IPCLib with SYS/BIOS and Vision SDK

Processor BU

ABSTRACT

This application note provides details of IPCLib integration with Vision SDK. It showcases which layers of the framework stack were modified to use IPCLib as inter processor communication layer. It also focuses on how IPCLib can be used without Vision SDK from an application running on SYS/BIOS.

Contents

Figures

Tables

1 Terms and Abbreviations

IPCLib – Bare metal IPCLib supported by PDK which can also be used with SYS/BIOS.

Bios IPC – Traditional IPC 3.x package (supports Notify, MessageQ and many other IPC modules) Vision SDK – ADAS video frame work for TDAxx devices

SYS/BIOS or sysbios – TI's operating system.

2 IPCLib integration with Vision SDK

2.1 Build

Vision SDK provides a configuration file build time argument IPC_LIB_INCLUDE Assuming the MAKECONFIG has been selected correctly in Rules.make

1. To build VisionSDK with IPCLib, update the configuration file with IPC_LIB_INCLUDE=yes

```
make –s –j  depend
make –s –j
```

2. To build VisionSDK with bios IPC, update the configuration file with IPC_LIB_INCLUDE=no

```
make –s –j  depend
make –s –j
```


2.2 Directory structure

System API IPCLib implementation
vision_sdk\links_fw\src\rtos\links_common\system\system_ipc_ipc_lib.c

System API Bios IPC implementation
vision_sdk\links_fw\src\rtos\links_common\system\system_ipc_bios_ipc.c

2.3 Integration

Vision SDK has different APIs (Link API, System API, Utils etc), these are used to put the framework together. IPCLib is integrated under System API. By not changing interface at System API all the links and applications remain unchanged and user can still switch between IPCLib and bios IPC at build time.

Figure 1 demonstrates different layers in Vision SDK and how IPCLib layer is integrated.

IPCLib is uses mailbox driver from PDK APIs to avail OS services like interrupt handler registration. The Link layer is agnostic of these details underneath and users IPCLib interface only.

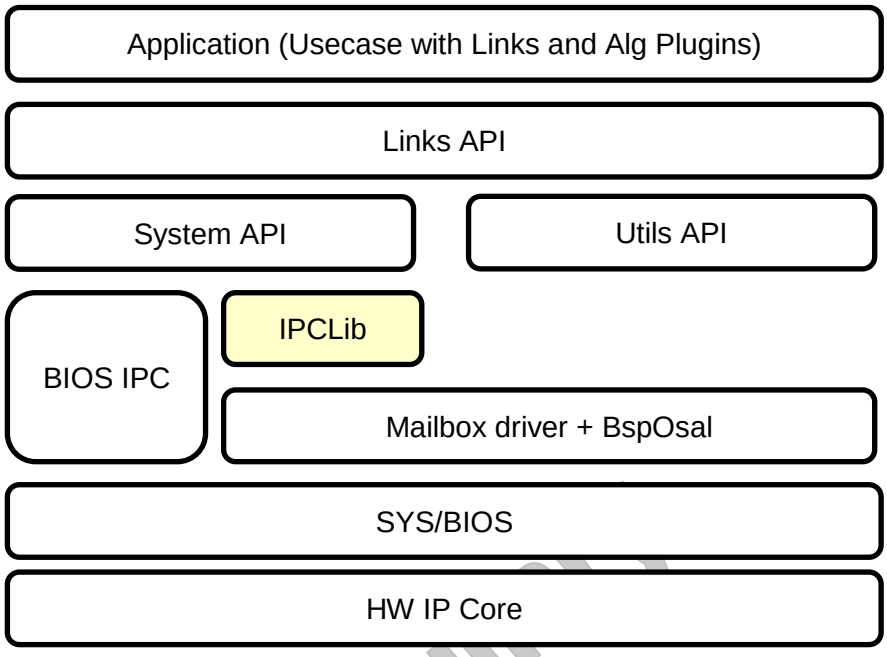

Figure 1 IPC integration with vision SDK

2.4 Synchronization logic

IPCLib gets used in multiprocessor applications. The synchronization after start up is a crucial thing and determines when it is safe to send / receive messages from remote core. Vision SDK implements synchronization logic using shared memory and following APIs.

```
void System_initDone(void)
void System_testInitDone(void)
void System_confirmInitDone(void)
```

These are implemented in vision_sdk\links_fw\src\rtos\links_common\system\system_common.c

These APIs use REMOTE_LOG_MEM shared memory section to ensure all cores sync to the master core where application is running and vice versa. The logic works with every core participating in application and ensures interrupts are used only after state transition from CORE_APP_INITSTATUS_INIT_DONE -> CORE_APP_INITSTATUS_TEST_INIT_DONE -> CORE_APP_INITSTATUS_CONFIRM_INIT_DONE happens for every core. There are core specific versions of same above APIs which can be used in usecases like fast boot where boot sequence is split between early and late.

3 IPCLib with SYS/BIOS

3.1 Interface

Interface remains same as that of bare metal IPCLib API.

ti_components\drivers\pdk_XX_XX_XX_XX\packages\ti\drv\ipc_lite\ipclib_interrupt.h

3.2 Sysbios based application with IPCLib

Using above mentioned interface one can write sender - receiver IPC application. IPCLib allows passing 32 bit payloads NxN fashion, where N is number of cores in the given SOC. Sysbios based application will have layers as shown in Figure 2 below. Application can use IPCLib interface directly.

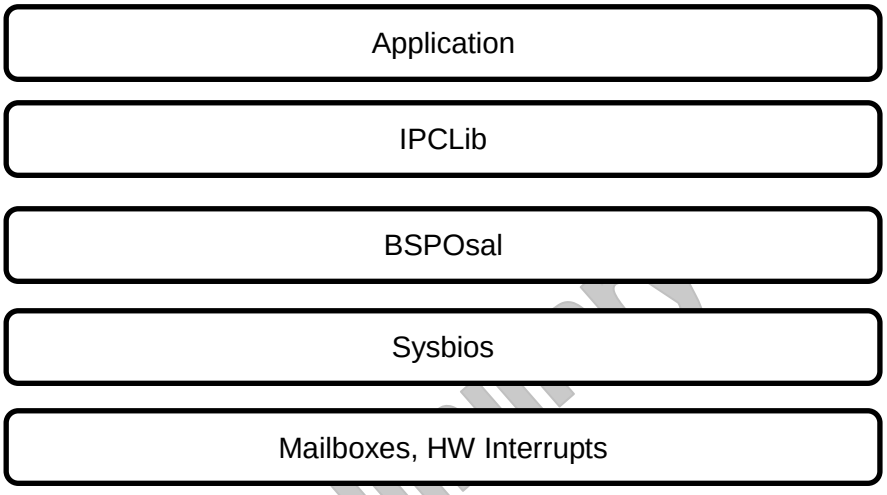

Figure 2 SYS/BIOS based application with IPCLib

Receiver's sequence
1. IPCLib_interruptSetDefaultInitPrm() (optional but advised)
2. Change params as required
3. IPCLib_interruptInit() - with valid params
4. IPCLib_interruptRegisterEvent() - with valid callback function

Sender's sequence
1. IPCLib_interruptSetDefaultInitPrm() (optional but advised)
2. Change params as required
3. IPCLib_interruptInit() - with valid params
4. IPCLib_interruptSendEvent() - with valid payload and event id

Essentially IPCLib allows receivers to register multiple callback functions. Number of callbacks is capped by IPCLIB_MAX_EVENTS defined in interface. The module maintains this callback table and delivers message/payload to right callback upon receiving interrupt. It multiplexes events over single interrupt line from sender M to receiver N. This gives applications more flexibility with minimal system mailbox usage.

Figure 3 below shows sender and receiver sequence and usage of system mailbox for a bidirectional communication between two cores.

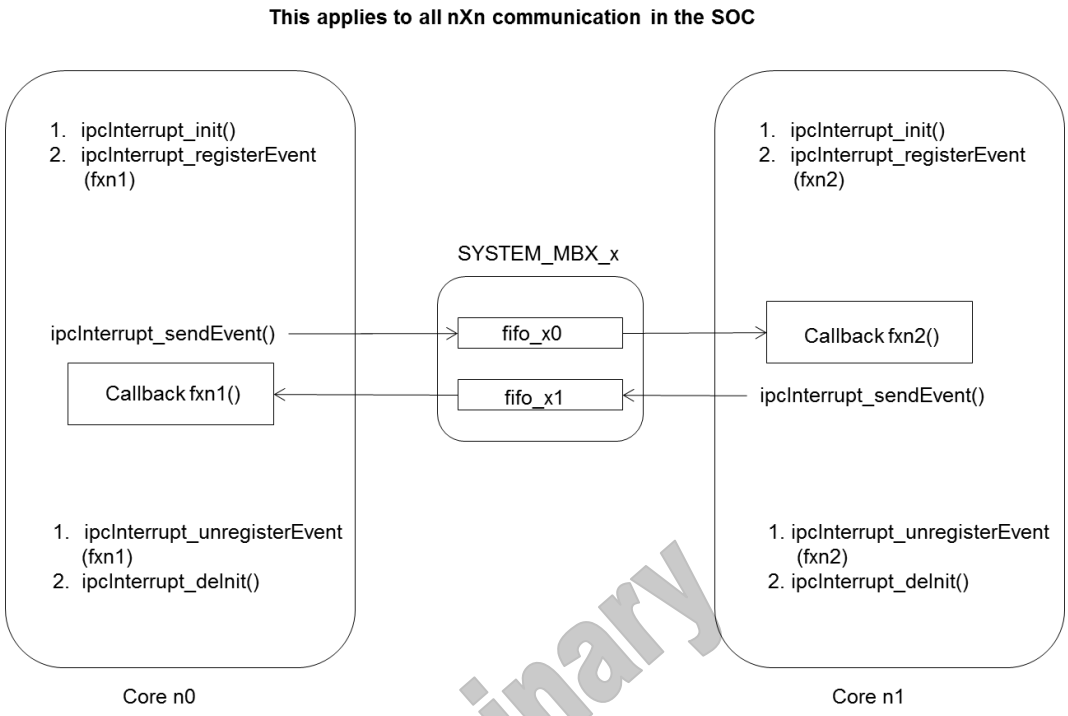

Figure 3 IPCLib sender-receiver sequence

Important note – Synchronization between cores is not taken care by IPCLib. It has to be implemented by application. For reference one can refer Vision SDK section below.

3.3 IPCLib configuration

pdk_XX_XX_XX_XX\packages\ti\drv\ipc_lite\src\<$SOC>\ipclib_interrupt<$SOC>ProcCfg.c
pdk_XX_XX_XX_XX\packages\ti\drv\ipc_lite\src\<$SOC>\ipclib_interrupt<$SOC>MailboxCfg.c
pdk_XX_XX_XX_XX\packages\ti\drv\ipc_lite\src\<$SOC>\ipclib_interrupt<$SOC>XbarCfg.c

These files constitute configuration for IPCLib. Mailbox base addresses, interrupt cross bar settings, interrupt numbers and procIds are defined here and can be changed as per requirement.

In most of the cases for writing an IPCLib application there is no need change these. More information about these files is present in the interface.

3.4 Hardware resource usage for IPCLib

By default IPCLib uses following hardware resources depending upon numValidProc and validProcIds array passed by application during initialization.

Important note – IPCLib doesn't establish connection or use any h/w resources for the processor that is not mentioned in the validProcIds array.

Table 1. Hardware resource usage for IPCLib

Platform	Mailboxes	Interrupts numbers	Cross bar mappings {IRQ_COSSBAR Instance Number, IRQ_CROSSBAR_X}
TDA2xx	System mailboxes MAILBOX5 MAILBOX6 MAILBOX7 MAILBOX8 EVE internal mailboxes EVE1_MBOX0 EVE1_MBOX1 EVE1_MBOX2 EVE2_MBOX0 EVE2_MBOX1 EVE2_MBOX2 EVE3_MBOX0 EVE3_MBOX1 EVE3_MBOX2 EVE4_MBOX0 EVE4_MBOX1 EVE4_MBOX2	EVE as dst 60U, /* EVE1 */ 60U, /* EVE2 */ 60U, /* EVE3 */ 60U, /* EVE4 */ 29U, /* DSP1 */ 30U, /* DSP2 */ 29U, /* IPU1_0 */ 30U, /* IPU2_0 */ 29U, /* A15 */ 30U, /* IPU1_1 */ 30U /* IPU2_1 */ DSP1 as dst 55U, /* EVE1 */ 56U, /* EVE2 */ 58U, /* EVE3 */ 59U, /* EVE4 */ 60U, /* DSP1 */ 60U, /* DSP2 */ 57U, /* IPU1_0 */ 60U, /* IPU2_0 */ 57U, /* A15 */ 57U, /* IPU1_1 */ 61U /* IPU2_1 */ DSP2 as dst 55U, /* EVE1 */ 56U, /* EVE2 */ 58U, /* EVE3 */ 59U, /* EVE4 */ 60U, /* DSP1 */ 60U, /* DSP2 */ 60U, /* IPU1_0 */ 57U, /* IPU2_0 */ 57U, /* A15 */ 61U, /* IPU1_1 */ 57U /* IPU2_1 */	/* Nothing for EVE */ {24U, 284U}, /* DSP1 <- EVE1 */ {25U, 293U}, /* DSP1 <- EVE2 */ {26U, 249U}, /* DSP1 <- IPU1_0, HOST, IPU1_1 */ {27U, 302U}, /* DSP1 <- EVE3 */ {28U, 311U}, /* DSP1 <- EVE4 */ {29U, 257U}, /* DSP1 <- DSP2, IPU2_0 */ {30U, 261U}, /* DSP1 <- IPU2_1 */ {24U, 287U}, /* DSP2 <- EVE1 */ {25U, 296U}, /* DSP2 <- EVE2 */ {26U, 253U}, /* DSP2 <- IPU1_0, HOST, IPU1_1 */ {27U, 305U}, /* DSP2 <- EVE3 */ {28U, 314U}, /* DSP2 <- EVE4 */ {29U, 258U}, /* DSP2 <- DSP1, IPU2_0 */ {30U, 262U}, /* DSP2 <- IPU2_1 */

		IPU1_0 as dst 64U, /* EVE1 */ 65U, /* EVE2 */ 67U, /* EVE3 */ 68U, /* EVE4 */ 66U, /* DSP1 */ 69U, /* DSP2 */ 69U, /* IPU2_0 */ 66U, /* A15 */ 69U, /* IPU1_1 */ 70U /* IPU2_1 */ IPU2_0 as dst 64U, /* EVE1 */ 65U, /* EVE2 */ 67U, /* EVE3 */ 68U, /* EVE4 */ 69U, /* DSP1 */ 66U, /* DSP2 */ 69U, /* IPU1_0 */ 66U, /* A15 */ 70U, /* IPU1_1 */ 19U /* IPU2_1 */ IPU1_1 as dst 71U, /* EVE1 */ 72U, /* EVE2 */ 74U, /* EVE3 */ 75U, /* EVE4 */ 73U, /* DSP1 */ 76U, /* DSP2 */ 76U, /* IPU1_0 */ 76U, /* IPU2_0 */ 73U, /* A15 */ IPU2_1 dst 71U, /* EVE1 */ 72U, /* EVE2 */ 74U, /* EVE3 */ 75U, /* EVE4 */ 76U, /* DSP1 */ 73U, /* DSP2 */ 76U, /* IPU1_0 */ 19U, /* IPU2_0 */ 73U, /* A15 */	{42U, 285U}, /* IPU1_0 / IPU1 <- EVE1 */ {43U, 294U}, /* IPU1_0 / IPU1 <- EVE2 */ {44U, 250U}, /* IPU1_0 / IPU1 <- DSP1, HOST */ {45U, 303U}, /* IPU1_0 / IPU1 <- EVE3 */ {46U, 312U}, /* IPU1_0 / IPU1 <- EVE4 */ {47U, 259U}, /* IPU1_0 / IPU1 <- DSP2, IPU2_0 */ {48U, 263U}, /* IPU1_0 / IPU1 <- IPU2_1 */ {42U, 288U}, /* IPU2_0 / IPU2 <- EVE1 */ {43U, 297U}, /* IPU2_0 / IPU2 <- EVE2 */ {44U, 254U}, /* IPU2_0 / IPU2 <- DSP2, HOST */ {45U, 306U}, /* IPU2_0 / IPU2 <- EVE3 */ {46U, 315U}, /* IPU2_0 / IPU2 <- EVE4 */ {47U, 260U}, /* IPU2_0 / IPU2 <- DSP1, IPU1_0 */ {48U, 264U}, /* IPU2_0 / IPU2 <- IPU1_1 */ {49U, 289U}, /* IPU1_1 <- EVE1 */ {50U, 298U}, /* IPU1_1 <- EVE2 */ {51U, 252U}, /* IPU1_1 <- DSP1, */ {52U, 307U}, /* IPU1_1 <- EVE3 */ {53U, 316U}, /* IPU1_1 <- EVE4 */ {54U, 263U}, /* IPU1_1 <- DSP2, IPU2_0 */ {49U, 289U}, /* IPU2_1 <- EVE1 */ {50U, 298U}, /* IPU2_1 <- EVE2 */ {51U, 256U}, /* IPU2_1 <- DSP2, HOST */ {52U, 307U}, /* IPU2_1 <- EVE3 */ {53U, 316U}, /* IPU2_1 <- EVE4 */ {54U, 264U}, /* IPU2_1 <- DSP1, IPU1_0 */

		A15 as dst (MPU IRQ + 32) 166U, /* EVE1 */ 167U, /* EVE2 */ 169U, /* EVE3 */ 170U, /* EVE4 */ 168U, /* DSP1 */ 173U, /* DSP2 */ 168U, /* IPU1-0 */ 173U, /* IPU2-0 */ 168U, /* IPU1_1 */ 173U /* IPU2_1 */	{127U, 286U}, /* A15 <- EVE1 */ {128U, 295U}, /* A15 <- EVE2 */ {129U, 251U}, /* A15 <- DSP1, IPU1_0, IPU1_1 */ {130U, 304U}, /* A15 <- EVE3 */ {131U, 313U}, /* A15 <- EVE4 */ {134U, 255U}, /* A15 <- DSP2, IPU2_0, IPU2_1 */
TDA3x	System mailboxes MAILBOX2 EVE mailboxes EVE1_MBOX0 EVE1_MBOX1	DSP1 or DSP2 as dst 71U, /* DSP1 */ 71U, /* DSP2 */ 71U, /* IPU1_0 */ 71U, /* IPU1_1 */ 72U, /* EVE1 */ IPU1_0 as dst 55U, /* DSP1 */ 55U, /* DSP2 */ 19U, /* IPU1_1 */ 62U /* EVE1 */ IPU1_1 as dst 56U, /* DSP1 */ 56U, /* DSP2 */ 19U, /* IPU1_0 */ 63U /* EVE1 */ EVE as dst 29U, /* DSP1 */ 29U, /* DSP2 */ 29U, /* IPU1_0 */ 30U, /* IPU1_1 */	{40U, 237U}, /* DSP1 <- IPU1/DSP2 */ {41U, 284U}, /* DSP1 <- EVE1 */ {40U, 238U}, /* DSP2 <- IPU1/DSP1 */ {41U, 285U}, /* DSP2 <- EVE1 */ {33U, 239U}, /* IPU1_0 <- DSP1/DSP2 */ {40U, 286U}, /* IPU1_0 <- EVE1 */ {34U, 240U}, /* IPU1_1 <- DSP1/DSP2 */ {41U, 287U}, /* IPU1_1 <- EVE1 */

TDA2Ex	System mailboxes MAILBOX5 MAILBOX6 MAILBOX7 MAILBOX8	DSP1 as dst 60U, /* DSP1 */ 57U, /* IPU1_0 */ 60U, /* IPU2_0 */ 57U, /* A15 */ 57U, /* IPU1_1 */ 61U /* IPU2_1 */ IPU1_0 as dst 66U, /* DSP1 */ 69U, /* IPU2_0 */ 66U, /* A15 */ 69U, /* IPU1_1 */ 70U /* IPU2_1 */ IPU2_0 as dst 69U, /* DSP1 */ 69U, /* IPU1_0 */ 66U, /* A15 */ 70U, /* IPU1_1 */ 19U /* IPU2_1 */ IPU1_1 as dst 73U, /* DSP1 */ 76U, /* IPU1_0 */ 76U, /* IPU2_0 */ 73U, /* A15 */ IPU2_1 as dst 76U, /* DSP1 */ 76U, /* IPU1_0*/ 19U, /* IPU2_0 */ 73U, /* A15 */ A15 as dst 168U, /* DSP1 */ 168U, /* IPU1-0 */ 173U, /* IPU2-0 */ 168U, /* IPU1_1 */ 173U /* IPU2_1 */	{26U, 249U}, /* DSP1 <- IPU1_0, HOST, IPU1_1 */ {29U, 257U}, /* DSP1 <- DSP2, IPU2_0 */ {30U, 261U}, /* DSP1 <- IPU2_1 */ {44U, 250U}, /* IPU1_0 / IPU1 <- DSP1, HOST */ {47U, 259U}, /* IPU1_0 / IPU1 <- DSP2, IPU2_0 */ {48U, 263U}, /* IPU1_0 / IPU1 <- IPU2_1 */ {44U, 254U}, /* IPU2_0 / IPU2 <- HOST */ {47U, 260U}, /* IPU2_0 / IPU2 <- DSP1, IPU1_0 */ {48U, 264U}, /* IPU2_0 / IPU2 <- IPU1_1 */ {51U, 252U}, /* IPU1_1 <- DSP1, HOST */ {54U, 263U}, /* IPU1_1 <- DSP2, IPU2_0 */ {51U, 256U}, /* IPU2_1 <- HOST */ {54U, 264U}, /* IPU2_1 <- DSP1, IPU1_0 */ {129U, 251U}, /* A15 <- DSP1, IPU1_0, IPU1_1 */ {134U, 255U}, /* A15 <- DSP2, IPU2_0, IPU2_1 */

TDA2Px	System mailboxes MAILBOX5 MAILBOX6 MAILBOX7 MAILBOX8 EVE internal mailboxes EVE1_MBOX0 EVE1_MBOX1 EVE1_MBOX2 EVE2_MBOX0 EVE2_MBOX1 EVE2_MBOX2 EVE3_MBOX0 EVE3_MBOX1 EVE3_MBOX2 EVE4_MBOX0 EVE4_MBOX1 EVE4_MBOX2	EVE as dst 60U, /* EVE1 */ 60U, /* EVE2 */ 29U, /* DSP1 */ 30U, /* DSP2 */ 29U, /* IPU1_0 */ 30U, /* IPU2_0 */ 29U, /* A15 */ 30U, /* IPU1_1 */ 30U /* IPU2_1 */ DSP1 as dst 55U, /* EVE1 */ 56U, /* EVE2 */ 60U, /* DSP1 */ 60U, /* DSP2 */ 57U, /* IPU1_0 */ 60U, /* IPU2_0 */ 57U, /* A15 */ 57U, /* IPU1_1 */ 61U /* IPU2_1 */ DSP2 as dst 55U, /* EVE1 */ 56U, /* EVE2 */ 60U, /* DSP1 */ 60U, /* DSP2 */ 60U, /* IPU1_0 */ 57U, /* IPU2_0 */ 57U, /* A15 */ 61U, /* IPU1_1 */ 57U /* IPU2_1 */ IPU1_0 as dst 64U, /* EVE1 */ 65U, /* EVE2 */ 66U, /* DSP1 */ 69U, /* DSP2 */ 69U, /* IPU2_0 */ 66U, /* A15 */ 69U, /* IPU1_1 */ 70U /* IPU2_1 */	/* Nothing for EVE */ {24U, 284U}, /* DSP1 <- EVE1 */ {25U, 293U}, /* DSP1 <- EVE2 */ {26U, 249U}, /* DSP1 <- IPU1_0, HOST, IPU1_1 */ {29U, 257U}, /* DSP1 <- DSP2, IPU2_0 */ {30U, 261U}, /* DSP1 <- IPU2_1 */ {24U, 287U}, /* DSP2 <- EVE1 */ {25U, 296U}, /* DSP2 <- EVE2 */ {26U, 253U}, /* DSP2 <- IPU1_0, HOST, IPU1_1 */ {29U, 258U}, /* DSP2 <- DSP1, IPU2_0 */ {30U, 262U}, /* DSP2 <- IPU2_1 */ {42U, 285U}, /* IPU1_0 / IPU1 <- EVE1 */ {43U, 294U}, /* IPU1_0 / IPU1 <- EVE2 */ {44U, 250U}, /* IPU1_0 / IPU1 <- DSP1, HOST */ {47U, 259U}, /* IPU1_0 / IPU1 <- DSP2, IPU2_0 */ {48U, 263U}, /* IPU1_0 / IPU1 <- IPU2_1 */

		IPU2_0 as dst 64U, /* EVE1 */ 65U, /* EVE2 */ 69U, /* DSP1 */ 66U, /* DSP2 */ 69U, /* IPU1_0 */ 66U, /* A15 */ 70U, /* IPU1_1 */ 19U /* IPU2_1 */ IPU1_1 as dst 71U, /* EVE1 */ 72U, /* EVE2 */ 73U, /* DSP1 */ 76U, /* DSP2 */ 76U, /* IPU1_0 */ 76U, /* IPU2_0 */ 73U, /* A15 */ IPU2_1 dst 71U, /* EVE1 */ 72U, /* EVE2 */ 76U, /* DSP1 */ 73U, /* DSP2 */ 76U, /* IPU1_0 */ 19U, /* IPU2_0 */ 73U, /* A15 */ A15 as dst (MPU IRQ + 32) 166U, /* EVE1 */ 167U, /* EVE2 */ 168U, /* DSP1 */ 173U, /* DSP2 */ 168U, /* IPU1-0 */ 173U, /* IPU2-0 */ 168U, /* IPU1_1 */ 173U /* IPU2_1 */	{42U, 288U}, /* IPU2_0 / IPU2 <- EVE1 */ {43U, 297U}, /* IPU2_0 / IPU2 <- EVE2 */ {44U, 254U}, /* IPU2_0 / IPU2 <- DSP2, HOST */ {47U, 260U}, /* IPU2_0 / IPU2 <- DSP1, IPU1_0 */ {48U, 264U}, /* IPU2_0 / IPU2 <- IPU1_1 */ {49U, 289U}, /* IPU1_1 <- EVE1 */ {50U, 298U}, /* IPU1_1 <- EVE2 */ {51U, 252U}, /* IPU1_1 <- DSP1, */ {54U, 263U}, /* IPU1_1 <- DSP2, IPU2_0 */ {49U, 289U}, /* IPU2_1 <- EVE1 */ {50U, 298U}, /* IPU2_1 <- EVE2 */ {51U, 256U}, /* IPU2_1 <- DSP2, HOST */ {54U, 264U}, /* IPU2_1 <- DSP1, IPU1_0 */ {127U, 286U}, /* A15 <- EVE1 */ {128U, 295U}, /* A15 <- EVE2 */ {129U, 251U}, /* A15 <- DSP1, IPU1_0, IPU1_1 */ {134U, 255U}, /* A15 <- DSP2, IPU2_0, IPU2_1 */

4 Revision History

Version	Date	Revision History
1.0	Nov 2016	Document First Version
1.1	Dec 2017	Updates based on the Processor SDK Vision 3.x releases.