Implementing a CPM4 PCI Express Interface in the ADM-VA600

Alpha Data Parallel Systems Ltd.

Document number: AD-AN-0143

Document revision: v1.0 / December 15, 2023

1 Introduction

> **Xilinx is now a part of AMD**
>
> The “Xilinx” trademark may be used in this document in order to avoid confusion when referring to Web pages, documentation and software that predates the phasing out of the “Xilinx” brand following its acquisition by AMD Inc. Any references to Xilinx should be interpreted as references to AMD.

The ADM-VA600 is a reconfigurable computing (FPGA) card in the form of a Space VPX (VITA 78) card, capable of operating 1, 2, 4 or 8 lanes of PCI Express (PCIe) at up to Gen 3 speed (approximately 8 Gb/s per lane) on the VPX Data Plane (P1) connector. This document provides guidelines on implementing a PCI Express interface in the ADM-VA600 using the AMD Versal “Control Interfaces & Processing System” IP. This IP provides a complete PCI Express solution with interfaces to fabric and/or Network-on-Chip (NoC).

An assumption that applies throughout this document is that the ADM-VA600 is to be used in a compliant Space VPX system. While Alpha Data may attempt to provide some guidance for a non-compliant system, precise rules cannot be provided without specific knowledge of the system in question.

The VC1902 Adaptive SoC provides two CPM4 blocks (0 & 1), though as explained in Section 1.1, use of CPM4 block 1 for PCI Express is not supported in the ADM-VA600. For further information about enablement options, lane widths and GT Quad assignments, see Section 2.1.

Throughout this document, the terms “PCI Express” and “PCIe” are used interchangeably, and likewise for the acronyms “FPGA” (Field Programmable Gate Array) and “Adaptive SoC” (Adaptive System on Chip). The term PERST# denotes the PCIe Fundamental Reset signal.

1.1 Overview of VPX connector in the ADM-VA600

Figure 1 shows how the Fat Pipes and wafers of the ADM-VA600’s VPX connector are allocated to particular functions.

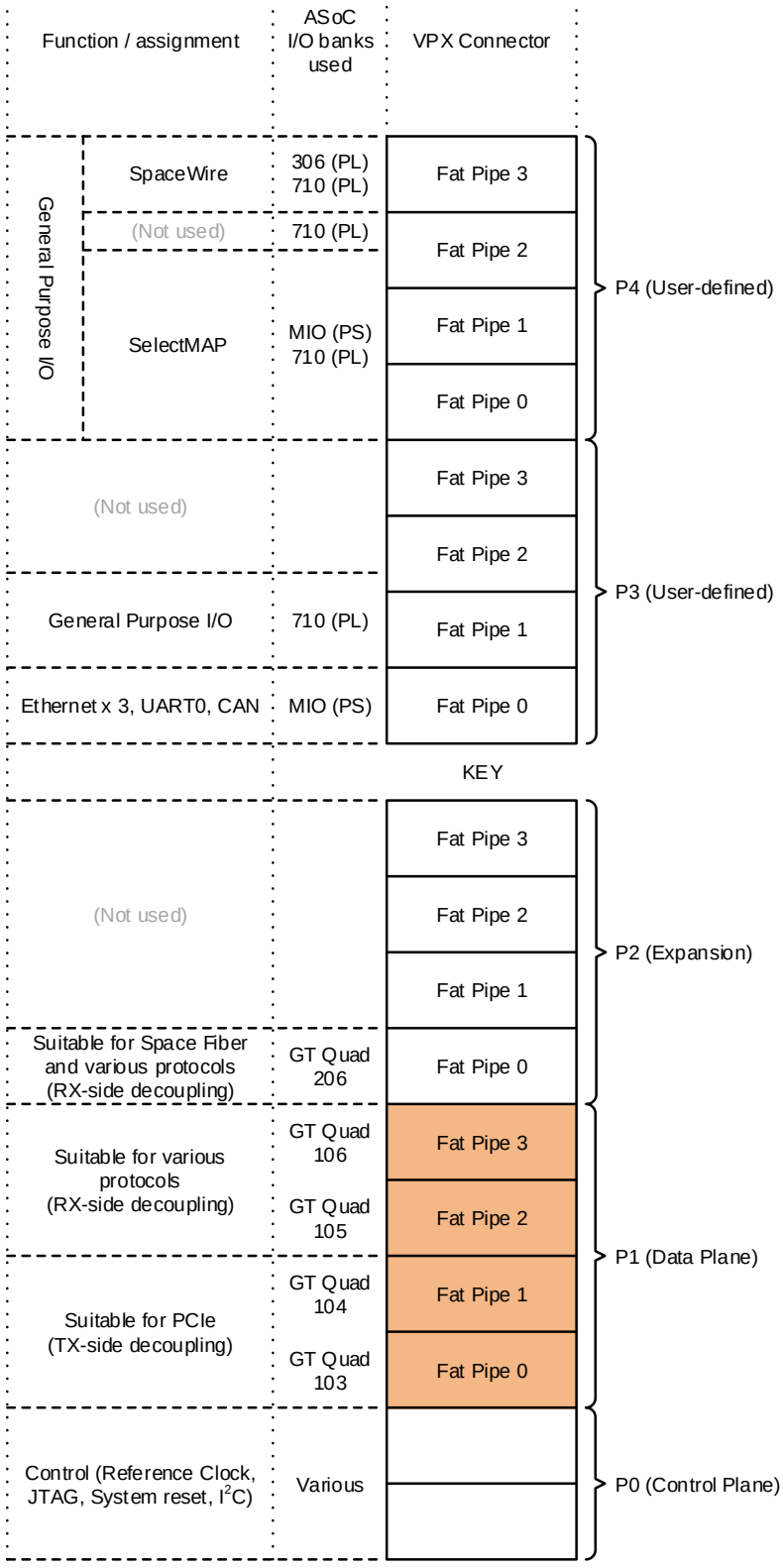

Figure 1: Overview of VPX connector assignments in the ADM-VA600

Fat Pipes 0 and 1 of the VPX P1 connector are suitable for PCI Express because they have decoupling capacitors fitted to the transmit (TX) side of each PCIe lane.

Fat Pipes 2 and 3 of the VPX P1 connector are **not** suitable for PCI Express because they have decoupling capacitors fitted to the receive (RX) side of each lane. They are intended for other communication protocols, most of which require decoupling capacitors on the receive (RX) side of each lane.

Therefore, up to 8 lanes of PCI Express are possible. When taking the GT Quad restrictions of the AMD Versal "Control Interfaces & Processing System" IP into account (see Section 2.1), only CPM4 block 0 can be expected to function reliably in the ADM-VA600.

Use of CPM4 block 1 is not supported in the ADM-VA600

Use of CPM4 block 1 in the ADM-VA600 is neither tested nor supported by Alpha Data because it can only drive GT Quads 105 & 106, corresponding to Fat Pipes 2 and 3 of the VPX P1 connector, which do not have the correct configuration of TX-side decoupling capacitors for PCI Express protocol.

Nevertheless, this document provides information on how to configure CPM4 block 1 to implement a second PCI Express endpoint, though Alpha Data makes no guarantees about the reliable operation of CPM4 block 1 in the ADM-VA600.

When using the ADM-VA600 in a VPX chassis, a CPU card in the same chassis typically drives Fat Pipe 0 for a x4 PCIe link or Fat Pipes 0 & 1 for a x8 PCIe link. However, when using the ADM-VA600-RTM together with the ADM-VA600 in a VPX chassis without a CPU card, it is possible to make a cabled PCIe connection from a standard PC (subject to certain requirements) in order to obtain PCIe connectivity. This arrangement is described in Section 5.

1.2 GTY Quad assignments in the ADM-VA600

Figure 2 shows the I/O banks of the VC1902 Adaptive SoC.

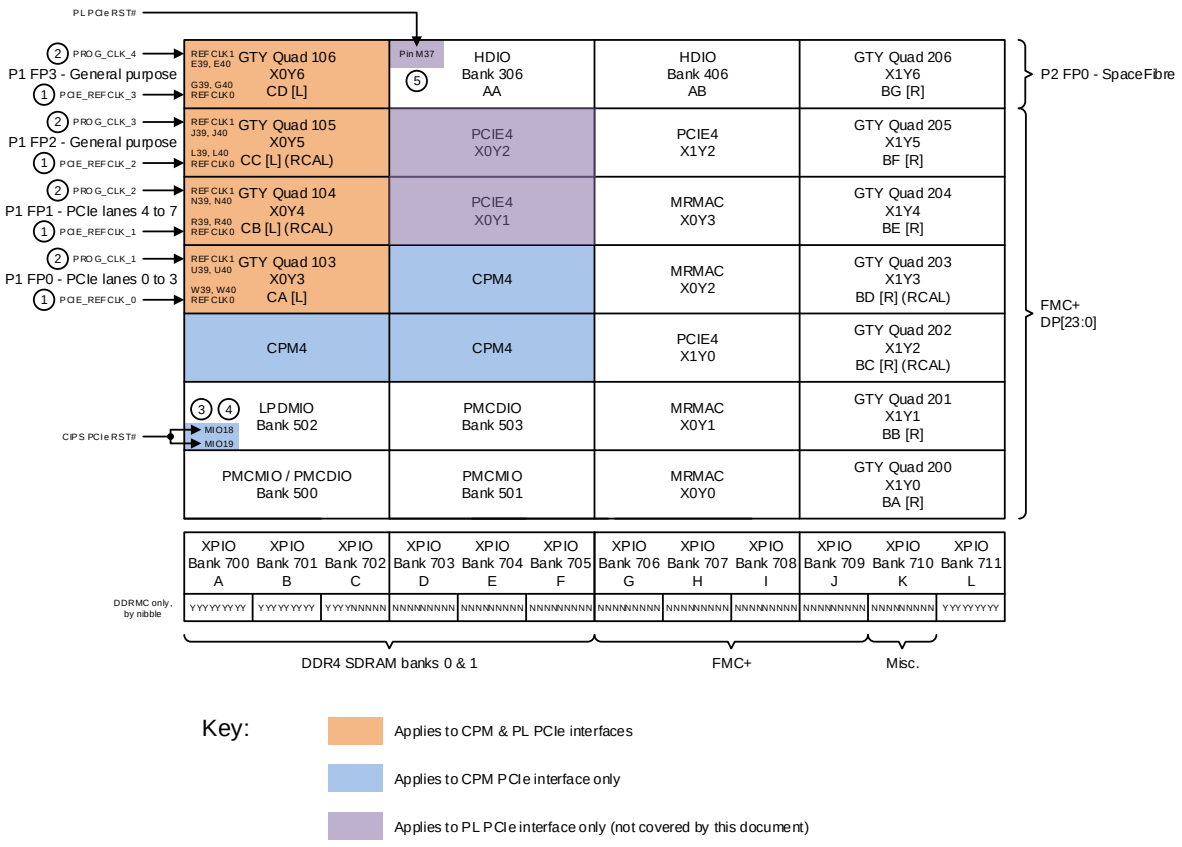

Figure 2: GTY quads in VC1902 Adaptive SoC (VSVA2197 or VSRA2197 package)

With reference to Figure 2 above, the following board features are highlighted:

1. A copy of the 100 MHz PCIe reference clock are provided to each of GT quads 103 to 106, generated locally on the ADM-VA600's circuit board. However, at most two of them are used for CPM PCIe applications - one for CPM4 block 0 and another for CPM4 block 1 (if used).
2. A programmable clock is provided to each of GT quads 103 to 106, generated locally on the ADM-VA600's circuit board. These clocks have a factory default frequency of 200 MHz, but if reprogrammed to either 100, 125 or 250 MHz, could be used as PCIe reference clocks for the CPM4 blocks.
3. A copy of PERST# for CPM4 block 0, which is necessarily routed to pin F35 (PS MIO18 in CIPS).
4. A copy of PERST# for CPM4 block 1, which is necessarily routed to pin G35 (PS MIO19 in CIPS).
5. A copy of PERST# for the PCIE4 primitive, which is routed to fabric pin M37. This cannot be used as the PCIe Fundamental Reset for a CPM4 block.

2 Using the CPM4 block

This section contains information specific to the ADM-VA600 about using the CPM4 block as a PCIe solution.

Information valid for Vivado 2021.1 or later

Screen captures depicting how to set various configuration options are obtained from the "Control Interfaces & Processing System" IP configuration GUI.

Because the configuration GUI of the "Control Interfaces & Processing System" IP underwent major changes between Vivado 2020.3 and 2021.1, the information concerning the "Control Interfaces & Processing System" IP in this document applies only to Vivado 2021.1 or later.

2.1 CPM PCIe lane widths and enablement options

The information in this section is a summary of information presented in AMD PG346.

In the VC1902 Adaptive SoC, the "Control Interfaces & Processing System" contains two CPM4 blocks, which can each operate as a PCI Express Gen 1/2/3/4 x1/x2/x4/x8/x16 endpoint. However, the total number of PCIe lanes in use by one or both CPM4 blocks cannot exceed 16 because only GT Quads 103 to 106 can be used with the CPM4 blocks. In addition, there are restrictions on which GT Quads can be used for each CPM4 block at various lane widths. Figure 3 below illustrates the allowed combinations of CPM4 block 0 and CPM4 block1 and their allowed lane widths.

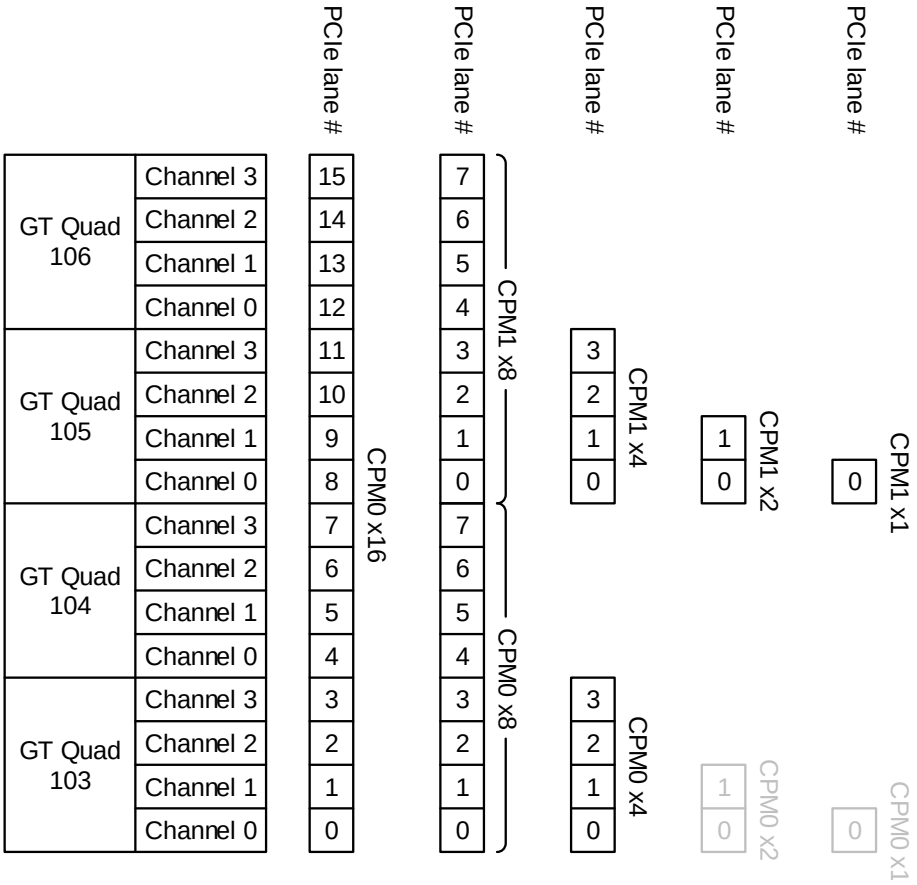

Figure 3: CPM PCIe lane width options

Points to consider when planning a project using CPM4 blocks are:

- As noted in Section 1.1, CPM4 block 1 is not guaranteed to work reliably in the ADM-VA600 because the GTY Quads that it can drive (105 & 106, corresponding to VPX P1 Fat Pipes 2 & 3) do not have the correct configuration of TX-side decoupling capacitors. Use of CPM4 block 1 for PCI Express protocol is not supported by Alpha Data.
- CPM4 block 1 cannot be enabled unless CPM4 block 0 is enabled.
- The lane widths of CPM4 block 0 and CPM4 block 1 can be configured independently. For example, CPM4 block 0 can be configured as x4 width and CPM4 block 1 can be configured as x8 width; this leaves GT Quad 104 unused, but nevertheless the combination is valid.
- CPM4 block 0 can be configured for x4, x8 or x16 width but not x1 or x2 width, represented by the grayed out graphics in Figure 3. The reason for this restriction is not known.
- CPM4 block 0 is “anchored” at channel 0 of GT Quad 103, whereas CPM4 block 1 is “anchored” at channel 0 of GT Quad 105. This cannot be changed, so compared to the PL PCIE4 primitives (also available in Versal Adaptive SoCs) in previous AMD FPGA architectures such as UltraScale+, there is less flexibility in the choice of GT Quads.

  For example, it is not possible to configure the two CPM4 blocks to create two independent x4 PCIe links using GT Quads 103 & 104.
- In the -1MP speed grade of the VC1902 Adaptive SoC, x16 lane width for a CPM4 block is not permitted at Gen 4 speed.
- Only CPM4 block 0 has an associated DMA subsystem, which means it can operate in XDMA, QDMA or “bare PCIe” functional mode. CPM4 block 1 does not have an associated DMA subsystem, and must always operate in “bare PCIe” functional mode.

2.2 Simple CPM PCIe block diagram design

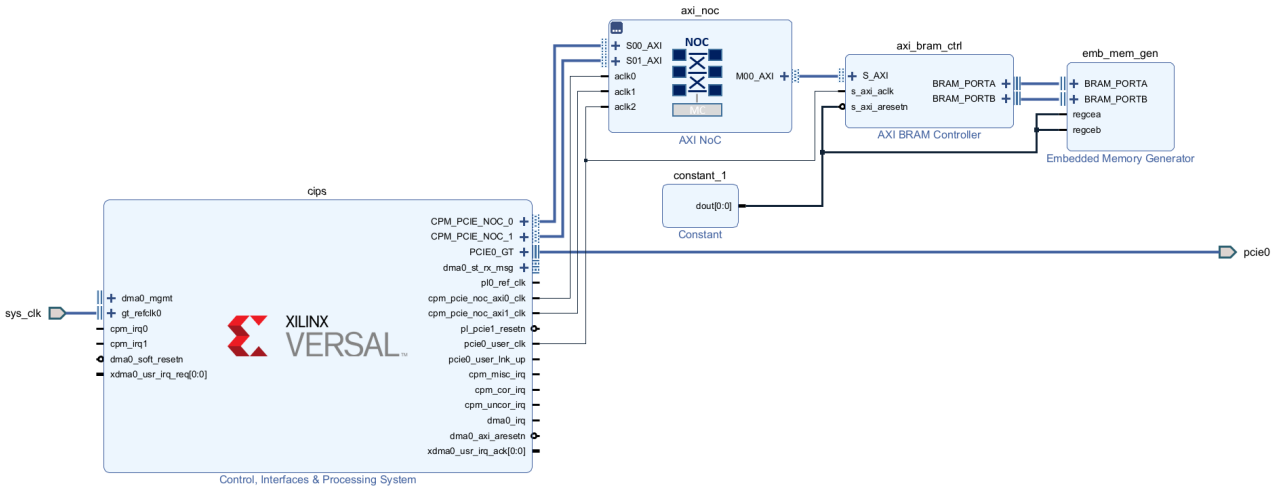

Figure 4: A simple CPM PCIe block diagram design

Guidelines on configuring the CPM PCIe, within the CIPS IP instance, are given in Section 3.

In Figure 4 above, the user-defined logic (the AXI BRAM Controller & Embedded Memory Generator) attached to the AXI NoC instance has no reset signal. The reset signal from the CIPS (`dma0_axi_resetn` when the functional mode is XDMA) is not used. This is a consequence of the fact that, apart from during configuration of the Adaptive SoC, the NoC is not reset once the application is up and running.

While this may be acceptable for some use cases, it means that a PCI Express reset (either Fundamental or Hot) cannot return the user-defined logic to a known state. If the design has not been fully debugged, or is operated in conditions where radiation may flip bits, failing to propagate PCIe reset to user-defined logic may be a problem. The design in Figure 4 does not conform to the principle that a PCIe reset should return a malfunctioning PCIe endpoint to a working state.

Connecting `dma0_axi_resetn` to the user-defined logic does not improve matters, and may cause the NoC to hang up when `dma0_axi_resetn` is asserted: if the reset happens to be asserted while there are outstanding AXI transactions, the state of the NoC and user-defined logic becomes inconsistent and the likely result is an AXI hang.

At a higher level, this issue highlights an apparent conflict between (i) the PCIe reset domain, which extends from some part of the host system and covers the CPM PCIe endpoint, and (ii) the NoC reset domain. In the following sections this reset domain issue is discussed in further detail and a solution is proposed.

2.3 Reset domain issue and resolution

Figure 5 shows simplified view of a system hosting a Versal CPM PCIe endpoint, where shaded regions indicate reset domains.

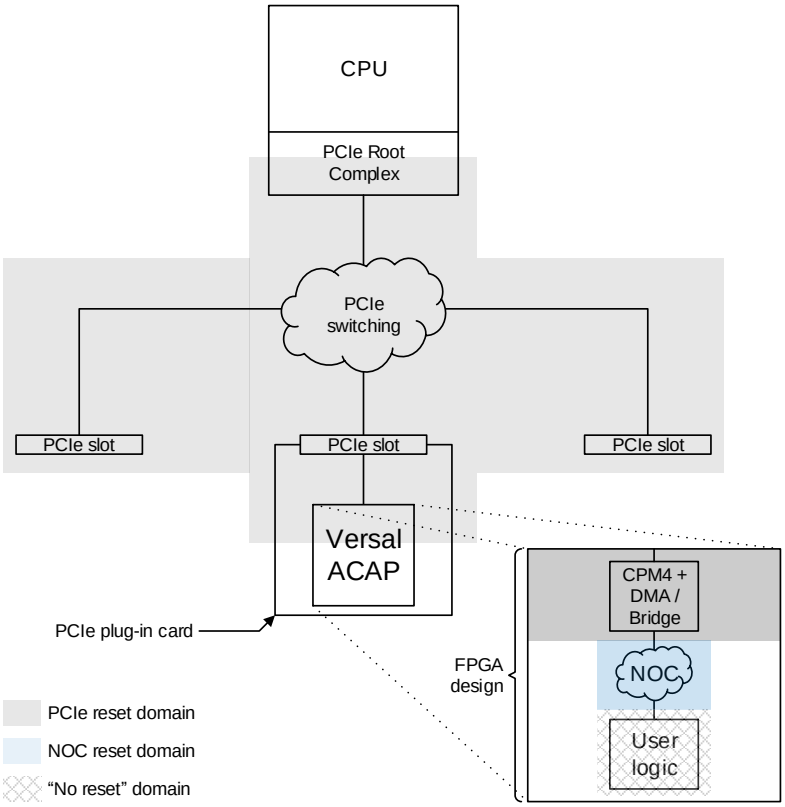

Figure 5: Simplified view of a system hosting a Versal CPM PCIe endpoint

The enlarged view of the Versal Adaptive SoC labeled “FPGA design” in Figure 5 corresponds to Figure 4. In this design, the “NoC reset domain” and “No reset domain” are effectively the same thing, since the NoC is not reset after configuration of the Adaptive SoC is complete. As the term suggests, the “No reset domain” is never reset. This design would work correctly in hardware, but if some event causes a malfunction in the user-defined logic, PCIe reset cannot return it to a working state.

To allow user-defined logic to be reset without risk of hanging up the NoC, an isolator is inserted between the NoC and user-defined logic. The isolator’s purpose, when triggered, is to complete all outstanding AXI transactions from the point of view of the NoC in a protocol-compliant way. Once the isolator has been triggered, it is safe to reset the user-defined logic. Figure 6 shows the same simplified view with the isolator added:

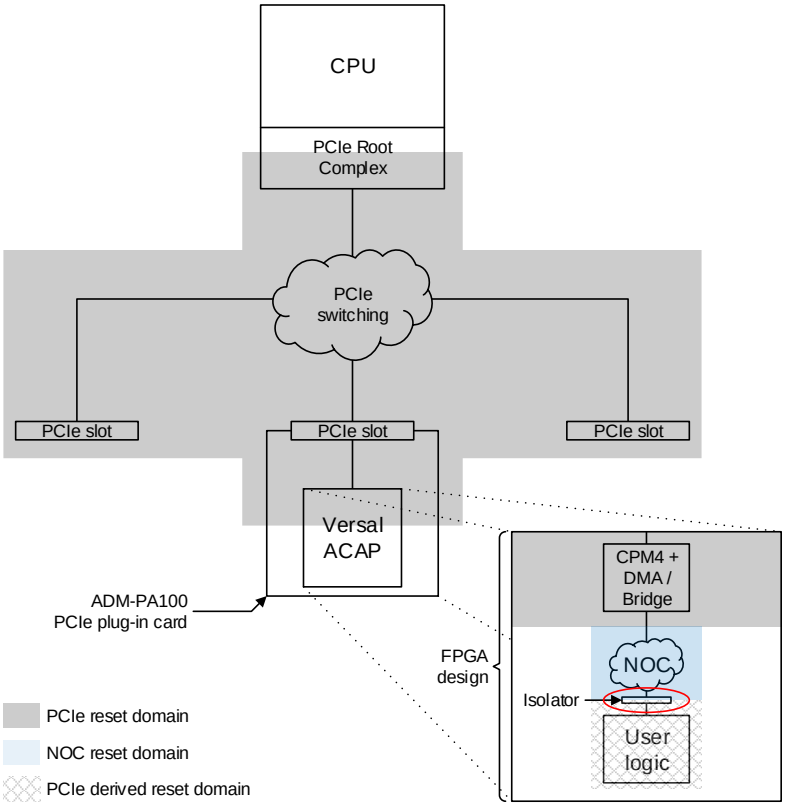

Figure 6: Simplified view of a system hosting a Versal CPM PCIe endpoint, with isolator

The “No reset domain” is now the “PCIe derived reset domain”, meaning that the reset for the user-defined logic is derived from the reset signal output by the CPM4 block, which in turn is derived from PCIe reset (either Fundamental or Hot). The next section presents a concrete example of the “FPGA design” of Figure 6.

2.4 CPM PCIe block diagram design with reset domain resolution

Example FPGA design

The ADM-VA600 PL Examples includes a complete example FPGA design, “ADM-VA600 CPM PCIe Demonstration FPGA Design”, that puts the solution presented in this section into practice.

Figure 7 shows a Vivado block diagram design that places an instance of the AXI Firewall IP between the NoC and user-defined logic.

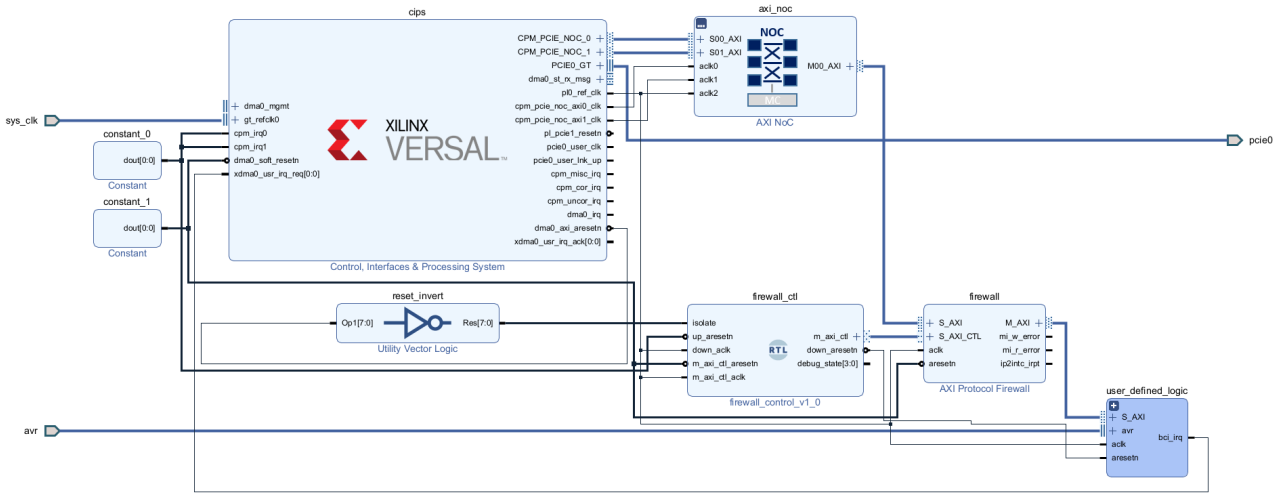

Figure 7: CPM PCIe block diagram design with isolator between reset domains

The AXI Firewall IP has an AXI Lite control interface (`S_AXI_CTL`) and, to put the firewall into a “Block” state where it isolates the NoC from user-defined logic, it requires at least one AXI write on its control interface. Likewise, unblocking the firewall requires at least one AXI write on its control interface.

The purpose of the firewall control module (`firewall_ctl`) is to convert transitions on `dma0_axi_aresetn` into the appropriate register writes on the firewall’s control interface. The firewall controller’s behavior can be summarised as the following FSM:

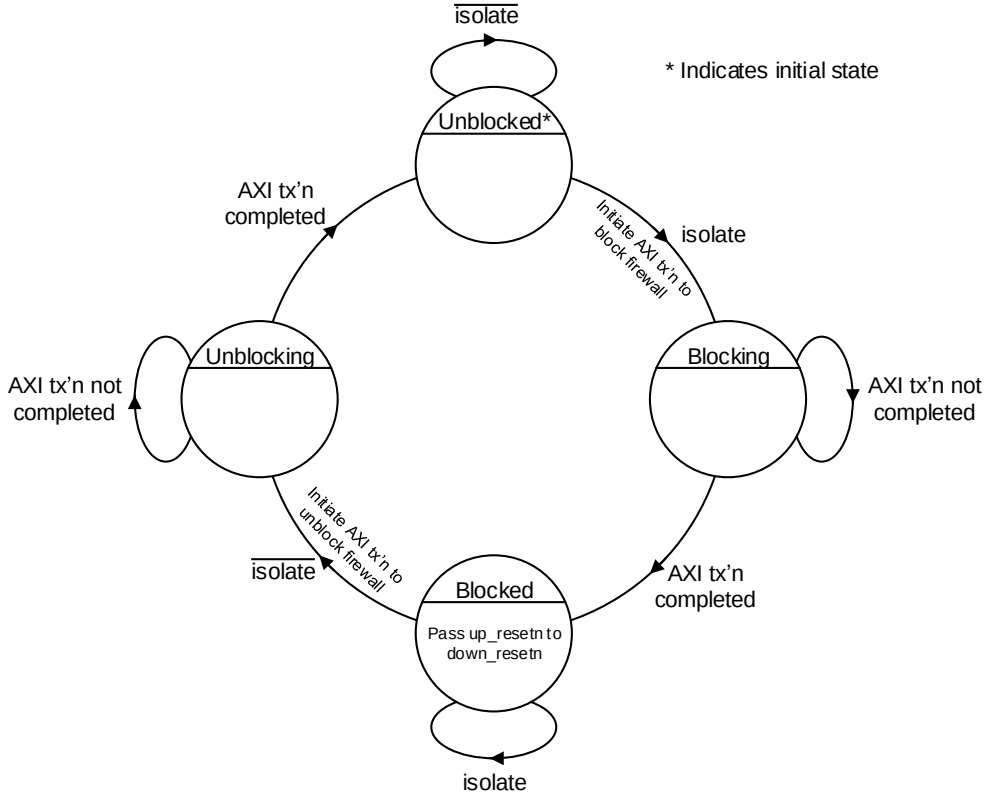

Figure 8: Firewall controller FSM

- When `dma0_axi_aresetn` transitions to low (asserted), the firewall controller writes to registers in the firewall so that the firewall enters the “Block” state. When in the “Block” state, the firewall provides protocol-compliant completions for any outstanding AXI transactions on its `S_AXI` interface and it ignores its `M_AXI`

interface.

- Whenever the firewall is in the “Block” state, the firewall controller passes the upstream reset (`up_aresetn`) through to the downstream reset (`down_aresetn`). The downstream reset is the reset for the user-defined logic. In Figure 7, `up_aresetn` is tied low (asserted) so that the user-defined logic is held in reset while the firewall is in the “Block” state.
- When `dma0_axi_aresetn` transitions to high (deasserted), the firewall controller writes to registers in the firewall so that the firewall exits the “Block” state. When not in the “Block” state, the firewall forwards AXI commands from its `S_AXI` interface to its `M_AXI` interface and forwards AXI responses in the other direction.

By guaranteeing that the NoC receives protocol-compliant completions on its AXI interfaces even when the user-defined logic is reset, the risk of hanging up the NoC is eliminated. Furthermore, the user-defined logic is properly reset by a PCI Express reset (Hot or Fundamental).

NOTE: There is no straightforward way to reset the NoC from the PL, but the NoC is assumed to be “reliable” provided that there are no protocol violations on its interfaces.

2.5 Caveats and pitfalls

2.5.1 `pcie0_user_clk` stops when PERST# is asserted

When the host asserts PERST# (`pcie0_user_clk`) stops until some point after the reset is deasserted. Internally, the CIPS IP makes use of a phase-locked loop (PLL), which is indirectly reset by PERST#, to generate `pcie0_user_clk`.

This means that if `pcie0_user_clk` is used as the clock for user-defined logic and the fabric side of the NoC, any logic that is designed to provide protocol-compliant completions to the NoC in order to complete outstanding AXI transactions while PERST# is asserted cannot begin to do so until PERST# is deasserted and `pcie0_user_clk` resumes toggling.

For this reason, a free-running clock, or at least one that is not affected by PERST#, may be a better choice for clocking user-defined logic in some applications. For example, in Figure 7, `pl0_ref_clk` (from the CIPS IP) is used.

3 How to customize the CPM PCIe within the CIPS IP

3.1 Starting point

The starting point for the instructions in this section is a Vivado Block Diagram that includes an instance of the CIPS IP which has not yet been customized in any way.

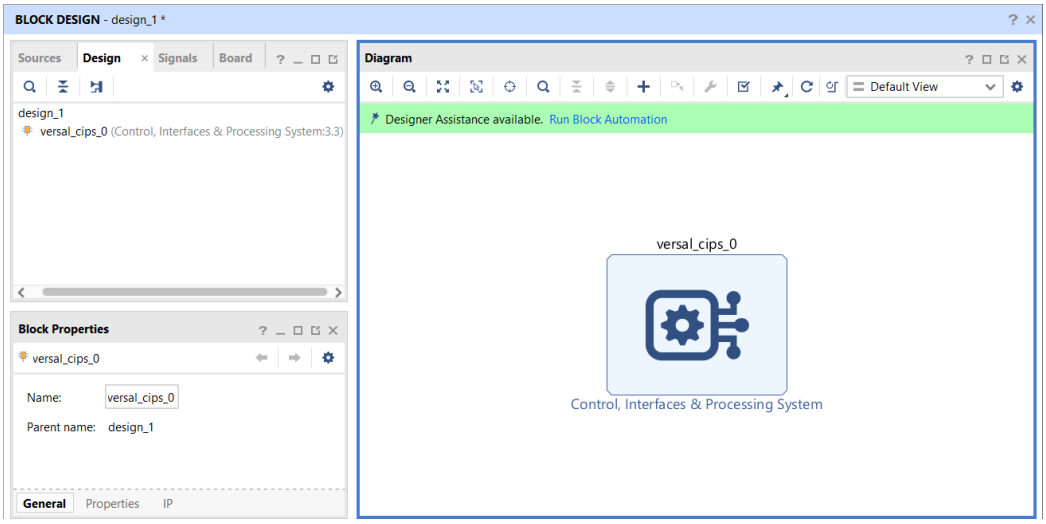

Figure 9: Starting point for CIPS customization

NOTE: Customizing a CIPS IP instance may prevent ***Run Block Automation...*** appearing in the right-click (context-sensitive) menu that appears for the IP. If it does not appear, it may be because it has been customized already. In that case, a new CIPS IP instance must be created and the old one deleted.

Fully customizing the CIPS IP to enable one or both CPM4 block block is a three-step process:

1. Ensure that certain critical settings, namely the PS reference clock frequency and the PS MIO bank I/O voltages, are correctly set for the ADM-VA600 in the PS/PMC module of the CIPS IP. This is described in Section 3.2.
2. Customize the CPM module of the CIPS IP. This is described in Section 3.3.
3. Complete customization of PS/PMC settings that were not available before customizing the CPM module of the CIPS IP. This is described in Section 3.4.

3.2 PS/PMC customization, part 1

Before the CPM module of the CIPS can be customized, it is essential to ensure that certain PS/PMC settings are correctly configured. From the starting point of Figure 9, the designer now has the choice of:

(a) Using Vivado Block Automation, in conjunction with the ADM-VA600 board file, to configure the PS/PMC settings, as described in Section 3.2.1.

(b) Using a preset from the ADM-VA600 board file to configure the PS/PMC settings, as described in Section 3.2.2.

(c) Configuring individual PS/PMC settings, as described in Section 3.2.3. This option may be appropriate if, for some reason, none of the presets defined in the ADM-VA600 board file are suitable for the application.

Alpha Data recommends option (a) or (b) as this eliminates many opportunities for error. The board file for the ADM-VA600 can be downloaded from the AMD Board Store when creating a new Vivado project and used as the board part for the project. An advantage of using the board file is that it provides various IP presets that correctly configure the PS/PMC module of the CIPS IP specifically for the ADM-VA600. In particular, the IP presets correctly configure:

- PS reference clock frequency - 50 MHz on ADM-VA600.
- Boot mode settings - JTAG, SD card, SelectMap & dual QSPI on ADM-VA600.
- PS I/O interfaces - Ethernet, I^2C etc.
- PS I/O voltages - set correctly for ADM-VA600.

For instructions on downloading the ADM-VA600 board file, see AD-AN-0142, “ADM-VA600 Board File”. Once you have created a Vivado project with the ADM-VA600 as the board part, there are two main ways to configure the CIPS IP with a suitable board file preset:

3.2.1 Using Block Automation to configure the PS/PMC

Right-clicking on a new CIPS IP instance and selecting ***Run Block Automation...*** results in the following dialog box appearing:

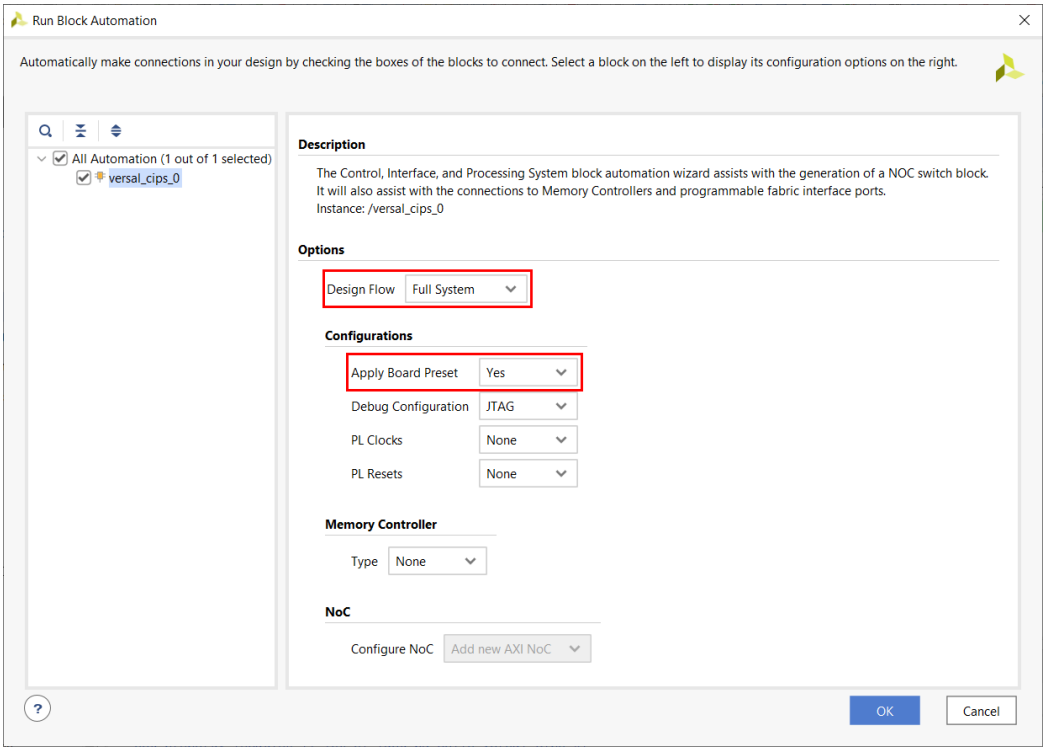

Figure 10: CIPS Block Automation dialog

Other than ensuring that ***Design Flow*** is ***Full System*** and ***Apply Board Preset*** is ***Yes***, the settings on this dialog can be set according to application requirements. Click the ***OK*** button to proceed with Block Automation.

The effect of running Block Automation on the CIPS IP is to apply the “board preset”, which is a particular configuration that is a useful starting point for most applications. Most importantly, the PS I/O voltages are set correctly and the PS reference clock frequency is set correctly for the ADM-VA600.

Having run Block Automation, it is now possible to proceed to customizing the CPM module of the CIPS IP, as described in Section 3.3

3.2.2 Configuring the PS/PMC with a board file preset

The ADM-VA600 board file includes several presets for the CIPS IP that are intended as useful starting points for various applications. They can be applied in the customization GUI for the CIPS IP by selecting a board preset, as shown in Figure 11 below:

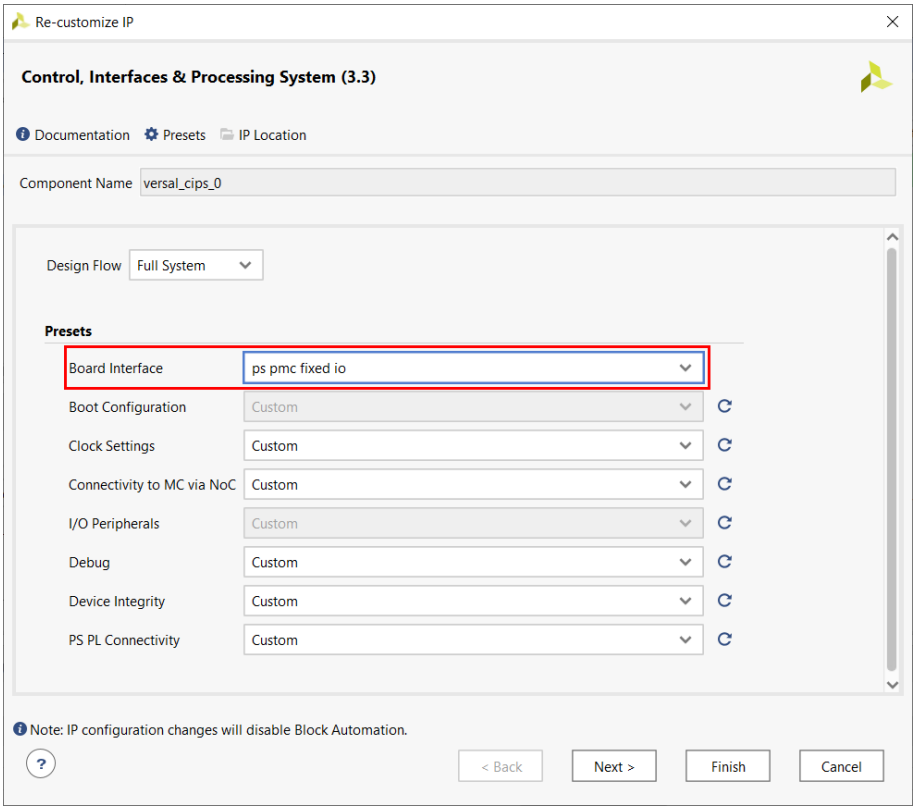

Figure 11: Selecting a preset for the CIPS IP

The ***ps pmc fixed io*** preset is the same one as used for Block Automation, but there are others; for example, ***ps pmc fixed io linux*** additionally enables some peripherals needed for running Petalinux in the ARM Cortex A72 CPU cores. For more details on the CIPS IP presets defined by the ADM-VA600 board file, refer to AD-AN-0142, “ADM-VA600 Board File”.

After selecting the preset in the above dialog, click the ***Next*** button. It is now possible to proceed to customizing the CPM module of the CIPS IP, as described in Section 3.3.

3.2.3 Configuring the PS/PMC without using a board file

Presets selection

On customizing an instance of the CIPS IP, the designer is presented with the following Wizard:

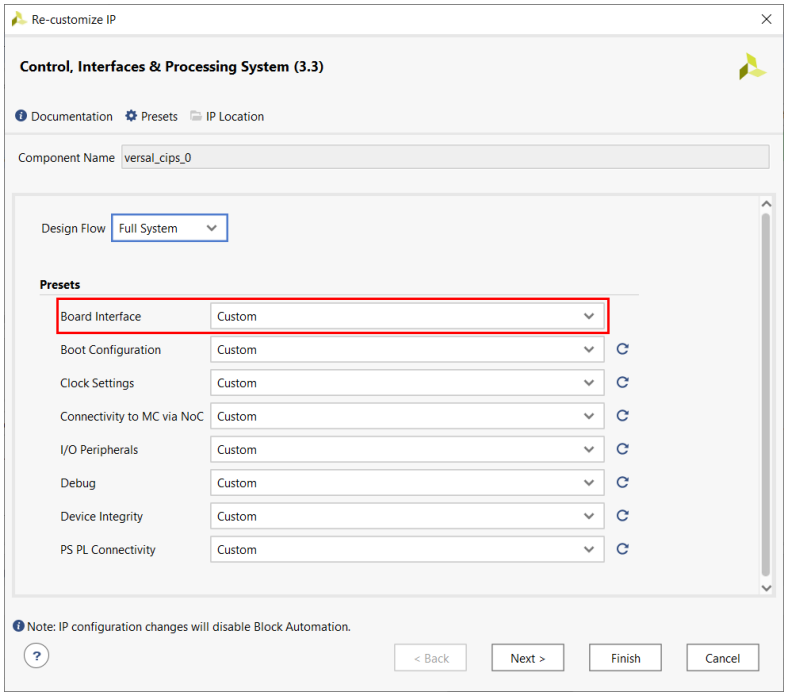

Figure 12: CIPS presets selection

If a board part has been set as the part for the Vivado project, the ***Board Interface*** control appears. If so, ensure that ***Board Interface*** is set to ***Custom***. If a board part has not been set for the project, this control is hidden and no action need be taken.

There is nothing else to do on this page except to click ***Next*** and proceed to configure the PS/PMC.

Module selection dialog

The “modules” page of the Wizard allows the user to choose to configure either the PS/PMC or the CPM:

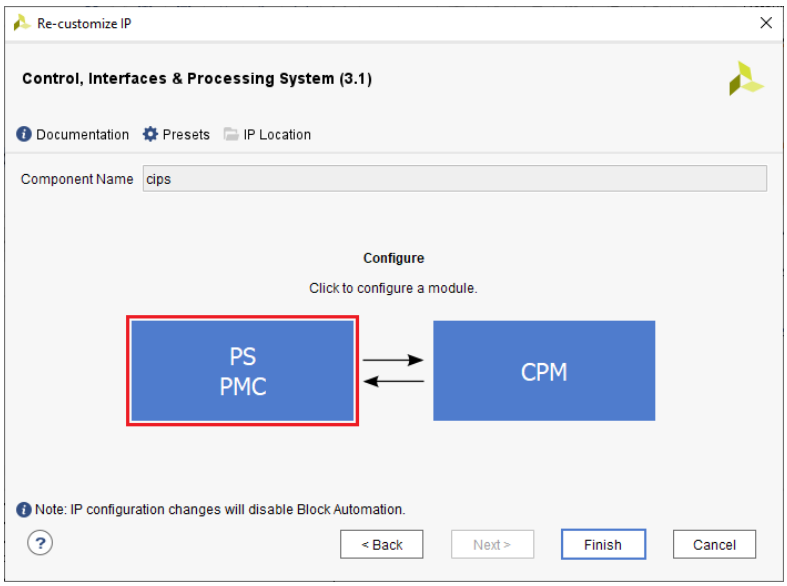

Figure 13: CIPS module selection highlighting PS/PMC

Begin PS/PMC configuration by clicking ***PS PMC***.

Clocking

Because the ADM-VA600 uses a 50 MHz reference clock for the PS/PMC, visit the ***Clocking*** tab and ensure this is set correctly:

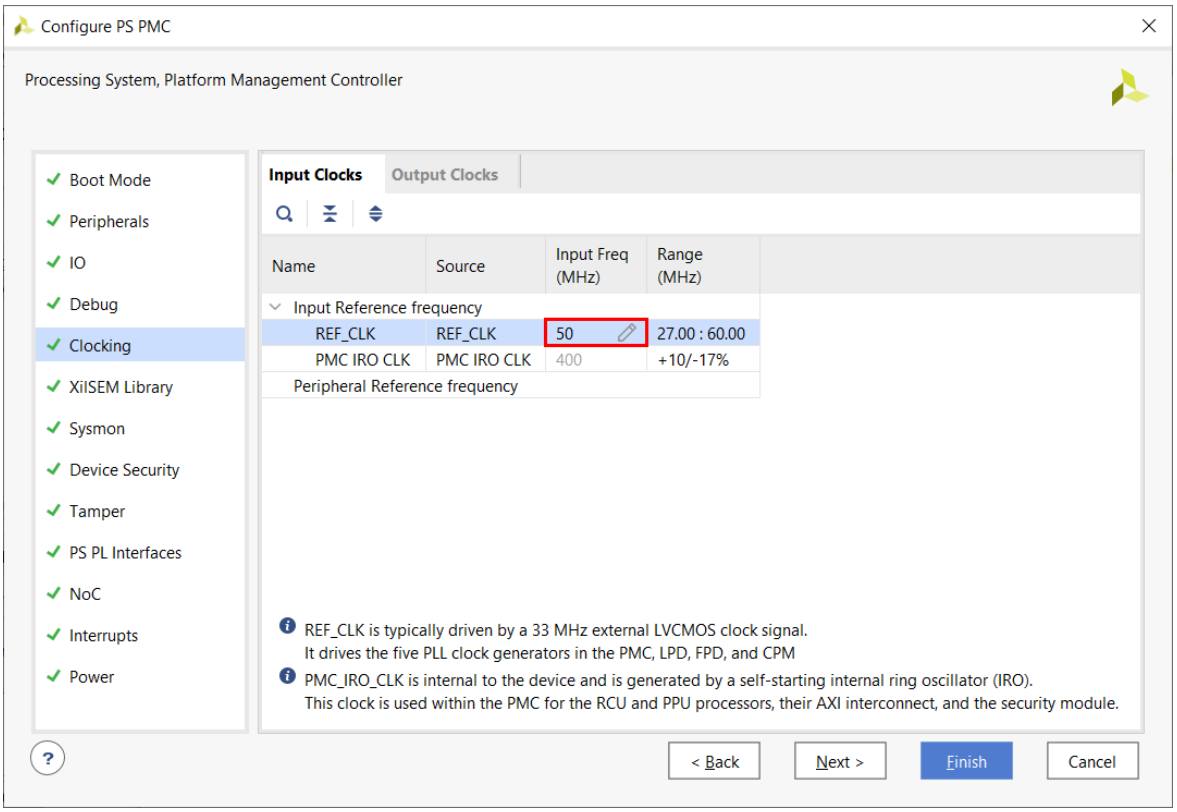

Figure 14: Setting PS/PMC reference clock frequency to 50 MHz

Power

Visit the ***Power*** tab and ensure that the I/O voltage for all four PS I/O banks is set to 3.3V:

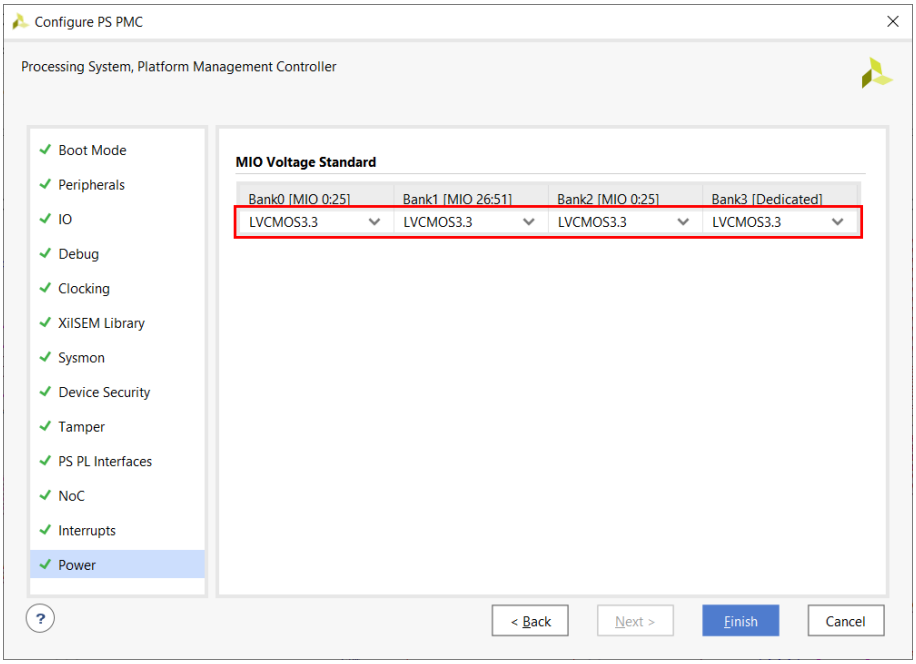

Figure 15: Setting PS/PMC I/O bank voltages to 3.3V

This concludes the first stage of PS/PMC configuration; click ***Finish*** to return to the “modules” page of the Wizard.

3.3 CPM Customization

CPM customization must be performed by selecting ***CPM*** on the “modules” page of the Wizard:

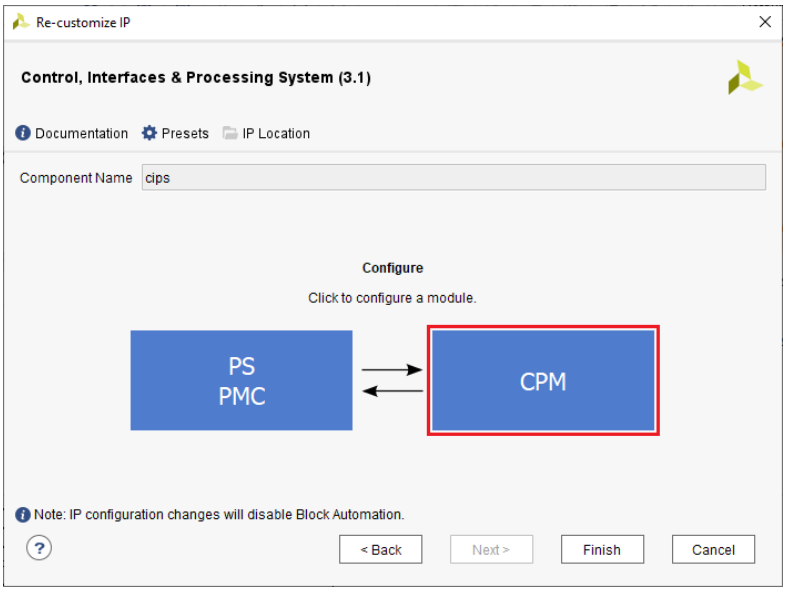

Figure 16: CIPS module selection highlighting CPM

3.3.1 CPM enablement

At the top level of CPM configuration, CPM4 block 0 must be enabled and CPM4 block 1 may be enabled if the application requires it. Here, the choice is made whether CPM4 block 0 (a) operates in “bare” PCIe mode or (b) has a PCIe-to-AXI Bridge and/or DMA engines “bolted on” (AXI Bridge, XDMA or QDMA):

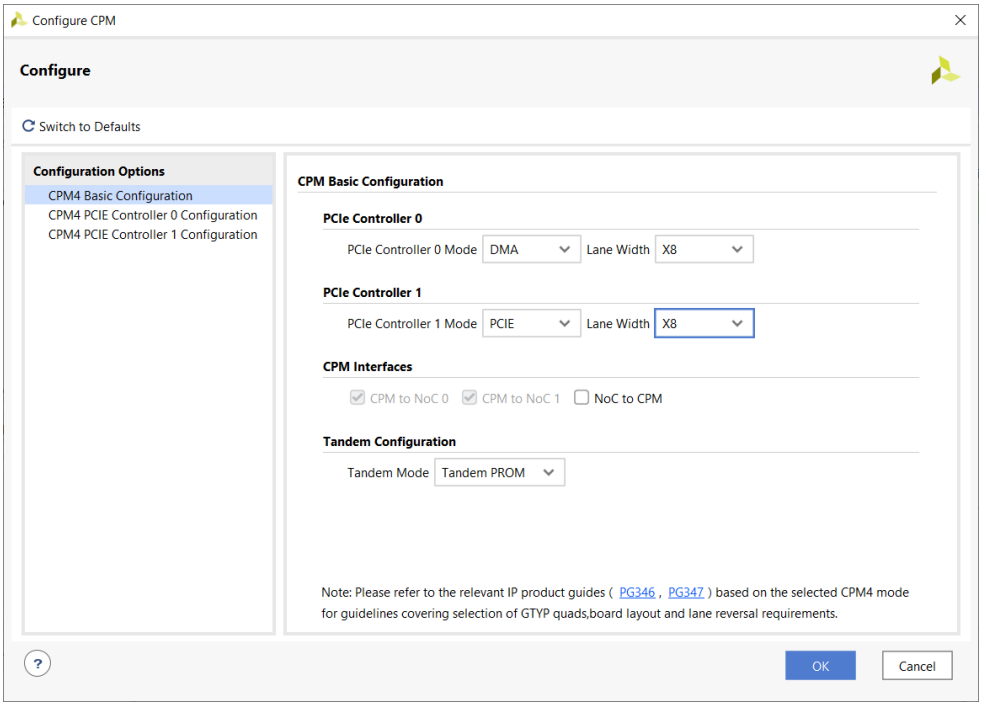

Figure 17: CPM controller mode selection

PCIe Controller 0 (CPM4 block 0)

Set ***Lane Width*** for ***PCIe Controller 0*** to ***X1***, ***X2***, ***X4***, ***X8*** or ***X16*** according to application requirements.

NOTE: Setting ***Lane Width*** for CPM4 block 0 to ***X16*** means that CPM4 block 1 cannot be used. See Section 2.1 for information about enablement options and PCIe lane widths.

Depending on application requirements, set ***PCIe Controller 0 Mode*** to either ***DMA*** or ***PCIe***:

- Selecting ***DMA*** means that AXI Bridge, XDMA or QDMA functional mode can be selected in the ***Basic*** tab of CPM4 block 0 configuration; see Section 3.3.2 below.
- Selecting ***PCIe*** means that CPM4 block 0 operates in “bare” PCIe functional mode.

PCIe Controller 1 (CPM4 block 1)

Set ***Lane Width*** for ***PCIe Controller 1*** to ***X1***, ***X2***, ***X4*** or ***X8*** according to application requirements.

Depending on application requirements, set ***PCIe Controller 1 Mode*** to either ***None*** or ***PCIe***:

- Selecting ***PCIe*** means that CPM4 block 1 operates in “bare” PCIe functional mode.
- Selecting ***None*** means that CPM4 block 1 is disabled. Note that this will be the only choice if ***Lane Width*** for CPM4 block 0 is ***X16***.

CPM Interfaces

The ***CPM to NoC 0*** & ***CPM to NoC 1*** checkboxes are affected by the ***PCIe Controller 0 Mode*** setting:

- If ***PCIe Controller 0 Mode*** is set to ***DMA***, then both the ***CPM to NoC 0*** & ***CPM to NoC 1*** checkboxes will be inoperable and checked. This is because in AXI Bridge, XDMA or QDMA functional mode, the data plane interface for CPM4 block 0 is a pair of NoC connections.
- If ***PCIe Controller 0 Mode*** is set to ***PCIe***, then ***CPM to NoC 0*** & ***CPM to NoC 1*** should be unchecked. In that case, the data plane interface for CPM4 block 0 is a pair of streaming interfaces which exchange PCIe packets with user-defined logic in the fabric.

The data plane interface for CPM4 block 1, if enabled, is always a pair of streaming interfaces which exchange PCIe packets with user-defined logic in the fabric. This is because CPM4 block 1 does not have an associated DMA subsystem.

Tandem Configuration

Alpha Data recommends setting ***Tandem Mode*** to either ***Tandem PROM*** or ***Tandem PCIe***, so that the design satisfies the PCIe "boot time" requirement:

- If there is no requirement for field upgradeability, select ***Tandem PROM***.
- For a field-upgradeable design, select ***Tandem PCIe***. This means that after the host system boots, software must supply the remainder of the configuration data for the Adaptive SoC via configuration space registers in the CPM PCIe endpoint. Until that is achieved, non-configuration accesses to the CPM PCIe endpoint will fail.

The following sections highlight important options for CPM4 PCIE Controllers 0 (and 1, if enabled) in various tabs of the configuration GUI. If a particular tab is not covered below (for example, the ***PCIe: DMA*** tab when in XDMA functional mode), it means that the options on that tab are at the discretion of the designer and can be set according to application requirements.

3.3.2 CPM4 PCIE Controller 0 Configuration - Basic tab

Figure 18 below highlights the options that are of relevance when using CPM4 block 0 in the ADM-VA600:

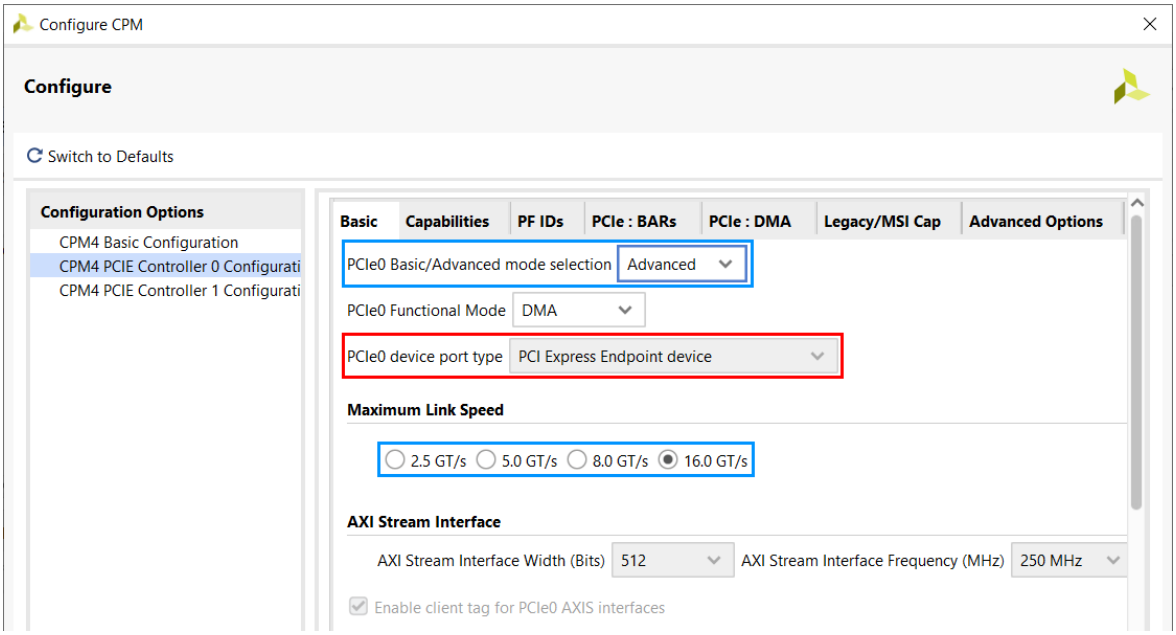

Figure 18: CPM controller 0 Basic tab (upper)

Set ***PCIe0 Basic/Advanced mode selection*** to ***Advanced***. This makes the ***Reference Clock Frequency (MHz)*** and ***Enable Lane Reversal*** options visible so that they can be set correctly (see below).

PCIe0 device port type must be ***PCI Express Endpoint Device***.

NOTE: It is conceivable that the ADM-VA600 could be used as a Root Complex in a custom motherboard, but this use case is outside the scope of this document.

Maximum Link Speed may be one of ***2.5 GT/s***, ***5 GT/S***, ***8 GT/s*** or ***16 GT/s*** according to application requirements and the capabilities of the VPX backplane (connector speed rating etc.). Note that if the PCIe lane width selected for CPM4 block 0 is ***X16***, the ***16 GT/s*** option is not available.

Figure 19 shows two more options that must be set a particular way:

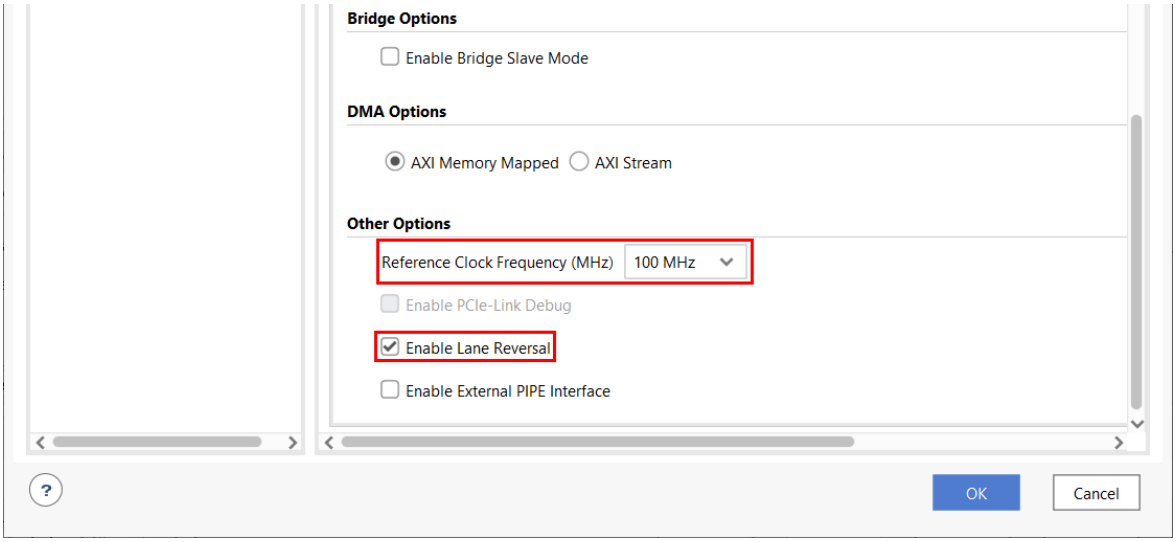

Figure 19: CPM controller 0 Basic tab (lower)

NOTE: The options highlighted in Figure 19 only appear when ***PCIe0 Basic/Advanced mode selection*** is ***Advanced***.

Reference Clock Frequency (MHz) must be set to ***100 MHz*** unless the `PCIE_REFCLK_0` clock has been changed from its factory default frequency of 100 MHz.

Enable Lane Reversal should be checked so that if plugged into a VPX system that reverses the PCIe lanes, the CPM PCIe endpoint will work regardless.

The rest of the options on the ***Basic*** tab, including ***PCIe0 Functional Mode***, are at the discretion of the designer and may be set according to application requirements.

3.3.3 CPM4 PCIE Controller 0 Configuration - Capabilities tab

Only one option on the ***Capabilities*** tab is critical to correct operation of the CPM4 block in the ADM-VA600, namely ***Enable Slot Clock Configuration***. Figure 20 happens to show the ***Capabilities*** tab when the functional mode is QDMA, but this option is present in all functional modes:

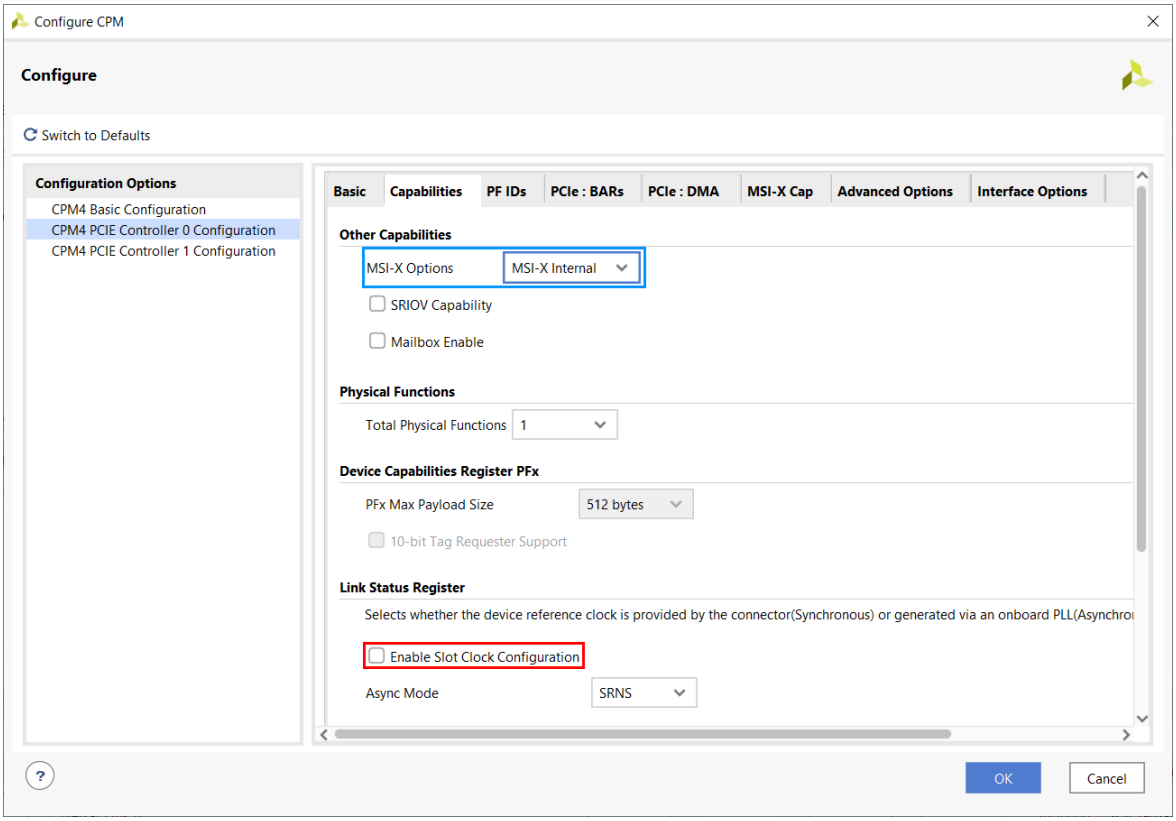

Figure 20: CPM controller 0 Capabilities tab

MSI-X Options may be set to either ***MSI-X Internal*** or ***None.***. Alpha Data recommends setting this option to ***MSI-X Internal***, which causes the ***MSI-X Cap*** tab to become visible (see below).

The ***Enable Slot Clock Configuration*** checkbox must be ***unchecked***, because the CPM PCIe endpoint uses a locally generated reference clock, namely `PCIE_REFCLK_0`, rather than one from the VPX P1 connector.

The rest of the options on the ***Capabilities*** tab are at the discretion of the designer and may be set according to application requirements.

3.3.4 CPM4 PCIE Controller 0 Configuration - MSI-X Cap tab

Alpha Data recommends enabling MSI-X capabilities in the CPM PCIe endpoint, though strictly speaking, it is optional. Figure 21 shows the ***MSI-X Cap*** tab with MSI-X enabled:

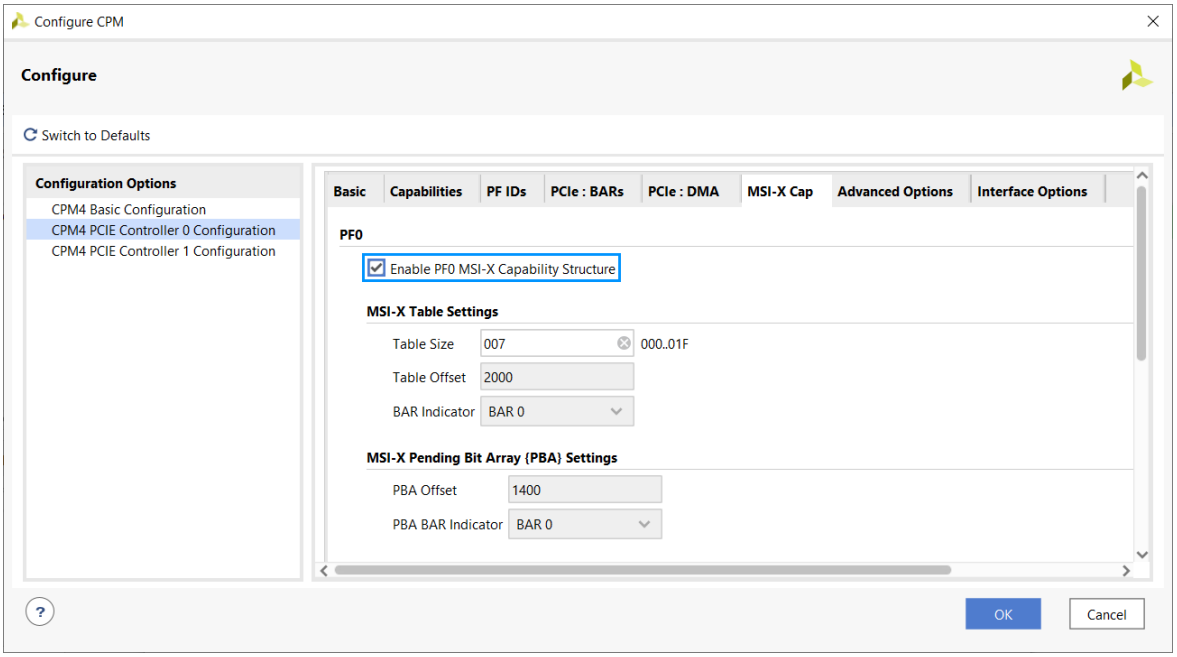

Figure 21: CPM controller 0 MSI-X Cap tab

Use of MSI-X can improve performance in x86-64 platforms because such platforms typically do not allocate more than one MSI (non-X) vector to any given PCIe endpoint. When the host allocates a single MSI vector to the PCIe endpoint, all sources of interrupt in the PCIe endpoint are forced to share a single hardware interrupt vector, resulting in less efficient interrupt servicing. MSI-X, by contrast, appears to be well supported on x86-64 hardware, with hosts typically allocating the requested number of MSI-X vectors to a PCIe endpoint.

Note that for the AXI Bridge functional mode, it is the designer's responsibility to set the options for ***MSI-X Table Settings*** and ***MSI-X Pending Bit Array {PBA} Settings*** correctly. This is mainly a matter of choosing a BAR (Base Address Register) to host the MSI-X vector table and Pending Bit Array (PBA), and choosing suitable offsets within that BAR. The AXI Bridge itself implements the vector table and PBA; it is not implemented by the designer.

Additionally, for the QDMA functional mode, it is the designer's responsibility to determine how many MSI-X vectors are required. This number depends upon the number of interrupt sources in the design. If there are more than 32 interrupt sources, sharing of interrupt vectors is inevitable.

3.3.5 CPM4 PCIE Controller 1 Configuration - Basic tab

CPM4 block 1 supports only "bare PCIe" functional mode, and the set of available options on the ***Basic*** tab reflects this fact. Figure 22 below highlights the options that are of relevance when using CPM4 block 1 in the ADM-VA600:

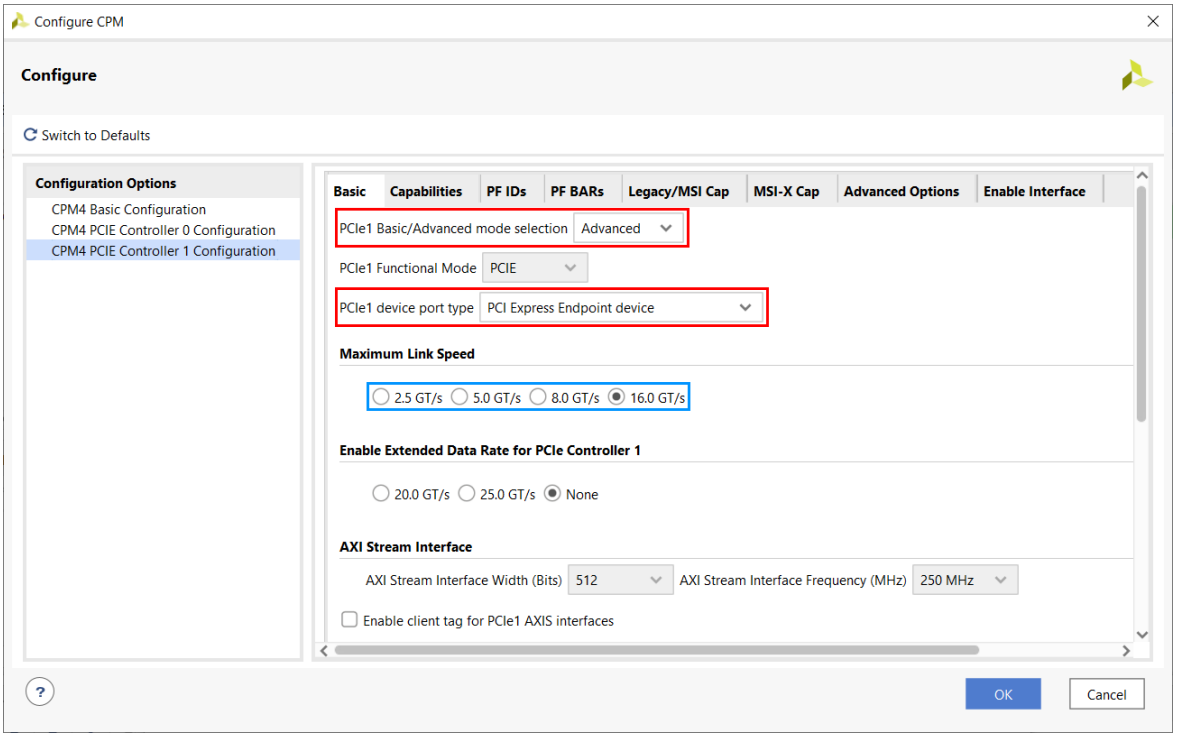

Figure 22: CPM controller 1 Basic tab (upper)

Set ***PCIe1 Basic/Advanced mode selection*** to ***Advanced***. This makes the ***Reference Clock Frequency (MHz)*** and ***Enable Lane Reversal*** options visible so that they can be set correctly (see below).

PCIe1 device port type must be ***PCI Express Endpoint Device***.

NOTE: It is conceivable that the ADM-VA600 could be used as a Root Complex in a custom motherboard, but this use case is outside the scope of this document.

Maximum Link Speed may be one of ***2.5 GT/s***, ***5 GT/S*** or ***8 GT/s*** according to application requirements and the capabilities of the VPX backplane (connector bandwidth etc.). The ADM-VA600's VPX P1 connector is not qualified for operation at ***16 GT/s***, so Alpha Data recommends selecting ***8 GT/s***.

NOTE: At time of writing, Alpha Data has neither tested nor qualified the ADM-VA600 for operating the CPM4 blocks at "extended data rates" of 20 or 25 GT/s.

Figure 23 shows two more options that must be set a particular way:

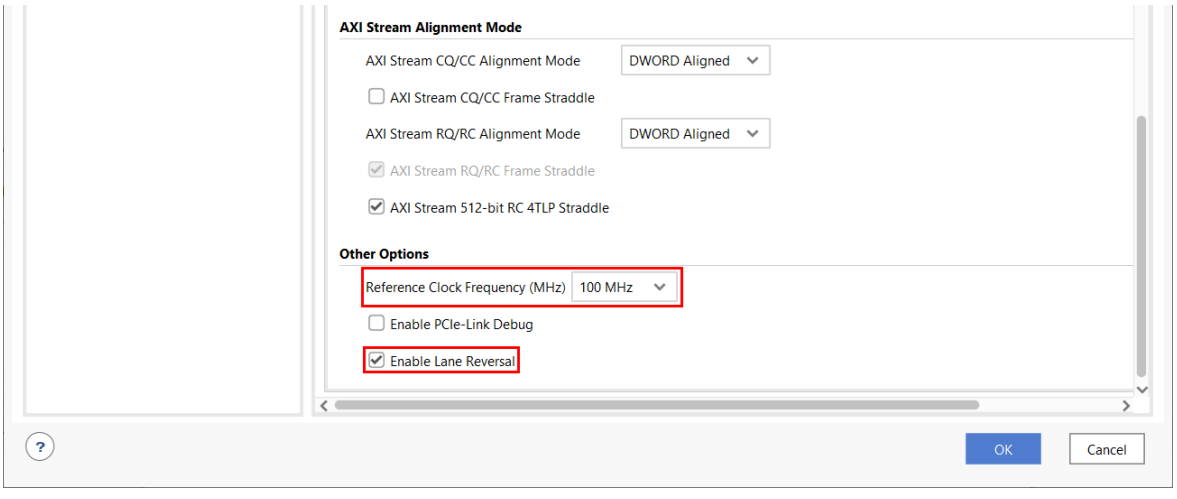

Figure 23: CPM controller 1 Basic tab (lower)

NOTE: The options highlighted in Figure 23 only appear when ***PCIe1 Basic/Advanced mode selection*** is ***Advanced***.

Reference Clock Frequency (MHz) must be set to ***100 MHz*** unless the `PCIE_REFCLK_2` clock has been changed from its factory default frequency of 100 MHz.

Enable Lane Reversal should be checked so that if plugged into a VPX system that reverses the PCIe lanes, the CPM PCIe endpoint will work regardless.

The rest of the options on the ***Basic*** tab are at the discretion of the designer and may be set according to application requirements.

3.3.6 CPM4 PCIE Controller 1 Configuration - Capabilities tab

Only one option on the ***Capabilities*** tab is critical to correct operation of the CPM4 block in the ADM-VA600, namely ***Enable Slot Clock Configuration***. Figure 24 happens to show the ***Capabilities*** tab when the functional mode is QDMA, but this option is present in all functional modes:

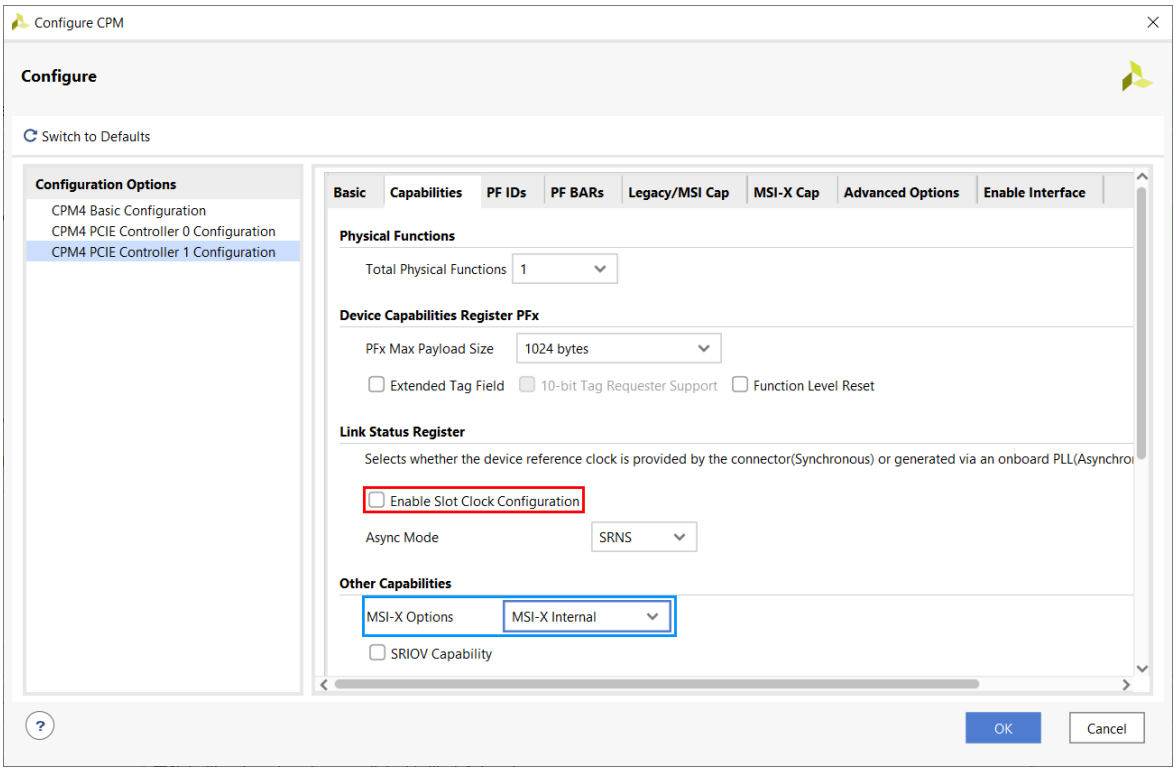

Figure 24: CPM controller 1 Capabilities tab

MSI-X Options may be set to several different values, but Alpha Data recommends setting this option to ***MSI-X Internal***, which causes the ***MSI-X Cap*** tab to become visible (see below). A value of ***MSI-X Internal*** means that the MSI-X logic is implemented within CPM4 block 1 and an implementation need not be provided by the designer.

The ***Enable Slot Clock Configuration*** checkbox must be ***unchecked***, because the CPM PCIe endpoint uses a locally generated reference clock, namely `PCIE_REFCLK_2`, rather than one from the VPX P1 connector.

The rest of the options on the ***Capabilities*** tab are at the discretion of the designer and may be set according to application requirements.

3.3.7 CPM4 PCIE Controller 1 Configuration - MSI-X Cap tab

Alpha Data recommends enabling MSI-X capabilities in the CPM PCIe endpoint, though strictly speaking, it is optional. Figure 25 shows the ***MSI-X Cap*** tab with MSI-X enabled:

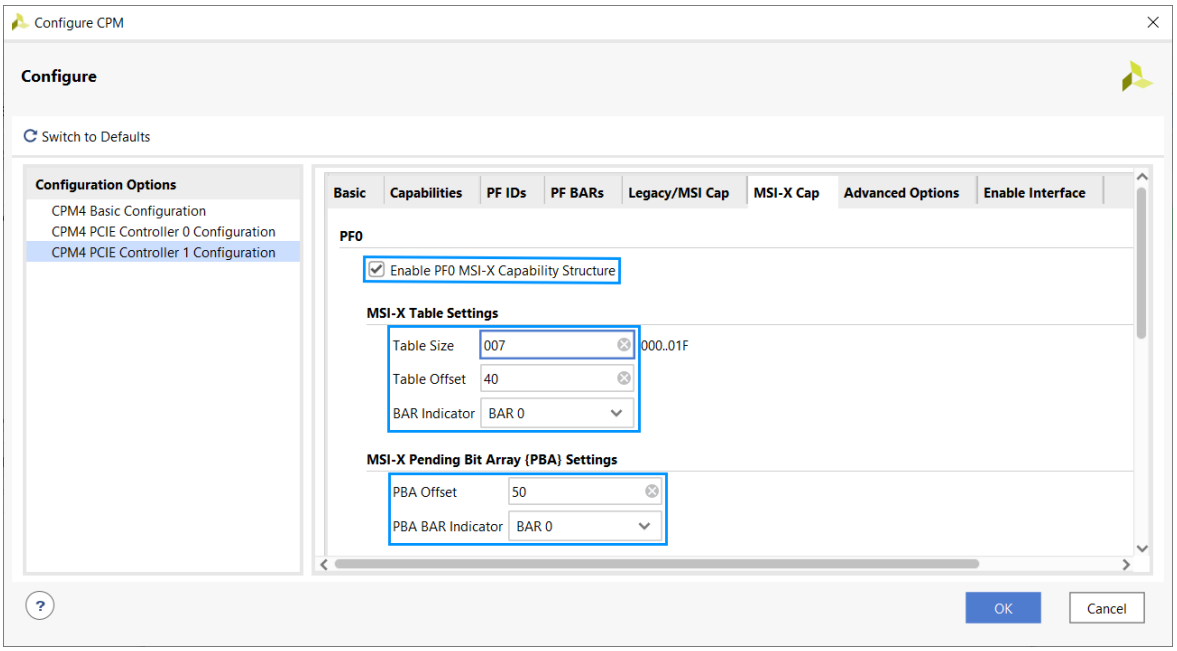

Figure 25: CPM controller 1 MSI-X Cap tab

Use of MSI-X can improve performance in x86-64 platforms because such platforms typically do not allocate more than one MSI (non-X) vector to any given PCIe endpoint. When the host allocates a single MSI vector to the PCIe endpoint, all sources of interrupt in the PCIe endpoint are forced to share a single hardware interrupt vector, resulting in less efficient interrupt servicing. MSI-X, by contrast, appears to be well supported on x86-64 hardware, with hosts typically allocating the requested number of MSI-X vectors to a PCIe endpoint.

Note that it is the designer's responsibility to set the options for ***MSI-X Table Settings*** and ***MSI-X Pending Bit Array {PBA} Settings*** correctly. This is mainly a matter of choosing a BAR (Base Address Register) to host the MSI-X vector table and Pending Bit Array (PBA), and choosing suitable offsets within that BAR. If, in the ***Capabilities*** tab, ***MSI-X Options*** is set to ***MSI-X Internal***, then CPM4 block 1 itself implements the vector table and PBA and the designer need not provide an implementation.

Additionally, it is the designer's responsibility to determine how many MSI-X vectors are required. This number depends upon the number of interrupt sources in the design. If there are more than 32 interrupt sources, sharing of interrupt vectors is inevitable.

3.4 PS/PMC customization, part 2

The final step is to enable the "PCIe reset" peripherals in the PS/PMC module. This could not be done on the first visit to the PS/PMC module because the PCIe reset peripherals are not configurable (that is, their UI elements are disabled) until they are enabled in the CPM module.

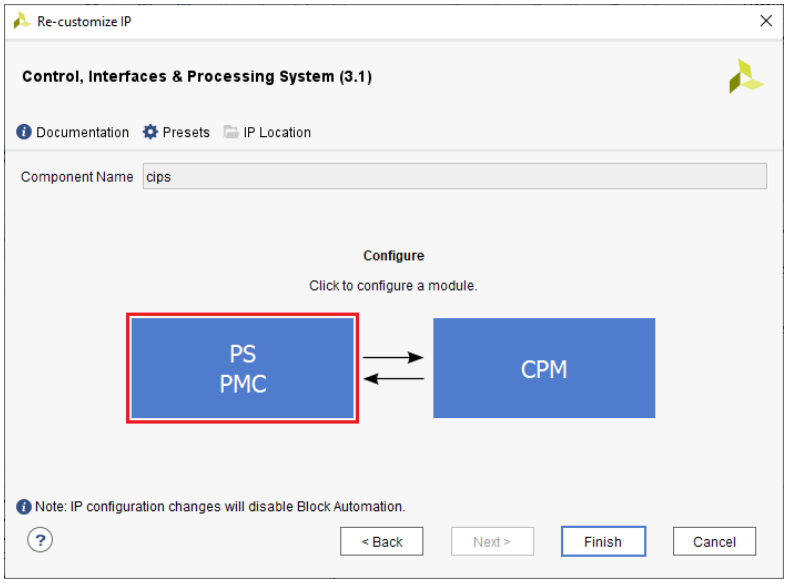

Figure 26: CIPS module selection highlighting PS/PMC

Returning to the module selection page of the wizard, continue PS/PMC configuration by clicking ***PS PMC***.

Peripherals

Select the “Peripherals” tab from the left-hand side of the dialog. The aim is to configure the PS/PMC so that the correct MIO pin(s) act as PCIe Fundamental Reset for the enabled CPM4 block(s). Enable the “PCIe Reset” peripheral:

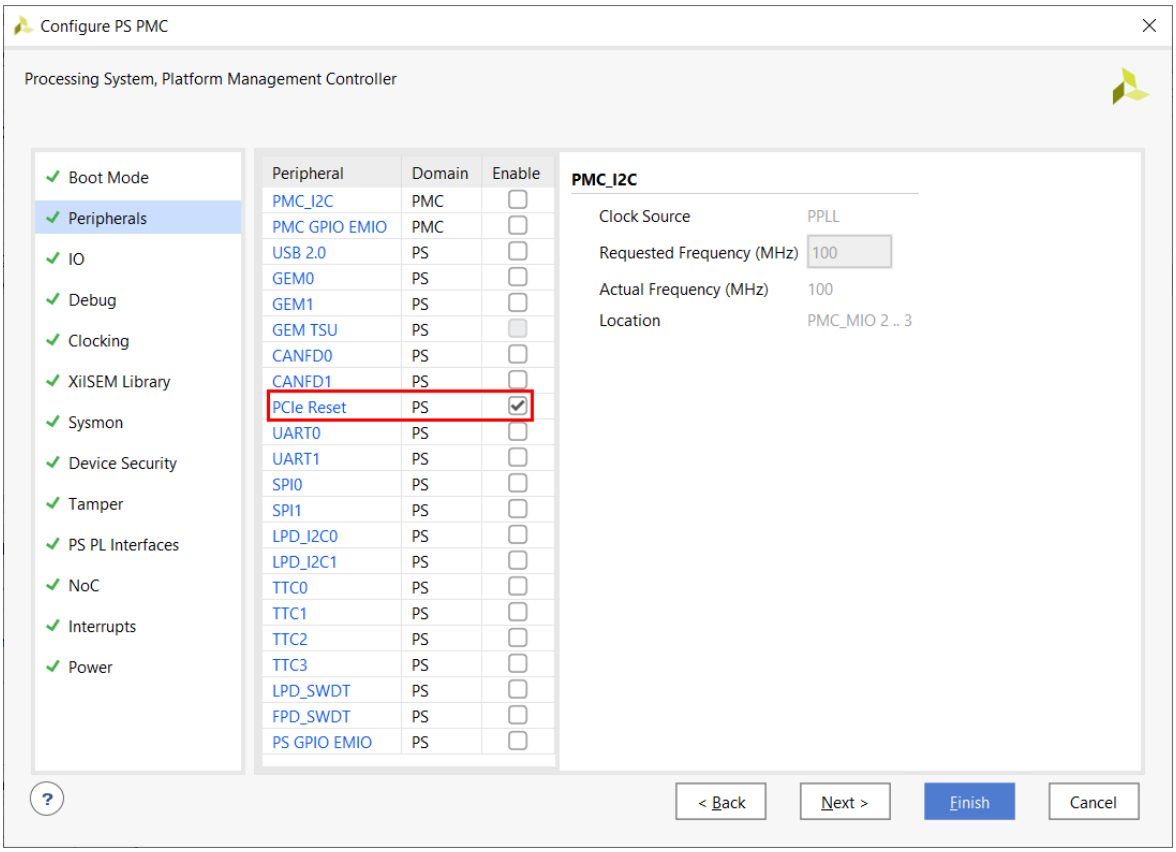

Figure 27: Enabling the “PCIe Reset” peripheral

IO

Now select “IO” from the left-hand side of the dialog. Ensure that MIO18 is chosen for the “PCIe Reset” pin for CPM4 block 0. If you enabled CPM4 block 1, ensure that MIO19 is chosen for the “PCIe Reset” pin for CPM4 block 1. Figure 28 illustrates these settings:

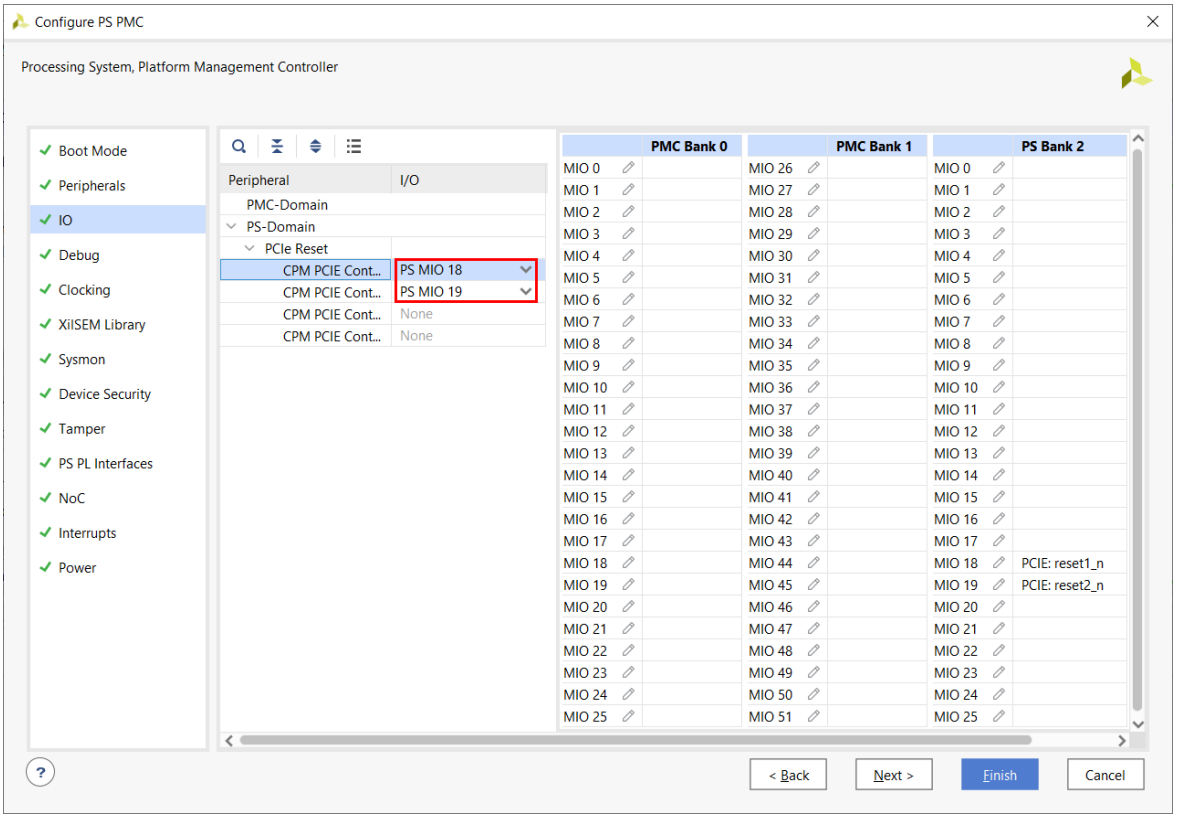

Figure 28: Selecting MIO18 for the “PCIe Reset” peripheral

This concludes the CIPS configuration. Click the ***Finish*** button to exit the PS/PMC module and click the ***Finish*** button to exit the CIPS customization GUI.

4 PACKAGE_PIN and other constraints

In general, the designer need not provide `PACKAGE_PIN` or other constraints for the top-level PCIe-related ports in a CPM PCIe-based design.

4.1 PCIe Fundamental Reset pins

In the ADM-VA600, CPM4 block 0 must use the LPD MIO18 pin (see Figure 2) for PCIe Fundamental Reset. There is no port for this pin in the top level of a CPM PCIe-based design, and it is not possible to provide a `PACKAGE_PIN` constraint for it. However, the designer must configure the CIPS IP so that LPD MIO18 is used for CPM4 block 0; see Section 3.4 for details.

Likewise, if enabled, CPM4 block 1 must use the LPD MIO19 pin (see Figure 2) for PCIe Fundamental Reset. The designer must configure the CIPS IP so that LPD MIO19 is used for CPM4 block 1; see Section 3.4 for details.

4.2 PCIe reference clock pins

CPM4 block 0 automatically uses `PCIE_REFCLK_0` (see Figure 2) and the designer need not provide `PACKAGE_PIN` constraints for it.

CPM4 block 1 (if enabled) automatically uses `PCIE_REFCLK_2` (see Figure 2) and the designer need not provide `PACKAGE_PIN` constraints for it.

A user-defined `create_clock` constraint should not be present for the PCIe reference clock(s) that are used in the design because the CIPS IP automatically defines them according to the ***Reference Clock Frequency (MHz)*** option (see Figure 19 and Figure 23).

4.3 PCIe lane pins

CPM4 block 0 and CPM4 block 1 (if enabled) use the GTY channels of Quads 103 to 106 (see Figure 2) as shown in Figure 3 and the designer need not provide `PACKAGE_PIN` constraints for them.

5 Cabled PCIe connection using ADM-VA600-RTM

When using the ADM-VA600 together with the ADM-VA600-RTM and without a CPU card in the VPX chassis, it is possible to create a cabled PCIe connection to a standard PC that takes the place of a VPX CPU card. The additional equipment required for this arrangement, besides the ADM-VA600, is:

- ADM-VA600 Rear Transition Module (RTM) – ADM-VA600-RTM board rev. 2 or later, Alpha Data
- Single-slot OpenVPX backplane – OVPX BKP6 1Slot, Hartmann Electronics
- PCIe to SFF8644 adapter – AD-PCIE-EXTEND-G3X8, board rev. 2 or later, Alpha Data
- Two standard SFF8644 to SFF8644 cables, maximum length 1m
- Standard AMD/Intel-based PC (see requirements below) which acts as the host machine

Figure 29 illustrates the assembled system:

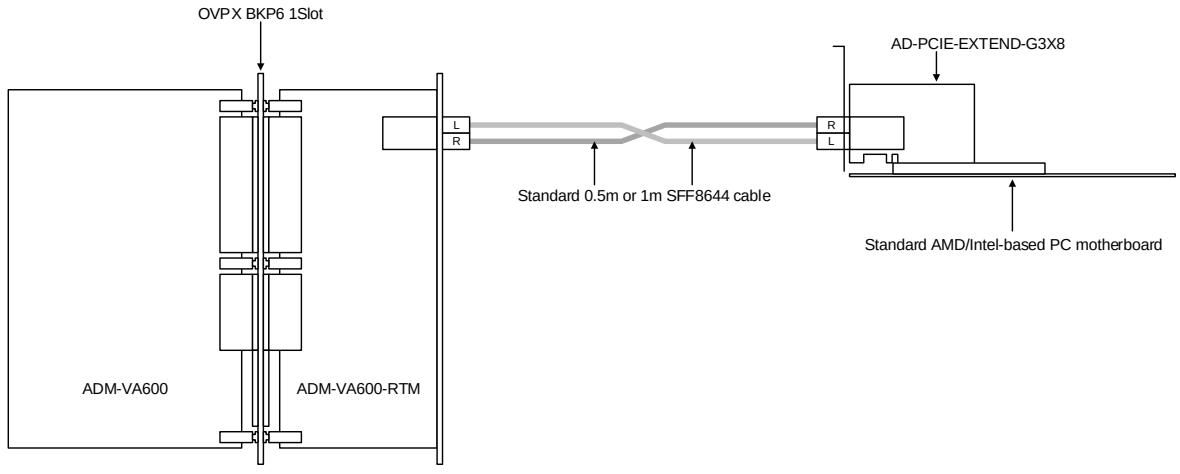

Figure 29: Assembled cabled PCIe system

The AMD/Intel-based PC must use a PCIe reference clock that is within the limits mandated by the PCI Express Specification (99.97 to 100.03 MHz) and Spread Spectrum must be disabled for the PCIe reference clock.

A shortcoming of this arrangement is that there is no physical wire to propagate PCIe Fundamental Reset (PERST#) to any PCIe endpoint that has been instantiated within the ADM-VA600's ASoC, which means it is

not a fully PCIe compliant solution. However, the host can still transmit PCIe Hot Reset packets to the PCIe endpoint, via the cables, and this is generally sufficient to reset the PCIe endpoint.

To assemble the system, follow these steps:

1. Assemble the ADM-VA600, ADM-VA600-RTM and OpenVPX backplane together in an OpenVPX-compliant chassis with a power supply.
2. Plug the AD-PCIE-EXTEND-G3X8 into the Standard AMD/Intel-based PC.
3. Connect the AD-PCIE-EXTEND-G3X8 to the ADM-VA600-RTM using the two SFF8644 to SFF8644 cables. Ensure that:
    - One cable uses the “left” SFF8644 socket on both the ADM-VA600 and the AD-PCIE-EXTEND-G3X8
    - The other cable uses the “right” SFF8644 socket on both the ADM-VA600 and the AD-PCIE-EXTEND-G3X8

Finally, to ensure that the CPM PCIe endpoints in the Adaptive SoC is are not held in reset in this hardware arrangement, please ensure that either:

- SW1-5 on the ADM-VA600 is OFF

  This isolates the VPX SYSRESET# signal from the PCIe Fundamental Reset signals for the two CPM4 blocks. The PCIe resets are pulled up on the circuit board, ensuring that the CPM4 blocks are not held in reset.
- SW1-5 on the ADM-VA600 is ON and SW2-7 & SW2-8 on the ADM-VA600-RTM are both OFF

  This connects the VPX SYSRESET# signal to the PCIe Fundamental Reset signals for the two CPM4 blocks but ensures that it is pulled up. This ensures that the CPM4 blocks are not held in reset.

6 Related documents

1. PG293 – LogiCORE IP Product Guide “AXI Protocol Firewall IP”, AMD Inc. (formerly Xilinx Inc.)
2. PG346 – Product Guide “Versal Adaptive SoC CPM Mode for PCI Express”, AMD Inc. (formerly Xilinx Inc.)
3. PG347 – Product Guide “Versal Adaptive SoC CPM DMA and Bridge Mode for PCI Express”, AMD Inc. (formerly Xilinx Inc.)
4. PG352 – LogiCORE IP Product Guide “Control Interfaces and Processing System”, AMD Inc. (formerly Xilinx Inc.)
5. AD-UG-1456 – ADM-VA600 User Manual, Alpha Data Parallel Systems Ltd.
6. AD-UG-1457 – ADM-VA600-RTM User Manual, Alpha Data Parallel Systems Ltd.
7. AD-AN-0142 – Application Note “ADM-VA600 Board File”, Alpha Data Parallel Systems Ltd.
8. AD-UG-0182 – User Guide “ADM-VA600 CPM PCIe Demonstration FPGA Design”, Alpha Data Parallel Systems Ltd.

Document version history

Document version	Notes
1.0	Initial version.

	Head Office	US Office
Address:	Suite L4A, 160 Dundee Street Edinburgh, EH11 1DQ, UK	Suite 250, 10822 West Toller Drive Deer Creek Technology Center, Littleton, CO 80127
Telephone:	+44 131 558 2600	(303) 954 8768
Fax:	+44 131 558 2700	(866) 820 9956 - toll free
email:	sales@alpha-data.com	sales@alpha-data.com
website:	www.alpha-data.com	www.alpha-data.com