirp is 64 so the recommendation is to set this value to correspond to (ADC sampling time / 64). If the slice is smaller, such that the ADC sampling time is longer than 64 primary slices, some regions of the valid duration of a chirp may go un-monitored. |
| SAT_MON_NUM_SLICES (N) | 2 | Number of (primary + secondary) time slices to monitor. Valid range: 1 to 127 NOTE1: Together with SAT_MON_PRIMARY_TIME_SLICE_DURATION, this determines the full duration of the ADC valid time that gets covered by the monitor. Primary slices = (N+1) / 2 Secondary slices = Primary slices - 1 NOTE2: The total monitoring duration is recommended to program slightly smaller than ADC sampling time |

Continued on next page

Table 7.29 – continued from previous page

| | | |
|-------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SAT_MON_RX_CHANNEL_MASK | 1 | Masks RX channels used for monitoring. In every slice, saturation counts for all unmasked channels are added together, and the total is capped to 127. The 8 bits are mapped (MSB->LSB) to: [RX3Q, RX2Q, RX1Q, RX0Q, RX3I, RX2I, RX1I, RX0I] 00000000 – All channels unmasked 11111111 – All channels masked |
| RESERVED | 1 | 0 |
| RESERVED | 1 | 0 |
| RESERVED | 1 | 0 |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.15.2 Sub block 0x01DC – AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB

This API is a monitoring configuration API which the host sends to the mmWave device, containing information related to signal and image band energy. The report is available as CQ1 (part of CQ) in CQ RAM. The application should transfer the report every chirp.

Table 7.30: AWR_MONITOR_RX_SIG_IMG_MONITOR_CONF_SB contents

| Field Name | Number of bytes | Description |
|------------------------|-----------------|-----------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01DC |
| SBLKLEN | 2 | Value = 16 |
| PROFILE.INDX | 1 | This field indicates the profile index for which this monitoring configuration applies. |
| SIG_IMG_MON_NUM_SLICES | 1 | Number of (primary + secondary) slices to monitor Valid range: 1 to 127 |

Continued on next page

Table 7.30 – continued from previous page

| | | |
|------------------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NUM_SAMPLES_PER_PRIMARY_TIME_SLICE | 2 | <p>This field specifies the number of samples constituting each time slice. The minimum allowed value for this parameter is 4. Valid range: 4 to NUM_ADC_SAMPLES (see NOTE2 below)</p> <p>NOTE1: The maximum number of (primary) time slices that will be monitored in a chirp is 64, so our recommendation is that this value should at least equal (NUM_ADC_SAMPLES / 64). If the slice is smaller, such that the number of ADC samples per chirp is larger than 64 primary slices, some regions of the valid duration of a chirp may go un-monitored.</p> <p>NOTE2: In Complex1x mode, the minimum number of samples per slice is 4 and for other modes it is 8. Also note that number of samples should be an even number.</p> <p>NOTE3:The total monitoring duration is recommended to program slightly smaller than ADC sampling time</p> |
| RESERVED | 4 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

NOTE: It is recommended to re-issue these rx saturation and/or signal image monitor configuration APIs each time before enabling these monitor and frame trigger. The right sequence is as below:

1. Issue rx saturation and/or signal image monitor configuration API.
2. Enable rx saturation and/or signal image monitor.
3. Frame start.
4. Frame stop.
5. Frame start. (Optional in case of multiple frames)
6. Frame stop. (Optional in case of multiple frames)
7. Disable rx saturation and/or signal image monitor (in case disabled for some reason)
8. Issue rx saturation and/or signal image monitor configuration API.
9. Enable rx saturation and/or signal image monitor.
10. Frame start.

7.16 RX mixer input power monitor

7.16.1 Sub block 0x01DD – AWR_MONITOR_RX_MIXER_IN_POWER_CONF_SB

This API is a monitoring configuration API which the host sends to the AWR device, containing information related to RX mixer input power monitoring. The report is sent as an async event [AWR_MONITOR_RX_MIXER_IN_POWER_REPORT_AE_SB](#).

NOTE: The RX input power monitor API is debug only API. Please refer latest DFP release note for more info.

Table 7.31: AWR_MONITOR_MIXER_IN_POWER_CONF_SB contents

| Field Name | Number of bytes | Description | | | | | | | | |
|----------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------|----|----------------------------------------------------------------|----|--------------------------------------------------------------------|----|-------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01DD | | | | | | | | |
| SBLKLEN | 2 | Value = 16 | | | | | | | | |
| PROFILE_INDEX | 1 | The RF analog settings corresponding to this profile are used for monitoring RX mixer input power using test chirps (static frequency, at the center of the profile's RF frequency band). | | | | | | | | |
| REPORTING_MODE | 1 | <table border="0"> <tr> <td>Value</td> <td>Definition</td> </tr> <tr> <td>0</td> <td>Report is sent every monitoring period without threshold check</td> </tr> <tr> <td>1</td> <td>Report is send only upon a failure (after checking for thresholds)</td> </tr> <tr> <td>2</td> <td>Report is sent every monitoring period with threshold check</td> </tr> </table> | Value | Definition | 0 | Report is sent every monitoring period without threshold check | 1 | Report is send only upon a failure (after checking for thresholds) | 2 | Report is sent every monitoring period with threshold check |
| Value | Definition | | | | | | | | | |
| 0 | Report is sent every monitoring period without threshold check | | | | | | | | | |
| 1 | Report is send only upon a failure (after checking for thresholds) | | | | | | | | | |
| 2 | Report is sent every monitoring period with threshold check | | | | | | | | | |
| TX_EN | 1 | <p>This field indicates if and which TX channels should be enabled while measuring RX mixer input power. Setting a bit to 1 enables the corresponding TX channel. Enabling a TX channel may help find reflection power while disabling may help find interference power.</p> <table border="0"> <tr> <td>Bit number</td> <td>TX Channel</td> </tr> <tr> <td>b0</td> <td>TX0</td> </tr> <tr> <td>b1</td> <td>TX1</td> </tr> <tr> <td>b2</td> <td>TX2</td> </tr> </table> | Bit number | TX Channel | b0 | TX0 | b1 | TX1 | b2 | TX2 |
| Bit number | TX Channel | | | | | | | | | |
| b0 | TX0 | | | | | | | | | |
| b1 | TX1 | | | | | | | | | |
| b2 | TX2 | | | | | | | | | |
| RESERVED | 1 | 0x00 | | | | | | | | |

Continued on next page

Table 7.31 – continued from previous page

| | | |
|------------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| THRESHOLDS | 2 | <p>The measured RX mixer input voltage swings during this monitoring is compared against the minimum and maximum thresholds configured in this field. The comparison result is part of the monitoring report message (Status bit is cleared if any measurement is outside this (minimum, maximum) range).</p> <p>Byte number Threshold</p> <p>0 Minimum Threshold</p> <p>1 Maximum Threshold</p> <p>Only the RX channels enabled in the static configuration APIs are monitored.</p> <p>1 LSB = 1800 mV/256, unsigned number Valid range: 0 to 255, maximum threshold \geq minimum threshold</p> |
| RESERVED | 2 | 0x00000000 |
| RESERVED | 4 | 0x00000000 |

7.17 Sub block 0x01DE – RESERVED

7.18 Analog Fault injection

7.18.1 Sub block 0x01DF – AWR_ANALOG_FAULT_INJECTION_CONF_SB

This API is a fault injection API which the host sends to the AWR device. It can be used to inject faults in the analog circuits to test the corresponding monitors. After the faults are injected, the regular enabled monitors will indicate the faults in their associated reports.

| | |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NOTE1: | This API should be issued when no frames are on-going. |
| NOTE2: | The fault injection should be tested by injecting one fault at a time and corresponding analog monitor should be observed, other monitors might show failure depending on type of fault, it can be discarded. |
| NOTE3: | It is recommended to perform device reset after enabling fault injection before moving to functional mode. |

Table 7.32: AWR_ANALOG_FAULT_INJECTION_CONF_SB contents

| Field Name | Number of bytes | Description |
|--------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SBLKID | 2 | Value = 0x01DF |
| SBLKLEN | 2 | Value = 24 |
| RESERVED | 1 | 0x00 |
| RX_GAIN_DROP | 1 | <p>Primary Fault: RX Gain This field indicates which RX RF sections should have fault injected. If the fault is enabled, the RX RF gain drops significantly. The fault can be used to cause significant gain change, inter-RX gain imbalance and an uncontrolled amount of inter-RX phase imbalance.</p> <p>Bit number RX Channel b0 RX0 b1 RX1 b2 RX2 b3 RX3 Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| RX_PHASE_INV | 1 | <p>Primary Fault: RX Phase This field indicates which RX channels should have fault injected. If the fault is enabled, the RX phase gets inverted. The fault can be used to cause a controlled amount (180o) of inter-RX phase imbalance.</p> <p>Bit number RX Channel b0 RX0 b1 RX1 b2 RX2 b3 RX3 Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |

Continued on next page

Table 7.32 – continued from previous page

| | | |
|--------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RX_HIGH_NOISE | 1 | <p>Primary Fault: RX Noise</p> <p>This field indicates which RX channels should have fault injected. If the fault is enabled, the RX IFA square wave loopback paths are engaged to inject high noise at RX IFA input. The fault can be used to cause significant RX noise floor elevation.</p> <p>Bit number RX Channel</p> <p>b0 RX0</p> <p>b1 RX1</p> <p>b2 RX2</p> <p>b3 RX3</p> <p>Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| RX_IF_STAGES_FAULT | 1 | <p>Primary Fault: Cutoff frequencies of RX IFA HPF & LPF, IFA Gain.</p> <p>This field indicates which RX channels should have fault injected. If the fault is enabled, the RX IFA HPF cutoff frequency becomes very high (about 15MHz). The fault can be used to cause the measured inband IFA gain, HPF and LPF attenuations to vary from ideal expectations.</p> <p>Bit number RX Channel</p> <p>b0 RX0</p> <p>b1 RX1</p> <p>b2 RX2</p> <p>b3 RX3</p> <p>Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> <p>NOTE: During the execution of RX_IFSTAGE_MONITOR, the RX_HIGH_NOISE faults are temporarily removed.</p> |

Continued on next page

Table 7.32 – continued from previous page

| | | |
|-----------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RX_LO_AMP_FAULT | 1 | <p>Primary Fault: RX Mixer LO input swing reduction</p> <p>This field indicates which RX channels should have fault injected. If the fault is enabled, the RX mixer LO input swing is significantly reduced. The fault is primarily expected to be detected by RX.INTERNAL_ANALOG_SIGNALS_MONITOR (under PWRDET_RX category).</p> <p>Bit number RX Channel</p> <p>b0 RX0</p> <p>b1 RX1</p> <p>b2 RX2</p> <p>b3 RX3</p> <p>Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| TX_LO_AMP_FAULT | 1 | <p>Primary Fault: TX PA input signal generator turning off.</p> <p>This field indicates which TX channels should have fault injected. If the fault is enabled, the amplifier generating TX power amplifier's LO input signal is turned off. The fault is primarily expected to be detected by TX<n>.INTERNAL_ANALOG_SIGNALS_MONITOR (under DCBIAS category).</p> <p>Bit number Channel</p> <p>b0 TX0 and TX1</p> <p>b1 TX2 (applicable only if available in the device)</p> <p>Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |

Continued on next page

Table 7.32 – continued from previous page

| | | | | | | | | | | | | | | | | | | |
|--------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------|----|---------------------------------------|----|----------|----|----------|--------|---------------|----|---------------|----|---------------|--------|----------|
| TX_GAIN_DROP | 1 | <p>Primary Fault: TX Gain (power)</p> <p>This field indicates which TX RF sections should have fault injected. If the fault is enabled, the TX RF gain drops significantly. The fault can be used to cause significant TX output power change, inter-TX gain imbalance and an uncontrolled amount of inter-TX phase imbalance.</p> <table border="0"> <tr> <td>Bit number</td> <td>Channel</td> </tr> <tr> <td>b0</td> <td>TX0</td> </tr> <tr> <td>b1</td> <td>TX1</td> </tr> <tr> <td>b2</td> <td>TX2</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> | Bit number | Channel | b0 | TX0 | b1 | TX1 | b2 | TX2 | Others | RESERVED | | | | | | |
| Bit number | Channel | | | | | | | | | | | | | | | | | |
| b0 | TX0 | | | | | | | | | | | | | | | | | |
| b1 | TX1 | | | | | | | | | | | | | | | | | |
| b2 | TX2 | | | | | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | | | | | |
| TX_PHASE_INV | 1 | <p>Primary Fault: TX Phase</p> <p>This field indicates if TX channels should have fault injected, along with some further programmability. If the fault is enabled, the TX BPM polarity (phase) is forced to a constant value as programmed. The fault can be used to cause a controlled amount (180 degree) of inter-TX phase imbalance as well as BPM functionality failure.</p> <table border="0"> <tr> <td>Bit number</td> <td>TX Channel</td> </tr> <tr> <td>b0</td> <td>TX_FAULT (Common for all TX channels)</td> </tr> <tr> <td>b1</td> <td>RESERVED</td> </tr> <tr> <td>b2</td> <td>RESERVED</td> </tr> <tr> <td>b3</td> <td>TX0_BPM_VALUE</td> </tr> <tr> <td>b4</td> <td>TX1_BPM_VALUE</td> </tr> <tr> <td>b5</td> <td>TX2_BPM_VALUE</td> </tr> <tr> <td>Others</td> <td>RESERVED</td> </tr> </table> <p>For each TXn_BPM_VALUE: Applicable only if TX_FAULT is enabled. Value = 0: force TX<n> BPM polarity to 0 Value = 1: force TX<n> BPM polarity to 1.</p> <p>NOTE: The TXn_BPM_VALUE takes effect only when TX_FAULT value is changed</p> | Bit number | TX Channel | b0 | TX_FAULT (Common for all TX channels) | b1 | RESERVED | b2 | RESERVED | b3 | TX0_BPM_VALUE | b4 | TX1_BPM_VALUE | b5 | TX2_BPM_VALUE | Others | RESERVED |
| Bit number | TX Channel | | | | | | | | | | | | | | | | | |
| b0 | TX_FAULT (Common for all TX channels) | | | | | | | | | | | | | | | | | |
| b1 | RESERVED | | | | | | | | | | | | | | | | | |
| b2 | RESERVED | | | | | | | | | | | | | | | | | |
| b3 | TX0_BPM_VALUE | | | | | | | | | | | | | | | | | |
| b4 | TX1_BPM_VALUE | | | | | | | | | | | | | | | | | |
| b5 | TX2_BPM_VALUE | | | | | | | | | | | | | | | | | |
| Others | RESERVED | | | | | | | | | | | | | | | | | |

Continued on next page

Table 7.32 – continued from previous page

| | | |
|------------------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYNTH_FAULT | 1 | <p>Primary Fault: Synthesizer Frequency This field indicates which Synthesizer faults should be injected.</p> <p>SYNTH.VCO_OPENLOOP: If the fault is enabled, the synthesizer is forced in open loop mode with the VCO control voltage forced to a constant. In order to avoid out of band emissions in this faulty state, this fault is injected just before the PLL.CONTROL.VOLTAGE.MONITOR is executed and released just after its completion.</p> <p>Note: In 1st generation devices (xWR1xxx devices) this feature is not supported.</p> <p>SYNTH.FREQ_MON_OFFSET: If the fault is enabled, the synthesizer frequency monitor's ideal frequency ramp waveform is forced to be offset from the actual ramp waveform by a constant, causing monitoring to detect failures.</p> <p>Bit number Enable Fault b0 SYNTH.VCO_OPENLOOP (Not supported in 1st gen devices) b1 SYNTH.FREQ_MON_OFFSET Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| SUPPLY_LDO_FAULT | 1 | <p>This field indicates whether some LDO output voltage faults should be injected or not.</p> <p>Bit number Enable Fault b0 SUPPLY_LDO_RX_LODIST_FAULT Others RESERVED</p> <p>SUPPLY_LDO_RX_LODIST_FAULT: if enabled, the RX LO distribution sub system's LDO output voltage is slightly changed compared to normal levels to cause INTERNAL_PMCLKLO_SIGNALS_MONITOR to detect failure (under SUPPLY category).</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> <p>NOTE: This fault injection is ineffective under LDO bypass condition.</p> |

Continued on next page

Table 7.32 – continued from previous page

| | | |
|-------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MISC_FAULT | 1 | <p>This field indicates whether a few miscellaneous faults should be injected or not.</p> <p>Bit number Enable Fault b0 GPADC_CLK_FREQ_FAULT Others RESERVED</p> <p>GPADC_CLK_FREQ_FAULT: if enabled, the GPADC clock frequency is slightly increased compared to normal usage to cause BSS DCC_CLOCK_FREQ_MONITOR to detect failure.</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| MISC_THRESH_FAULT | 1 | <p>This field indicates whether faults should be forced in the threshold comparisons in the software layer of some monitors. If a fault is enabled, the logic in the min-max threshold comparisons used for failure detection is inverted, causing a fault to be reported. During these faults, no hardware fault condition is injected in the device.</p> <p>Bit number Enable Fault b0 GPADC_INTERNAL_SIGNALS_MONITOR Others RESERVED</p> <p>For each bit, 1 = inject fault, 0 = remove injected fault</p> |
| RESERVED | 3 | 0x000000 |
| RESERVED | 4 | 0x00000000 |

8 Unsupported Features/APIs and Debug APIs

8.1 Unsupported Features/APIs and Debug APIs

The list of unsupported features, APIs and debug APIs are highlighted in latest DFP release note. Please refer latest DFP release note for each specific devices.

9 Chirp Parameters (CP) and Chirp Quality (CQ) data

9.1 Chirp Parameters data

Chirp parameter information is always updated in the CP registers DSS.REG_VBUSM_CPREG[0-3] for single chirp use case.

NOTE: Chirp Number is always reset every burst by the hardware.

| | | Channel 0 | | | | Channel 1 | | | | Channel 2 | | | | Channel 3 | | | | | | | | | | | |
|------|---|----------------|----------|-------------------|--------------------|--------------------|----------|-------------------|--------------------|----------------|----------|--------------------|--------------------|-------------------|----------|-------------------|--------------------|----------------|-------------------|----------------|--------------------|----------------|-------------------|----------------|--------------------|
| | | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 | Byte 10 | Byte 11 | Byte 12 | Byte 13 | Byte 14 | Byte 15 | | | | | | | | |
| Bits | 0 | Channel Number | Reserved | Chirp Number[7:0] | Chirp Number[11:8] | Channel Number | Reserved | Chirp Number[7:0] | Chirp Number[11:8] | Channel Number | Reserved | Chirp Number[7:0] | Chirp Number[11:8] | Channel Number | Reserved | Chirp Number[7:0] | Chirp Number[11:8] | | | | | | | | |
| | 1 | | | | | | | | | | | | | | | | | Profile Number | Reserved | Profile Number | Reserved | Profile Number | Reserved | Profile Number | Reserved |
| | 2 | Reserved | | | Chirp Number[7:0] | Reserved | | | Chirp Number[11:8] | Reserved | | | Chirp Number[7:0] | Reserved | | | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | Reserved | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | Reserved | Chirp Number[11:8] |
| | 3 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 4 | Reserved | | Chirp Number[7:0] | Reserved | Chirp Number[11:8] | | Reserved | Chirp Number[7:0] | Reserved | | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | | Reserved | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | Reserved | Chirp Number[11:8] | | | | |
| | 5 | | | | | | | | | | | | | | | | | | | | | Reserved | Reserved | Reserved | Reserved |
| | 6 | Reserved | | Chirp Number[7:0] | Reserved | Chirp Number[11:8] | | Reserved | Chirp Number[7:0] | Reserved | | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | | Reserved | Chirp Number[11:8] | Reserved | Chirp Number[7:0] | Reserved | Chirp Number[11:8] | | | | |
| | 7 | | | | | | | | | | | | | | | | | | | | | Reserved | Reserved | Reserved | Reserved |

Figure 9.1: Chirp parameter information fields

| | 31 | 23 | 16 | 15 | 8 | 7 | 0 |
|-------------------------|---------|---------|---------|---------|---------|---------|---------|
| DSS_REG_VBUSM.CH0CPREG0 | Byte 3 | Byte 2 | Byte 1 | Byte 1 | Byte 0 | Byte 0 | Byte 0 |
| DSS_REG_VBUSM.CH0CPREG1 | Byte 7 | Byte 6 | Byte 5 | Byte 5 | Byte 4 | Byte 4 | Byte 4 |
| DSS_REG_VBUSM.CH0CPREG2 | Byte 11 | Byte 10 | Byte 9 | Byte 9 | Byte 8 | Byte 8 | Byte 8 |
| DSS_REG_VBUSM.CH0CPREG3 | Byte 15 | Byte 14 | Byte 13 | Byte 13 | Byte 12 | Byte 12 | Byte 12 |

Figure 9.2: Chirp parameter information from DSS registers

For multichip use case, the CP data is available for up to 8 chirps in DSS_REG.VBUSM.CH[0-7]CPREG[0-3].

9.2 Chirp Quality data

Chirp quality information is divided into 3 parts

1. CQ0 – Wideband signal and image energy information (Reserved for future use)
2. CQ1 – RX signal and image band energy statistics
3. CQ2 – RX ADC and IF saturation information

CQ data will be available in CQ RAM which is a ping-pong memory when the CQ monitors are enabled. Currently supported CQ monitors are AWR_MONITOR_RX_SATURATION_DETECTOR_CONF_SB for CQ2 and AWR_MONITOR_SIG_IMG_MONITOR_CONF_SB for CQ1. CQ data will be refreshed every chirp by the hardware. User has to ensure that before the next chirp finishes, the current chirps' CQ data is either processed or transferred to a local memory for further processing.

NOTE: CQ0 is not supported by firmware currently, but the CQ RAM will be updated for CQ0 data. Maximum size of CQ0 data is 256 bytes. Users should ignore the CQ RAM for CQ0.

The starting location (on 128 bit boundary) of each CQ data within the CQ memory can be configured by programming DSS_REG.CQCFG1[12:4] for CQ0, DSS_REG.CQCFG1[21:13] for CQ1 and DSS_REG.CQCFG1[30:22] for CQ2.

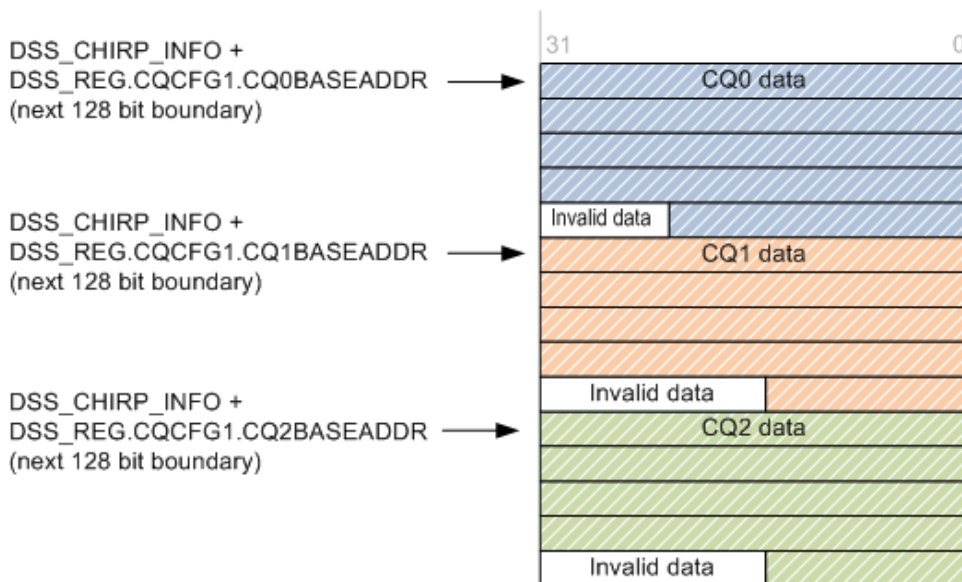


Figure 9.3: CQ data start address configuration in single chirp use case

For N-chirp use case, when user wishes to process N chirps simultaneously, then CQ0 for all N chirps will be concatenated together in memory. Similarly CQ1 and CQ2 for all N chirps will also be concatenated together.

NOTE: When CQ data is concatenated in N-chirp use case, the CQ data for new chirp starts on the next 128 bit boundary.

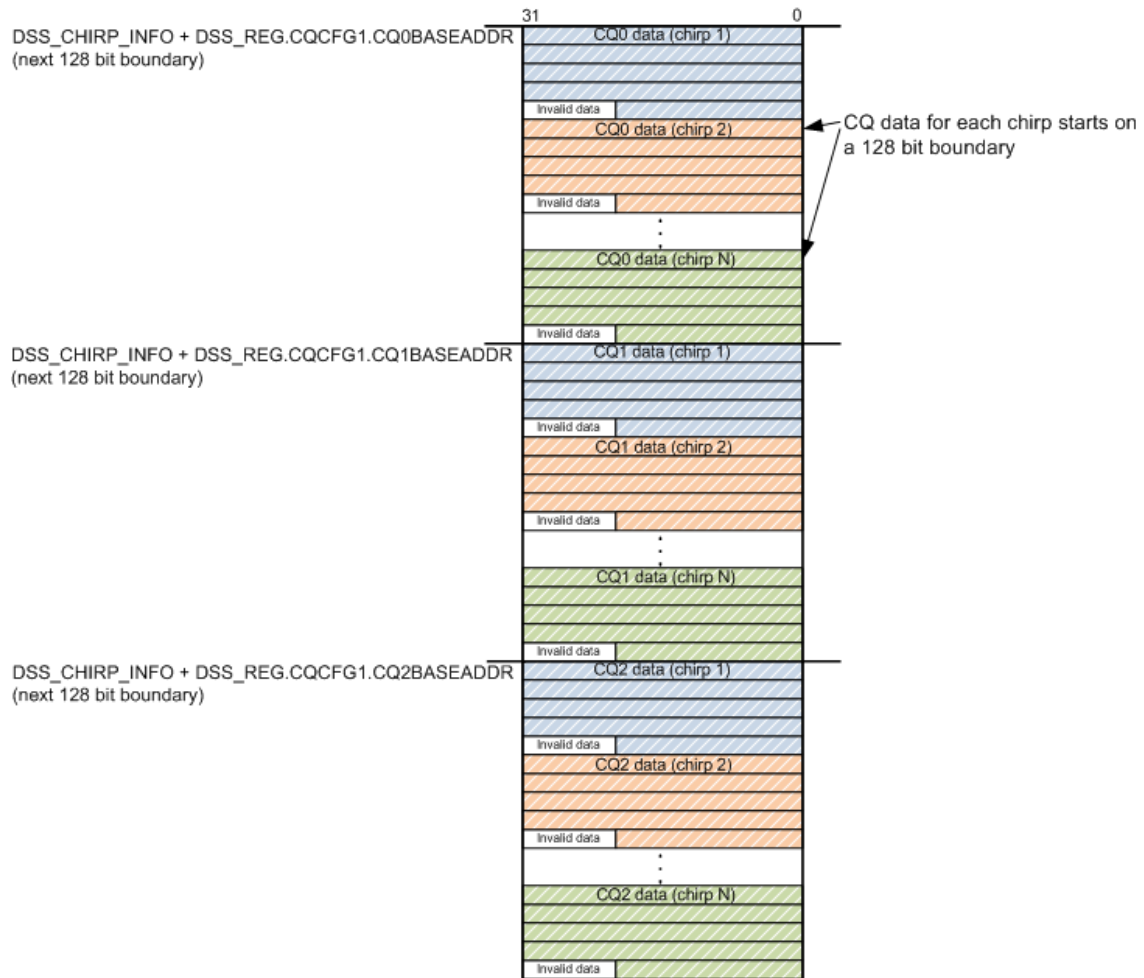


Figure 9.4: CQ data start address configuration in multi chirp use case

The `CQDATAWIDTH` parameter in `DSS_REG.CQCFG1` defines the packing of the CQ data in the CQ memory in either 16-bit mode, 12-bit mode or in 14-bit mode.

9.2.1 CQ1

The signal band and image band are separated using a two-channel filter bank and the ADC sampling time duration is monitored in terms of primary and secondary time slices, as shown below.

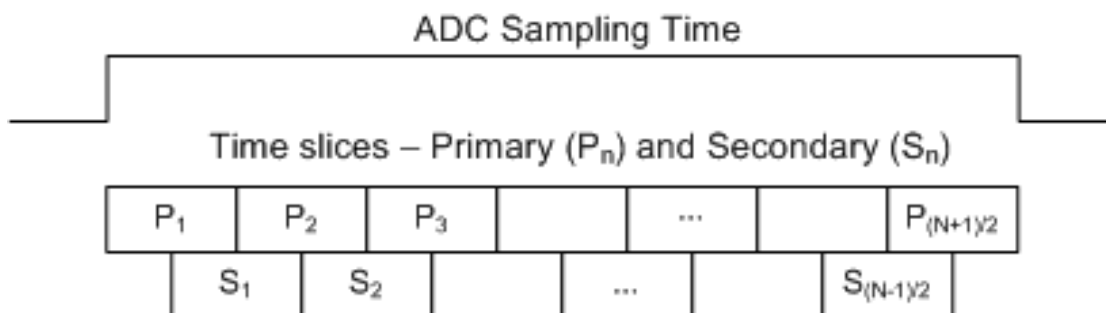


Figure 9.5: Time slices during RX signal and image band monitor and saturation monitor

For each of the two bands (signal and image), for each time slice, the input-referred average power in the slice in negative dBm is recorded as an 8-bit unsigned number, with 1 LSB = -0.5 dBm

CQ1 data is stored in memory as shown below (in 16-bit mode)

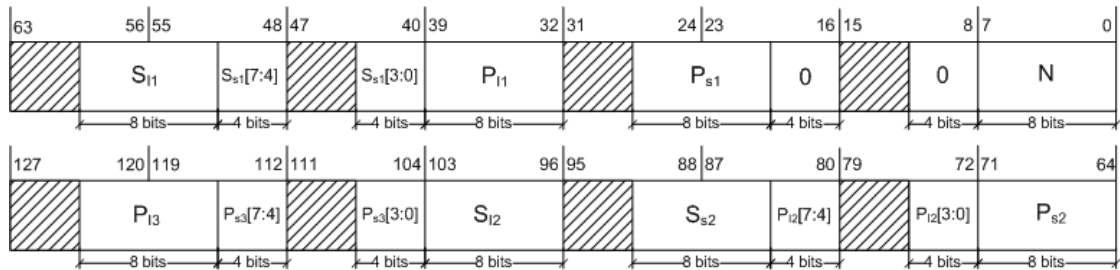


Figure 9.7: CQ1 data format in memory in 12-bit mode

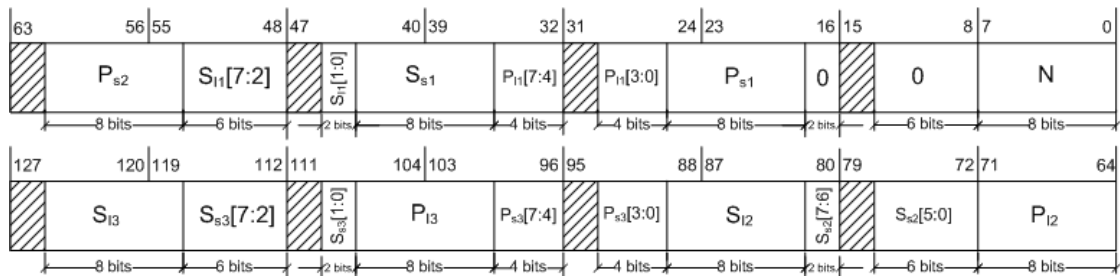


Figure 9.8: CQ1 data format in memory in 14-bit mode

9.2.2 CQ2

The analog to digital interface includes a 100 MHz bit stream indicating saturation events in the ADC/IF sections, for each channel. This one-bit indicator for each channel is monitored during the ADC sampling time duration in a time-sliced manner, as shown in Figure 9.5.

For each time slice, a saturation event count is recorded. This count is the sum of saturation event counts across all RX channels selected for monitoring, capped to a maximum count of 255 (8 bits). The saturation counts are stored in memory as shown below

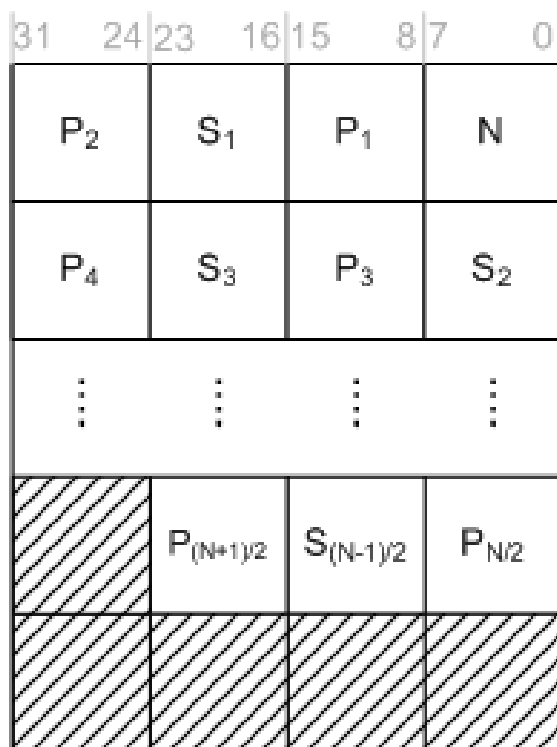


Figure 9.9: CQ2 data format in memory in 16-bit mode

N indicates the total number of primary and secondary slices which are monitored (maximum value of N is 127). P_n indicates the accumulated saturation count for all enabled RX channels in primary slice n, S_n indicates the accumulated saturation count for all enabled RX channels in secondary slice n.

Since maximum value of N is 127, the maximum size of CQ2 data in 16-bit mode is 128 bytes. Similarly, in 12-bit and 14-bit modes, the CQ2 data in CQ memory will be packed as shown below. Only the relevant bits in each 16 bits of memory (either 12 bits or 14 bits) are useful and other bits are not written by hardware.

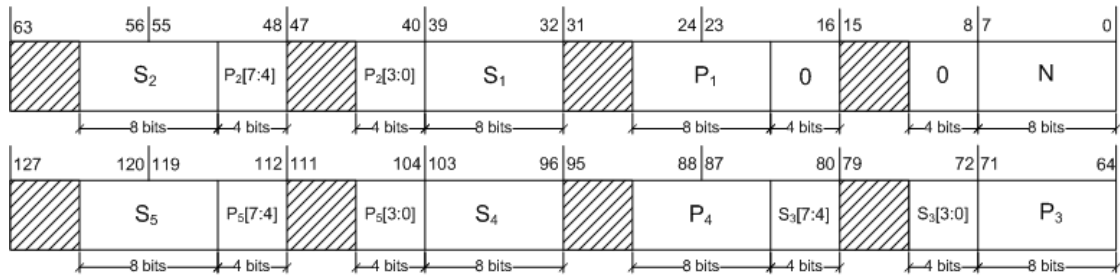


Figure 9.10: CQ2 data format in memory in 12-bit mode

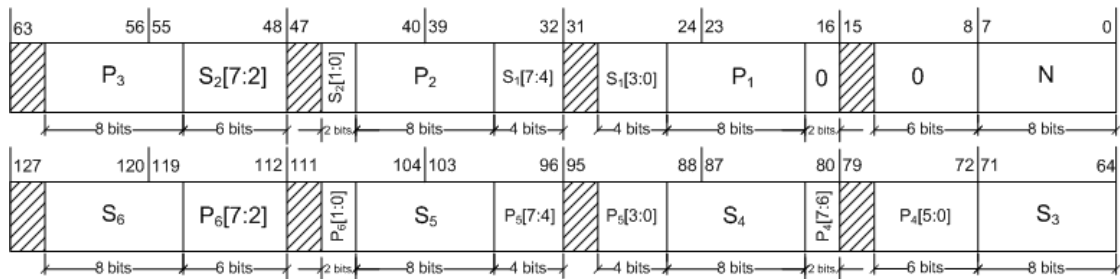


Figure 9.11: CQ2 data format in memory in 14-bit mode

10 Calibration and monitoring durations

10.1 Boot time calibration durations

Table 10.1: Duration of boot time calibrations

| Sl. No. | Calibration | Duration (μ s) |
|---------|----------------------------------|---------------------|
| 1 | APLL | 330 |
| 2 | Synth VCO | 1300 |
| 3 | LO DIST | 12 |
| 4 | ADC DC | 600 |
| 5 | HPF cutoff | 3500 |
| 6 | LPF cut off | 3200 |
| 7 | Peak detector | 4200 |
| 8 | TX power (assumes 2 TX use-case) | 6000 |
| 9 | RX gain | 2300 |
| 10 | TX phase | 40 000 |
| 11 | RX IQMM | 32 000 |

10.2 Run time calibration durations

Table 10.2 lists the duration of all run time calibrations. Note that the firmware performs calibrations in small chunks of 250 μ s. User has to ensure that the total idle time in one CAL_MON_TIME_UNIT is sufficient to fit the enabled calibrations.

To configure CALIB.MON.TIME.UNIT, user has to calculate the total available IDLE time in the frame and subtract 100 μ s for every frame to allow for preparation of frame. The duration for all the enabled calibrations should be included and following software overheads should be added to that number

Table 10.2: Duration of run time calibrations

| Sl. No. | Calibration | Duration (μ s) |
|---------|---------------------------------------------------------------------------|---------------------|
| 1 | APLL | 150 |
| 2 | Synth VCO | 350 |
| 3 | LO DIST | 30 |
| 4 | Peak detector | 500 |
| 5 | TX power (assumes 1 TX, 1 profile) | 800 |
| 6 | RX gain | 30 |
| 7 | Application of calibration to hardware (This needs to be included always) | 150 |

10.3 Monitoring duration

Table 10.3 lists the duration of all analog monitors and Table 10.4 lists the duration of all digital monitors

Table 10.3: Duration of analog monitors

| Sl. No. | Monitors | Duration (μ s) |
|---------|--------------------------------------------------------|---------------------|
| 1 | RX gain phase (assumes 1 RF frequency) | 1250 |
| 2 | RX noise figure (assumes 1 RF frequency) | 250 |
| 3 | RX IF stage (assumes 1 RF frequency) | 1000 |
| 4 | TX power (assumes 1 TX, 1 RF frequency) | 200 |
| 5 | TX ballbreak (assumes 1 TX) | 250 |
| 6 | TX gain phase mismatch (assumes 1 TX, 1 RF frequency) | 400 |
| 7 | TX BPM (assumes 1 TX) | 575 |
| | - TX phase shifter (assumes 1 TX) | 525 |
| 8 | Synthesizer frequency | 0 |
| 9 | External analog signals (all 6 GPADC channels enabled) | 150 |
| 10 | TX Internal analog signals (assumes 1 TX) | 200 |
| 11 | RX internal analog signals | 1700 |
| 12 | PMCLKLO internal analog signals | 400 |
| 13 | GPADC internal signals | 50 |
| 14 | PLL control voltage | 210 |
| 15 | Dual clock comparator (assumes 6 clock comparators) | 110 |
| 16 | RX saturation detector | 0 |
| 17 | RX signal and image band monitor | 0 |
| 18 | RX mixer input power | 350 |
| 19 | Temperature monitor | 200 |

Table 10.4: Duration of digital monitors

| Sl. No. | Monitors | Duration (μ s) |
|---------|------------------------------------------|---------------------|
| 1 | Periodic configuration register readback | 100 |
| 2 | DFE LBIST monitoring | 1000 |
| 3 | Frame timing monitoring | 10 |

10.4 Software overheads

When the calibrations or monitorings are enabled, the software needs certain time for reading the temperature sensors, reading the DFE statistics, preparing the calibration or monitoring reports and to clear the watchdog (WDT). All these time durations should also be accounted when computing the CALIB_MON_TIME_UNIT. The details of the software overheads are given in the Table 10.5

Table 10.5: Software overheads every FTTI that should be accounted to program CALIB_MON_TIME_UNIT and CALIBRATION_PERIODICITY

| Sl. No. | Software overhead | Duration (μ s) |
|---------|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Periodic monitoring of stack usage | 20 |
| 2 | Minimum monitoring duration (report formation, digital energy monitor at the end of FTTI, temperature read every FTTI) | 1000 |
| 3 | Minimum calibration duration (report formation, temperature read every CAL_MON_TIME_UNIT)+ | 500 |
| 4 | Idle time needed per FTTI for windowed watchdog (WDT) | $\text{Frame period} \times \text{CALIB_MON_TIME_UNIT}/8$ i.e. $\sim 12.5\%$ of Frame period \times CALIB_MON_TIME_UNIT is reserved for watchdog clearing time |

10.4.1 Note on idle time for clearing the watchdog (WDT)

The clearing window of the watchdog is 12.5% of total FTTI as shown in the figure below. One FTTI can have multiple frames in legacy frame configuration or in advanced frame configuration - each frame can have multiple sub-frames and each sub-frame can have multiple bursts.

The required idle time for clearing watchdog is absolute 12.5% of the overall FTTI interval, this 12.5% clearing window can have multiple frames or subframes or bursts. The granularity of the required watchdog idle time calculation is limited to sub-frame period.

Example

A user has enabled advanced frame configuration where each frame consists of 3 sub-frames and each sub-frame is of 5 ms duration. FTTI is configured as 25 frames. Each sub-frame contains 100 chirps, each chirp consisting of 4 μ s idle time and 21 μ s ramp time. i.e. duty cycle is 50%. The watchdog clearing window and time for calibration and monitoring is calculated as follows

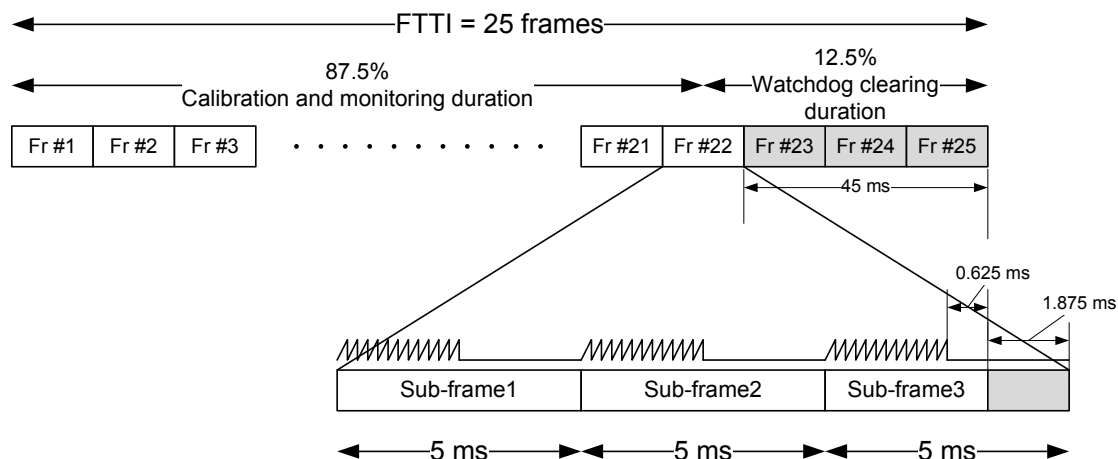


Figure 10.1: Watchdog idle time calculation

| | | | |
|-----------------------------------------------------------------------------------------------------------------------|---|---------|----|
| Frame duty cycle | = | 50 | % |
| Idle time per frame (50% of 15 ms) | = | 7.5 | ms |
| FTTI (15 ms × 25 frames) | = | 375 | ms |
| Available idle time per FTTI (50% of 375 ms) | = | 187.5 | ms |
| Ideal watchdog clearing window (12.5% of 375 ms) | = | 46.875 | ms |
| The calculated watchdog clearing window in firmware is as follows | | | |
| Duration of complete frames which can be fit in watchdog clearing window ($\lceil 46.875/15 \rceil \times 15$) | = | 45 | ms |
| Fractional watchdog clearing time (which will be fit in the sub-frame idle time) ($46.875 - (15 \times 3)$) | = | 1.875 | ms |
| Time available for calibration/monitoring per FTTI (21 frames × 7.5 ms) + (2 sub-frames × 2.5 ms) + 0.625 ms | = | 163.125 | ms |

The following examples show how the user can budget for calibration and monitoring time and configure the FTTI correctly.

Example 1

A user has enabled 2 TX, uses only 1 profile, frame configuration consists of 64 chirps, each chirp is of duration is 66 μs (56 μs ramp time and 10 μs chirp idle time) and frame periodicity is 10 ms. User has enabled all run time calibrations. None of the analog monitoring is enabled.

| | |
|----------------------------------------------------------------------------------------------------------|------------------------|
| Frame duty cycle | = 42.24% |
| Idle time per frame (57.76% of 10 ms) | = 5.776 ms |
| Idle time available for calibration/monitoring per frame (100 μs is for frame preparation) | } = 5.676 ms |
| Time needed for all run time calibrations..... | } = 2760 μs |
| 150 + 300 + 30 + 500 + (800 \times 2) + 30 + 150 | } |
| Minimum time for software overheads..... | } = 2770 μs |
| 20 + 1000 + 500 + (10000 \times 1/8) | } |
| Total time needed per frame for calibration 2760 μs + 2770 μs | } = 5530 μs |

Total time needed per frame for calibration is 5.530 μs which is less than the frame idle time (5.676 ms) and hence this configuration will be honored by the MMIC device.

User can set CALIB.MON.TIME.UNIT to 1 and CALIBRATION.PERIODICITY as 100. With this setting calibrations are triggered once every 100 frames (i.e. once every 1 s)

Example 2

Consider another example where the frame configuration remains the same as in example 1, but frame periodicity is reduced to 8 ms.

| | |
|----------------------------------------------------------------------------------------------------|------------------|
| Frame duty cycle | = 52.80% |
| Idle time per frame (47.20% of 8 ms) | = 3.776 ms |
| Idle time available for calibration/monitoring per frame (100 μ s is for frame preparation) | } = 3.676 ms |
| Time needed for all run time calibrations..... | } = 2760 μ s |
| 150 + 300 + 30 + 500 + (800 \times 2) + 30 + 150 | } |
| Minimum time for software overheads..... | } = 2520 μ s |
| 20 + 1000 + 500 + (8000 \times 1/8) | } |
| Total time needed per frame for calibration 2760 μ s + 2520 μ s | } = 5280 μ s |

Total time needed per frame for calibration is 5.280 μ s which is more than the frame idle time (3.676 ms) and hence this configuration will **not** be honored by the MMIC device.

User can set CALIB_MON_TIME_UNIT to 2 and CALIBRATION_PERIODICITY as 63. With this setting calibrations are triggered once every 126 frames (i.e. once every 1.008 s)

Example 3

A user has enabled 2 TX, uses 2 profiles, frame configuration consists of 32 chirps, each chirp is of duration is 90 μs (80 μs ramp time and 10 μs chirp idle time) and frame periodicity is 6 ms. User has enabled all run time calibrations. None of the analog monitoring is enabled.

| | |
|----------------------------------------------------------------------------------------------------------|------------------------|
| Frame duty cycle | = 48.00% |
| Idle time per frame (52.00% of 6 ms) | = 3.120 ms |
| Idle time available for calibration/monitoring per frame (100 μs is for frame preparation) | = 3.020 ms |
| Time needed for all run time calibrations..... | } = 4360 μs |
| 150 + 300 + 30 + 500 + (800 \times 2 \times 2) + 30 + 150 | |
| Minimum time for software overheads..... | } = 2270 μs |
| 20 + 1000 + 500 + (6000 \times 1/8) | |
| Total time needed per frame for calibration 4360 μs + 2270 μs | = 6630 μs |

Total time needed per frame for calibration is 6.630 μs which is more than the frame idle time (3.020 ms) and hence this configuration will **not** be honored by the MMIC device.

User can set CALIB_MON_TIME_UNIT to 3 and CALIBRATION_PERIODICITY as 56. With this setting, minimum required time is 8.13 ms and available idle time for calibration/monitoring is 9.06 ms and calibrations are triggered once every 168 frames (i.e. once every 1.008 s)

Example 4

A user has enabled 2 TX, uses 2 profiles, frame configuration consists of 32 chirps, each chirp is of duration is 90 μ s (80 μ s ramp time and 10 μ s chirp idle time) and frame periodicity is 6 ms. User has enabled all run time calibrations. Analog monitorings which are enabled are (a) TX output power monitor for TX0 and TX1 (b) TX BPM monitor for TX0 and TX1 (c) RX gain phase monitor and (d) RX noise figure monitor. Each of the monitors are configured to be run for 1 profile and 3 RF frequencies (low, mid and high) as defined by the profile.

| | |
|----------------------------------------------------------------------------------------------------|-------------------|
| Frame duty cycle | = 48.00% |
| Idle time per frame (52.00% of 6 ms) | = 3.120 ms |
| Idle time available for calibration/monitoring per frame (100 μ s is for frame preparation) | = 3.020 ms |
| Time needed for all run time calibrations..... | } = 4360 μ s |
| 150 + 300 + 30 + 500 + (800 \times 2 \times 2) + 30 + 150 | |
| Time needed for all monitoring..... | } = 6850 μ s |
| (1250 \times 3) + (250 \times 3) + (200 \times 3 \times 2) + (575 \times 2) | |
| Minimum time for software overheads..... | } = 2270 μ s |
| 20 + 1000 + 500 + (6000 \times 1/8) | |
| Total time needed per frame for calibration and monitoring | } = 13480 μ s |
| 4360 μ s + 6850 μ s + 2270 μ s | |

Total time needed per frame for calibration is 13.480 μ s which is more than the frame idle time (3.020 ms) and hence this configuration will **not** be honored by the MMIC device.

User can set CALIB_MON_TIME_UNIT to 6 and CALIBRATION_PERIODICITY as 28. With this setting, minimum required time for calibration and monitoring is 16.48 ms and available idle time for calibration/monitoring is 18.72 ms. Monitoring is triggered once in 6 frames and calibration is triggered once in 168 frames (i.e. once every 1.008 s)

10.5 Sample Application

For sample application please refer DFP (device firmware package) user guide document.

IMPORTANT NOTICE

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your noncompliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), evaluation modules, and samples (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated