ADM-PA120 Board File

Alpha Data Parallel Systems Ltd.

Document number: **AD-AN-0167**

Document revision: **v1.0 / February 13, 2025**

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.

A board file is a high-level description of the major features of the ADM-PA120, which can be used with Vivado’s IP Integrator tool. The purpose of the board file is to streamline the process of creating a Block Diagram (BD) design, and:

- Provides ***presets*** for certain IPs that are frequently used in designs for the ADM-PA120.
- Defines key ***board interfaces*** such as differential clocks, LPDDR4 SDRAM interfaces etc.
- Defines the pinout and certain physical constraints for the Adaptive SoC in the ADM-PA120, so that the user need not provide `DIFF_TERM`, `IOSTANDARD`, `PACKAGE_PIN` etc. constraints for the board interfaces defined by the board file.

1.1 Board files and order codes

As the ADM-PA120 may be manufactured with more than one Adaptive SoC part, there are multiple board files for the ADM-PA120. The available board files, organized by order code prefix, are given in the following table:

Order code prefix	Board file VLNV
ADM-PA120	`alpha-data.com:admpa120_vp1202_2ms:part0:*`
ADM-PA120/VP1502	`alpha-data.com:admpa120_vp1502_2ms:part0:*`

Table 1: Available ADM-PA120 files by order code prefix

A particular board file might cover several order codes because certain ordering options are considered “cosmetic” as far as AMD tools such as Vivado, Vitis and Petalinux are concerned. An example of such a cosmetic option might be a conformal coating option.

“Board file VLNV” is the string that Vivado uses to identify a particular board file. A board file VLNV may be passed to various Vivado Tcl commands; for example `get_board_parts`. The board VLNVs map to particular Adaptive SoCs as in the following table:

Board file VLNV	Adaptive SoC targeted by board file
`alpha-data.com:admpa120_vp1202_2ms:part0:*`	XCVP1202-2MSEVSVA2785
`alpha-data.com:admpa120_vp1502_2ms:part0:*`	XCVP1502-2MSEVSVA2785

Table 2: Adaptive SoCs targeted by board file VLNV

2 Obtaining the board files

The recommended way to obtain the ADM-PA120 board files is to download them from within Vivado as described in AMD UG994.

To download a board file, first open the Vivado Store dialog by selecting ***Tools*** →***Vivado Store...*** from the Vivado main menu. The following graphic shows the steps for downloading a board file within the Vivado Store dialog.

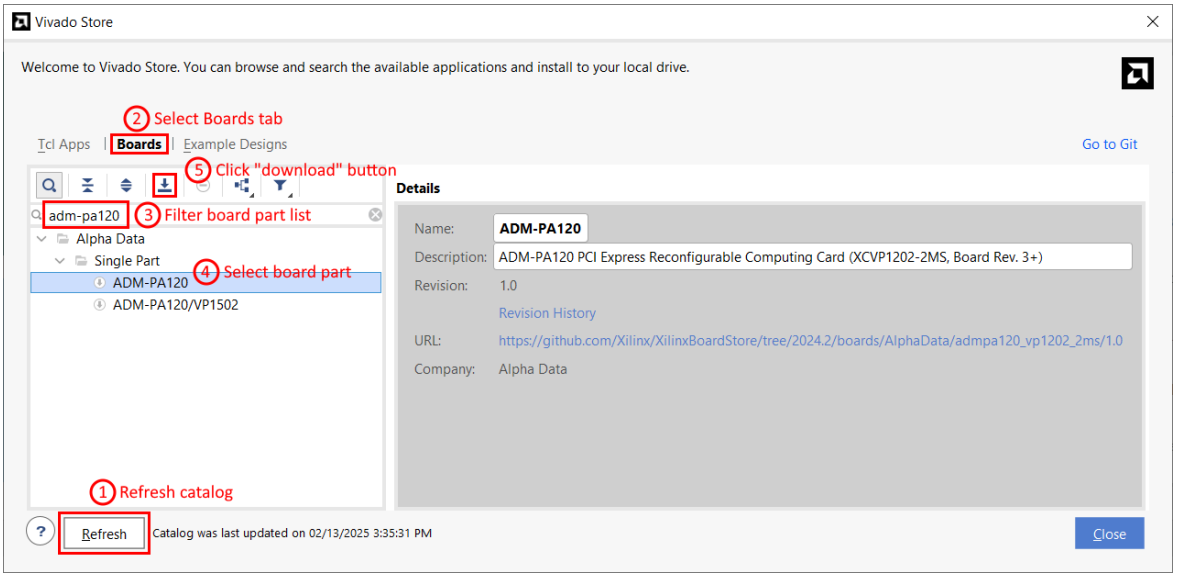


Figure 1: Steps for downloading a board file

If successful, a green tick appears next to the board as shown in Figure 2 below:

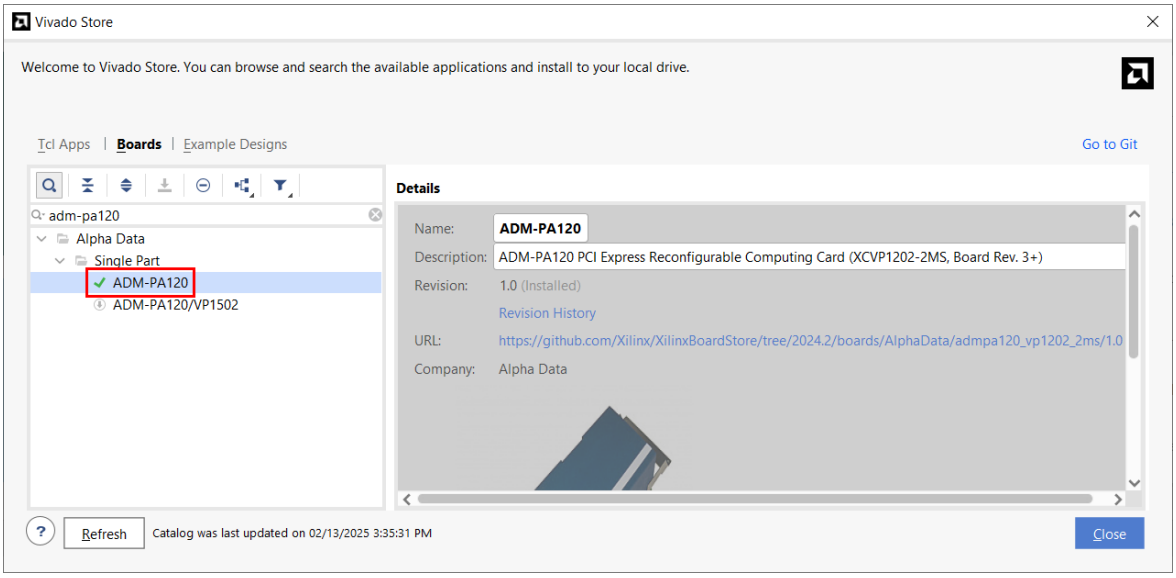


Figure 2: After successfully downloading a board file

The Vivado Store dialog can now be closed by clicking on the ***Close*** button. Any downloaded board parts will be available when creating new Vivado projects.

3 Board file features

This section provides a brief overview of the IP presets and board interfaces defined by an ADM-PA120 board file.

3.1 CIPS Block Automation

The board file defines a preset for Control Interfaces & Processing System (CIPS) IP that is used by Vivado's Block Automation feature. To use this feature, create a new CIPS IP instance in a Vivado Block Diagram (BD) design. Then right-click on the CIPS IP instance and select ***Run Block Automation...*** as in Figure 3:

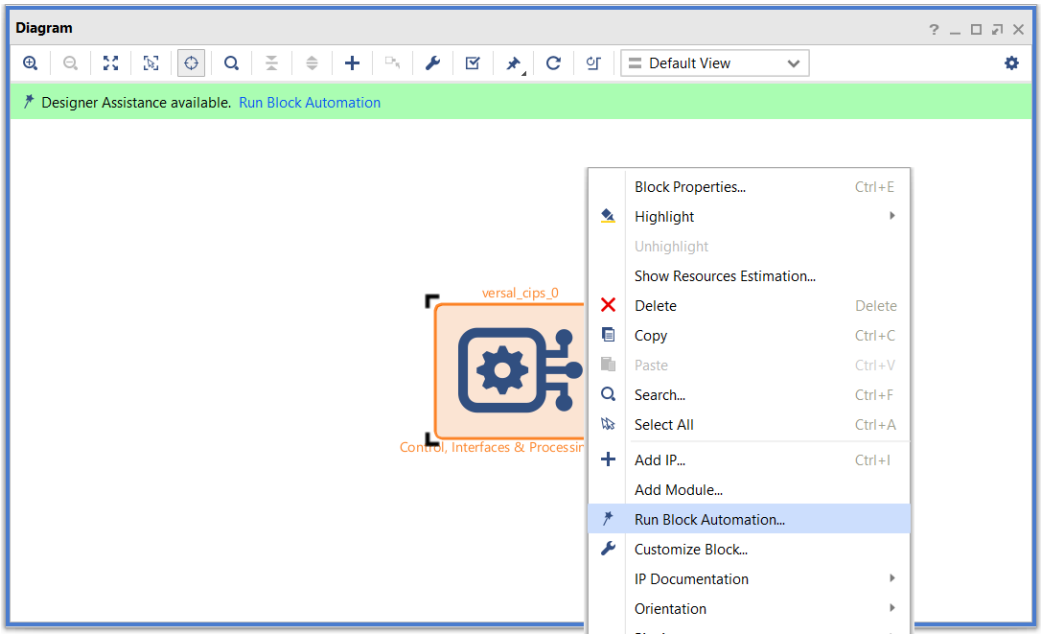


Figure 3: Running Block Automation on the CIPS IP instance

The Block Automation dialog for the CIPS IP has various options which can be set as per applicament requirements, for the most part. However, in order to configure the PS/PMC module correctly for the ADM-PA120, set ***Apply Board Preset*** to ***Yes*** as in Figure 4:

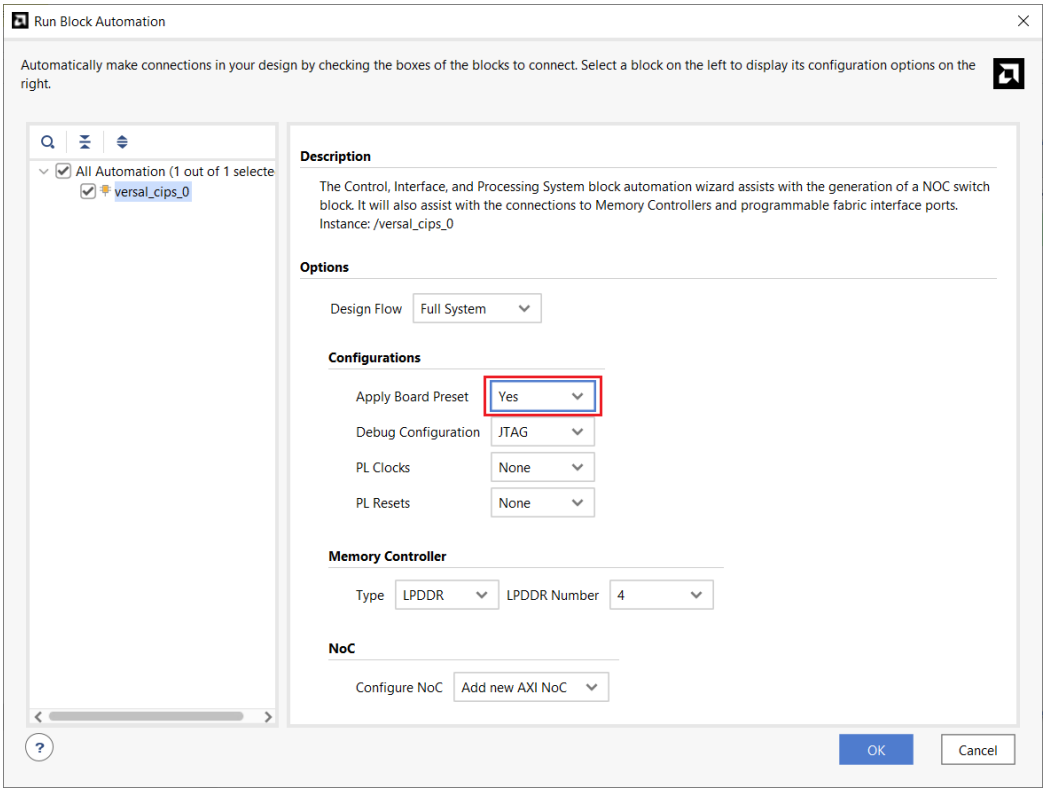


Figure 4: Typical CIPS Block Automation settings

NOTE: If the ***Apply Board Preset*** option is ***Yes***, Block Automation applies a particular CIPS IP preset named `ps_pmc_fixed_io`. See Section 3.2 for information about CIPS IP presets.

The result of the above Block Automation settings is shown in Figure 5:

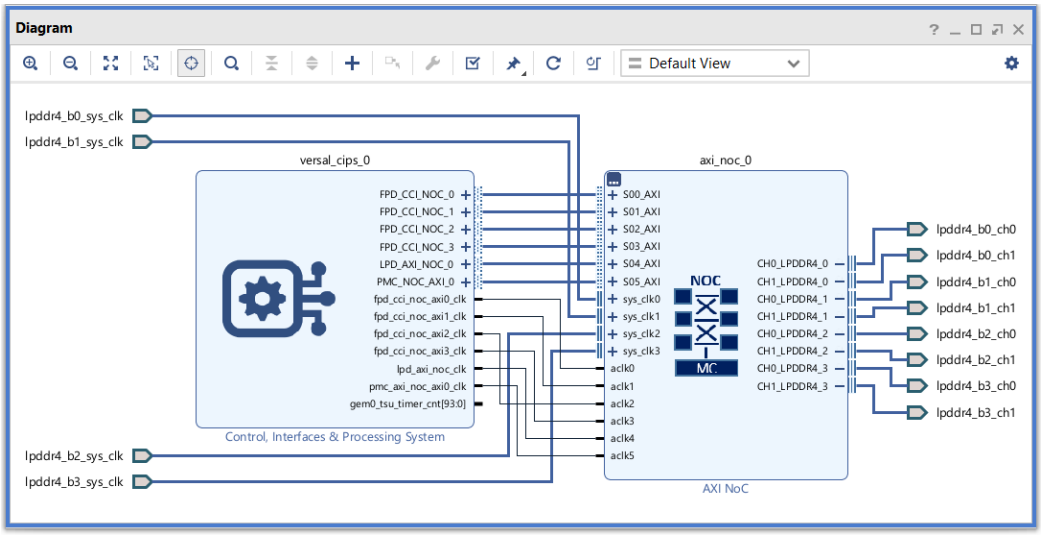


Figure 5: After running Block Automation

The above is a good starting point for a design that uses the Adaptive SoC's APU and/or RPU embedded processors, since it includes LPDDR4 SDRAM controllers.

3.2 CIPS IP presets

The board file defines several presets for the CIPS IP which offer a convenient way to configure the CIPS IP without using Block Automation. They are summarized in Table 3 below:

Preset name	PS I/O interfaces	Boot modes	IPI & TTCs
`ps_pmc_fixed_io`	Configured	JTAG, QSPI, SD card (2.0)	Not configured
`ps_pmc_minimal`	Not configured	JTAG, QSPI, SD card (2.0)	Not configured
`ps_pmc_fixed_io_linux`	Configured	JTAG, QSPI, SD card (2.0)	Configured
`ps_pmc_fixed_io_linux_qspis`	Configured	JTAG, QSPI, SD card (2.0)	Configured

Table 3: Available CIPS IP presets

In all presets, the PS reference clock is set to 33.333 MHz, the PS I/O bank voltages are correctly set and UART0 is enabled (for debugging boot problems).

NOTE: The `ps_pmc_fixed_io` preset is used by Block Automation if the ***Apply Board Preset*** option is ***Yes*** (see Figure 4).

The suffices that make up the names of the presets have the following meanings:

- Presets that include a suffix `_fixed_io` configure the PS reference clock, UART0, UART1, boot modes, and those PS interfaces that are fixed by the board design.ADM-PA120.
- Presets that include a suffix `_minimal` only configure the PS reference clock, UART0 and boot modes, and omit configuring the rest of the PS I/O interfaces supported in the ADM-PA120.
- Presets that include a suffix `_linux` additionally configure (i) Inter-Processor Interrupts (IPI) so that the APU and RPU can communicate with the PMC and (ii) all four Triple Timer/Counter (TTC) peripherals. This preset is suitable for running a Standalone ("bare metal") application or Petalinux / Yocto Linux in the APU.
- Presets that include a suffix `_qspis` configure the PS QSPI Flash periperhal in "x4 Single" mode instead of "x4 Dual Parallel" mode; see box immediately below.

u-boot 2023.2 / 2024.1 Dual Parallel QSPI Flash issue

The `ps_pmc_fixed_io_linux_qspis` preset is provided in order to work around an issue where u-boot generated by Petalinux 2023.2 / 2024.1 (at least) does not successfully retrieve Petalinux images from QSPI Flash in “x4 Dual Parallel” mode. This preset works around the issue by configuring the QSPI Flash in “x4 Single” mode, which allows Petalinux to boot successfully at the cost of reduced Flash read/write performance and half of the expected Flash memory capacity. Refer to Section 5.3 for further details about this issue.

A CIPS IP preset is applied by customizing the CIPS IP instance and selecting it for ***Board Interface***, as shown in Figure 6:

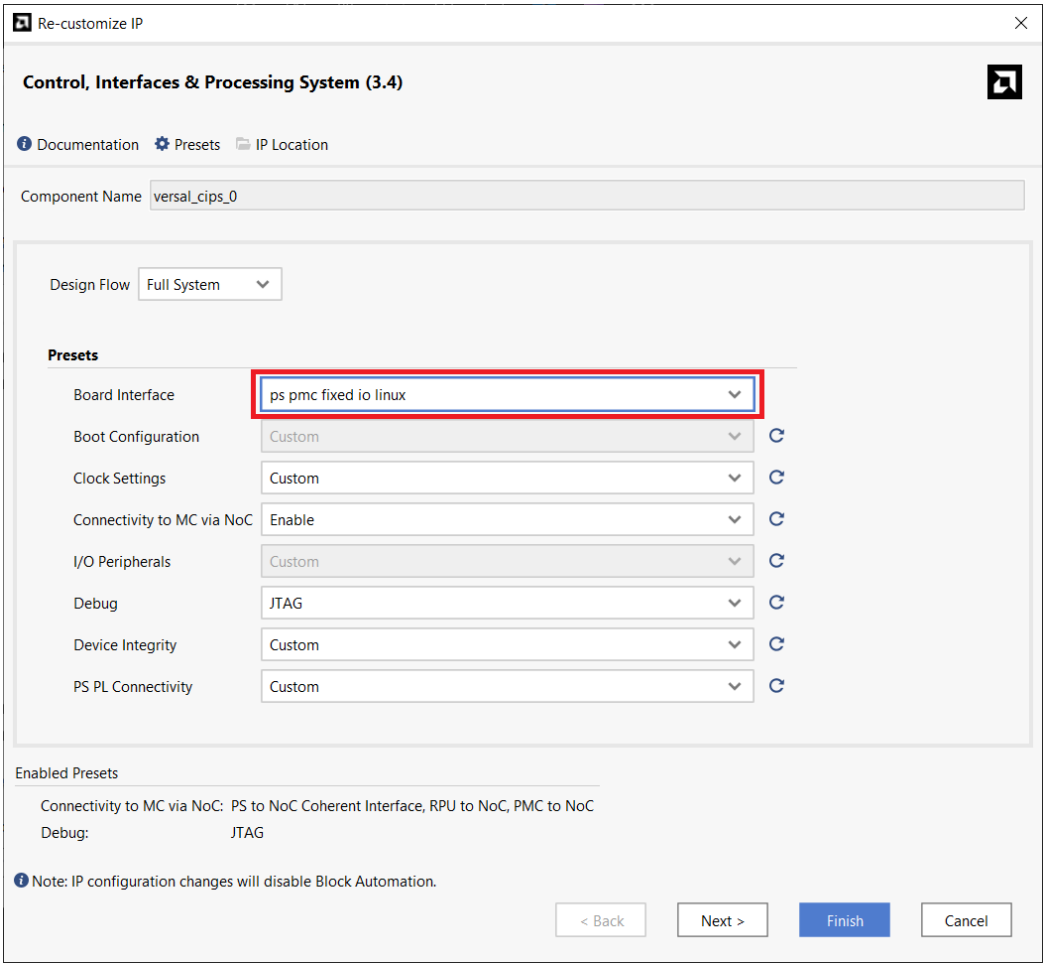


Figure 6: Applying a CIPS IP preset

NOTE: It is possible to change the preset after running Block Automation. Thus, a convenient way to generate a Block Diagram design capable of running Petalinux in the APU is to run Block Automation (see Section 3.1) and then change ***Board Interface*** to the `ps_pmc_fixed_io_linux` presets.

3.3 LPDDR4 SDRAM interfaces

In Versal architecture, memory controllers are built into the silicon and are considered part of the Network on Chip (NoC). The ADM-PA120 has four banks of LPDDR4 SDRAM and the board file accordingly provides pinouts and IP presets for them. The LPDDR4 SDRAM-related board interfaces defined in the ADM-PA120 board file are summarized in Table 4:

Interface name	Description
`lpddr4_b0_ch0`	Signals compromising the physical interface to LPDDR4 SDRAM bank 0 channel 0.
`lpddr4_b0_ch1`	Signals compromising the physical interface to LPDDR4 SDRAM bank 0 channel 1.
`lpddr4_b1_ch0`	Signals compromising the physical interface to LPDDR4 SDRAM bank 1 channel 0.
`lpddr4_b1_ch1`	Signals compromising the physical interface to LPDDR4 SDRAM bank 1 channel 1.
`lpddr4_b2_ch0`	Signals compromising the physical interface to LPDDR4 SDRAM bank 2 channel 0.
`lpddr4_b2_ch1`	Signals compromising the physical interface to LPDDR4 SDRAM bank 2 channel 1.
`lpddr4_b3_ch0`	Signals compromising the physical interface to LPDDR4 SDRAM bank 3 channel 0.
`lpddr4_b3_ch1`	Signals compromising the physical interface to LPDDR4 SDRAM bank 3 channel 1.
`lpddr4_b0_sys_clk`	Differential reference clock for memory controller for LPDDR4 SDRAM bank 0.
`lpddr4_b1_sys_clk`	Differential reference clock for memory controller for LPDDR4 SDRAM bank 1.
`lpddr4_b2_sys_clk`	Differential reference clock for memory controller for LPDDR4 SDRAM bank 2.
`lpddr4_b3_sys_clk`	Differential reference clock for memory controller for LPDDR4 SDRAM bank 3.

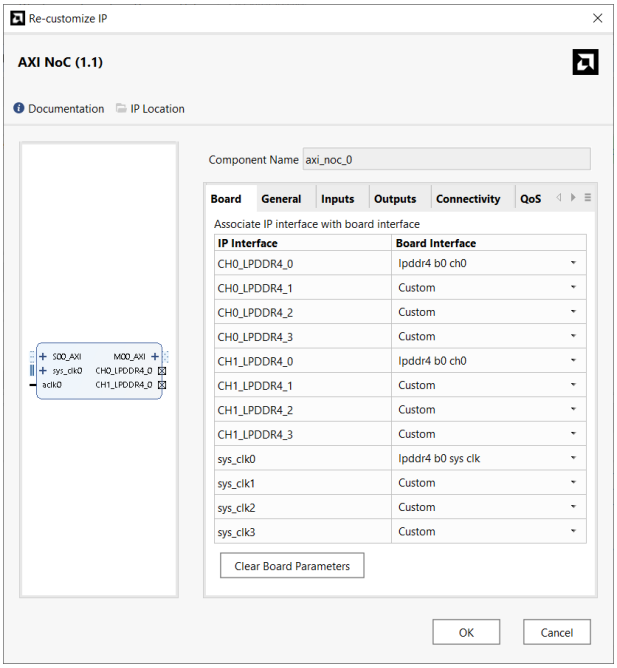
Table 4: Available LPDDR4 SDRAM-related interfaces

LPDDR4 SDRAM banks 0, 1 & 2 operate at LPDDR4-3900 data rate (1950 GT/s) whereas LPDDR4 SDRAM bank 3 operates at LPDDR4-3200 (1600 GT/s). The slower data rate for bank 3 is due to limitations of the VP1202 & VP1502 ASoCs in the -2MS speed grade.

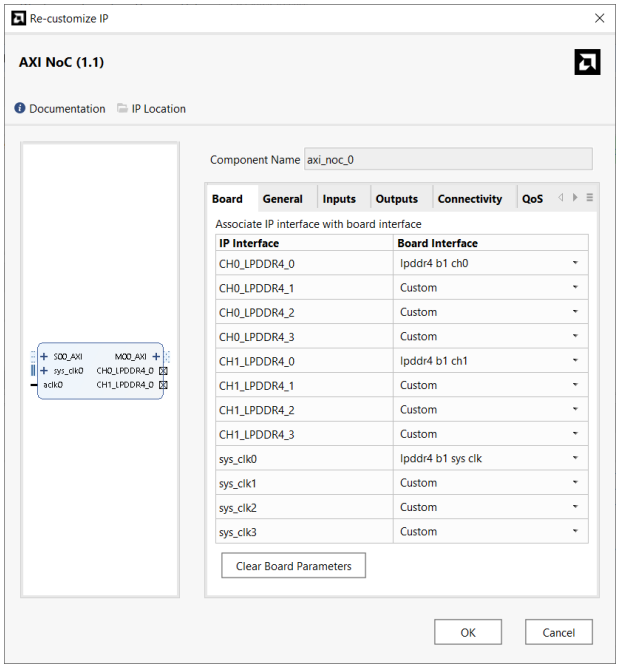
The AXI NoC IP is used to implement LPDDR4 SDRAM controller(s), and there are several operations for interleaving LPDDR4 banks:

- No interleaving - the AXI NoC IP instance controls a single LPDDR4 bank (0, 1, 2 or 3).
- Interleaving of two banks - the AXI NoC IP instance controls either (i) LPDDR4 banks 0 & 1 or (ii) LPDDR4 banks 2 & 3.
- Interleaving of four banks - the AXI NoC IP instance controls LPDDR4 banks 0, 1, 2 & 3.

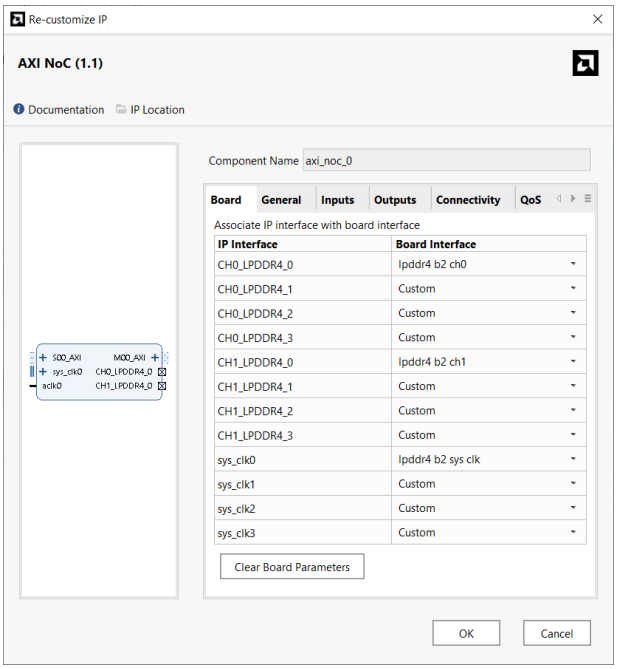
Single-bank cases are illustrated in Figure 7a, Figure 7b, Figure 7c & Figure 7d, for LPDDR4 SDRAM banks 0, 1, 2 & 3 respectively:



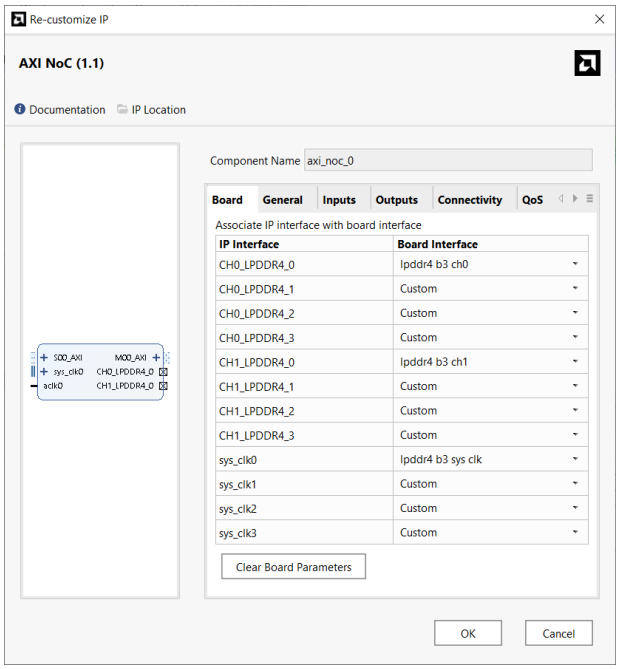
(a) AXI NoC IP configured for bank 0



(b) AXI NoC IP configured for bank 1

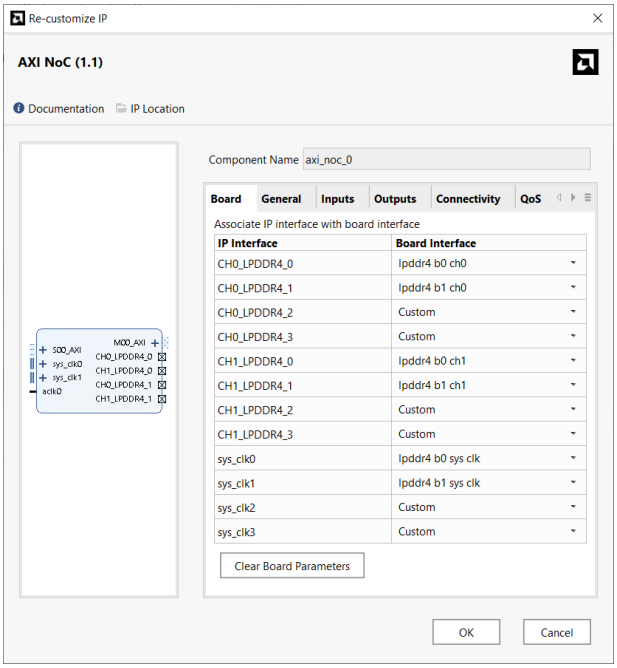


(c) AXI NoC IP configured for bank 2

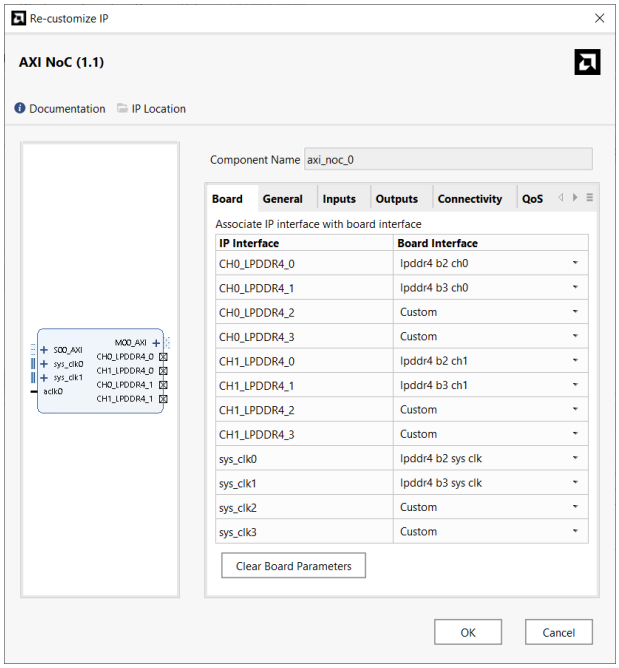


(d) AXI NoC IP configured for bank 3

Dual-bank interleave cases are illustrated in Figure 8a & Figure 8b:



(a) AXI NoC IP configured for banks 0 & 1



(b) AXI NoC IP configured for banks 2 & 3

Finally, Figure 9 illustrates the quad-bank interleave case:

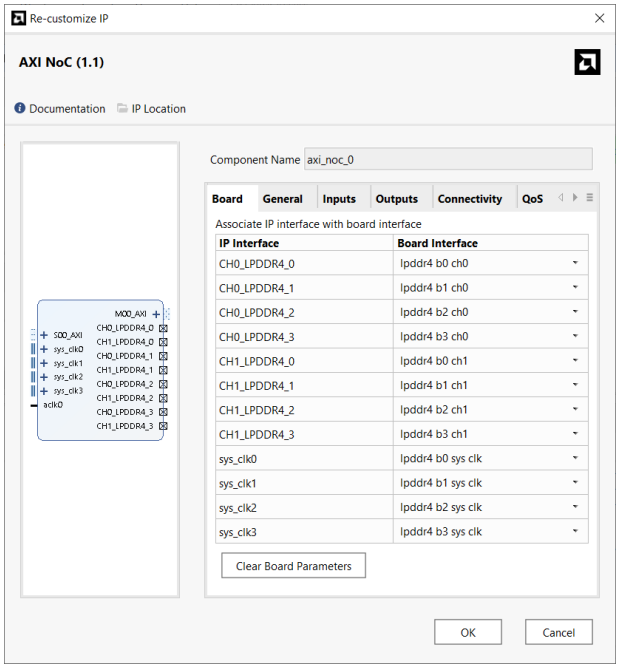


Figure 9: AXI NoC IP configured for banks 0, 1, 2 & 3

Interleaving LPDDR4 SDRAM bank 3

NOTE: Because LPDDR4 SDRAM bank 3 operates at LPDDR4-3200 data rate, interleaving it with bank 2 or with banks 0, 1 & 2 reduces the data rate of the other bank(s) to LPDDR4-3200.

Setting the board interfaces as described above is sufficient to correctly configure all of the LPDDR4 SDRAM-related properties of the AXI NoC IP such as memory timings and geometry. The pinout(s) for the chosen

LPDDR4 SDRAM bank(s) are defined by the board file, and therefore the designer need not provide constraints files that define LPDDR4 SDRAM bank pinouts.

However, some LPDDR4-related settings can still be tweaked by the designer. For example:

- DBI usage for reads
- DBI usage for writes
- Individual memory timings (e.g. t_{FAW})

3.4 Differential clock interfaces

The board file defines various differential clock board interfaces:

- GT Quad reference clocks for all GT Quads used in the ADM-PA120.
- PL (fabric) clocks, connected to clock-capable pins.

Please see the ***Board*** tab in IP Integrator to see the complete list of differential clocks defined by the board file. These clocks can be used in a Block Diagram design by right-clicking on an item and selecting ***Connect Board Component***:

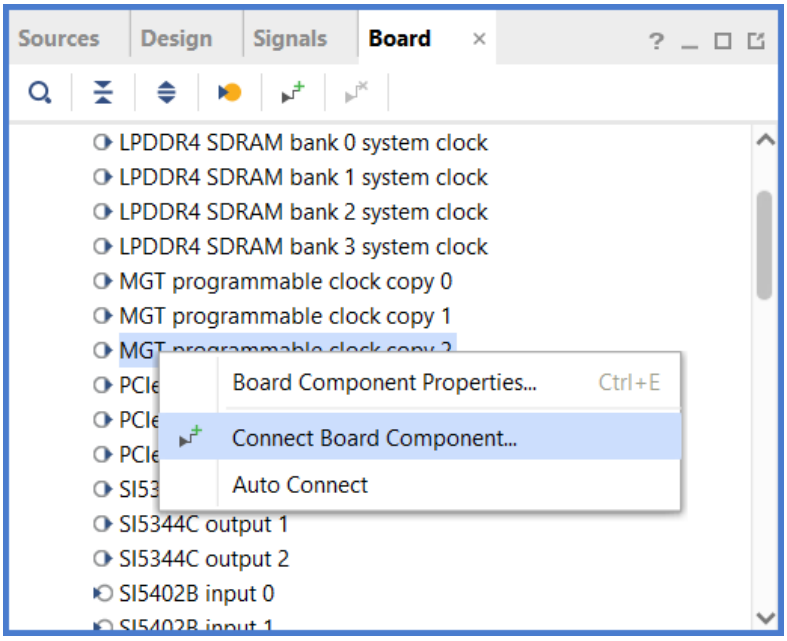


Figure 10: Connecting a board file-defined clock

On doing so, a list of compatible IPs and IP interfaces is offered to the designer:

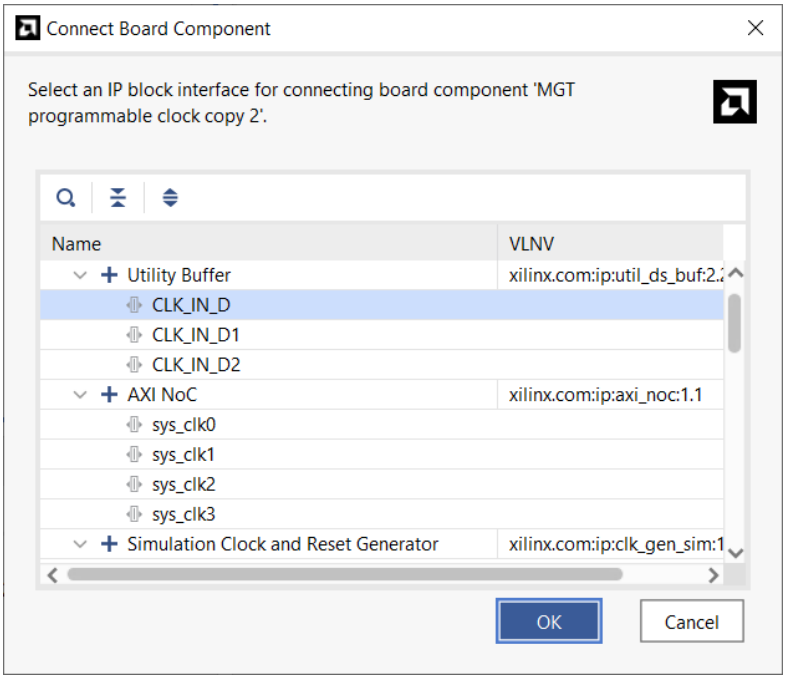


Figure 11: Choosing an IP and IP interface to use with a board-file defined clock

Vivado then creates an instance of the chosen IP, creates an interface port appropriate for the chosen clock and connects the two together. Figure 12 shows the outcome of this process for a board file-defined `mgt_progclk_2`:

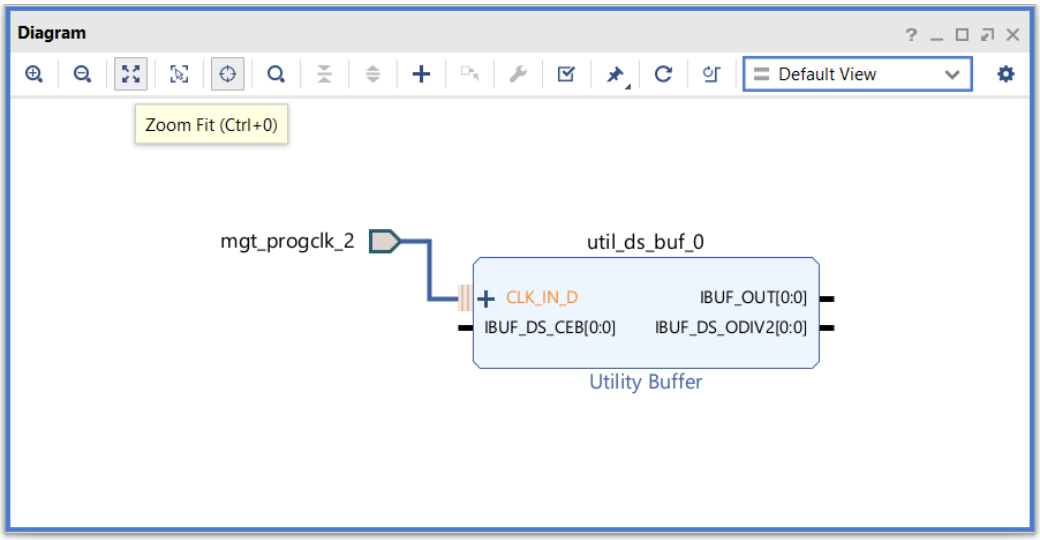


Figure 12: Board file-defined clock connected to chosen IP

Note that when connecting a board-file defined clock to an instance of the Utility Buffer IP, the correct IP interface ***must*** be selected based on the type of buffer that the board file requires for a particular clock. For example, for `mgt_progclk_0`, the instantiated Utility Buffer IP instance is configured to have a `CONFIG.C_BUF_TYPE` parameter of `IBUFDS_GTME5`, so that the IP interface selected must be `CLK_IN_D1`. Table 5 below details the correct Utility Buffer IP interface to use for a given board-file defined clock.

Board clock name	CONFIG.C_BUF_TYPE	Utility Buffer IP Interface
itu_10mhz	IBUFDS	CLK_IN_D
lpddr4_b0_sys_clk	IBUFDS	CLK_IN_D
lpddr4_b1_sys_clk	IBUFDS	CLK_IN_D
lpddr4_b2_sys_clk	IBUFDS	CLK_IN_D
lpddr4_b3_sys_clk	IBUFDS	CLK_IN_D
mgt_progclk_0	IBUFDS_GTME5	CLK_IN_D1
mgt_progclk_1	IBUFDS_GTME5	CLK_IN_D1
mgt_progclk_2	IBUFDSGTE	CLK_IN_D
pcie_lcl_refclk	IBUFDSGTE	CLK_IN_D
pcie_refclk_0	IBUFDSGTE	CLK_IN_D
pcie_refclk_1	IBUFDSGTE	CLK_IN_D
si5344_out0	IBUFDS_GTME5	CLK_IN_D1
si5344_out1	IBUFDS_GTME5	CLK_IN_D1
si5344_out2	IBUFDSGTE	CLK_IN_D
si5402_out1	IBUFDSGTE	CLK_IN_D
si5402_out2	IBUFDSGTE	CLK_IN_D
si5402_out3	IBUFDS_GTME5	CLK_IN_D1
si5402_out4	IBUFDS_GTME5	CLK_IN_D1
si5402_out5	IBUFDS	CLK_IN_D
si5402_out6	IBUFDS_GTME5	CLK_IN_D1
si5402_out7	IBUFDS_GTME5	CLK_IN_D1

Table 5: Utility Buffer IP interface requirement by board clock

Issue affecting Utility Buffer IP configured as IBUFDS_GTME5 buffer type

When the Utility Buffer IP is configured as IBUFDS_GTME5 buffer type as a result of using it with a differential clock board interface, incorrect board constraints are generated when the IP is synthesized. See Section 5.2 for more details about this issue.

3.5 Single-ended clock interfaces

The board file defines one single-ended clock board interface: fabric_clk.

Few IPs are provided by Vivado that can make use of such a board interface, most notably Clock Wizard IP, but the interface is nevertheless provided for completeness. Connecting a single-ended clock board interface can be done in a similar manner to that of a differential clock, as described in Section 3.4.

Issue affecting single-ended board clock interfaces in Vivado 2024.x

Single-ended clock board interfaces cannot be connected to any IPs in Vivado 2024.x IP Integrator. See Section 5.1 for more details of about this issue.

3.6 General purpose I/O (GPIO) interfaces

The board file defines the following GPIO-style interfaces:

Interface name	Description
`board_monitoring`	Board monitoring signals (1 bit input only) [0] `fan_fail_l`
`pmod_io`	PMOD connector (8 bits input / output) [7:0] `pmod_io[7:0]`
`qsfp0_mgmt_out`	QSFP-DD 0 management signals (2 bits output only) [0] `qsfp0_reset_l` [1] `qsfp0_lpmode`
`qsfp1_mgmt_out`	QSFP-DD 1 management signals (2 bits output only) [0] `qsfp1_reset_l` [1] `qsfp1_lpmode`
`qsfp1_mgmt_out`	QSFP-DD 2 management signals (2 bits output only) [0] `qsfp2_reset_l` [1] `qsfp2_lpmode`
`si5344_mgmt_in`	SI5344 status signals (3 bits input only) [0] `si5344_intr_n` [1] `si5344_lol_xaxb_n` [2] `si5344_lol_n`
`si5344_mgmt_out`	SI5344 reset signal (1 bit input / output) [0] `si5344_rst_n`
`si5402_gpio`	SI5402 general purpose I/O signals (3 bits input / output) [2:0] `si5402_gpio[2:0]`
`si5402_mgmt_out`	SI5402 reset signal (1 bit input / output) [0] `si5402_rst_n`
`spare_wp`	Write protect for user-definable I^2C EEPROM [0] `spare_wp`
`user_led`	User-definable LEDs (6 bits output only) [5:0] `user_led_g_l[5:0]`
`user_switch`	User-definable switches (2 bits input only) [1:0] `user_sw_[1:0]`

Table 6: Available GPIO-style interfaces

The GPIO interfaces can be used in a Block Diagram design by right-clicking on an item and selecting ***Connect Board Component***. For an example of this method of connecting a board interface to an IP instance, see Section 3.4.

Alternatively, a compatible IP may be instantiated (for example, AXI GPIO) and customized to set its IP interfaces. For example, Figure 13 shows an AXI GPIO IP instance configured to use the `si5344_mgmt_in` and `si5344_mgmt_out` GPIO-style board interfaces:

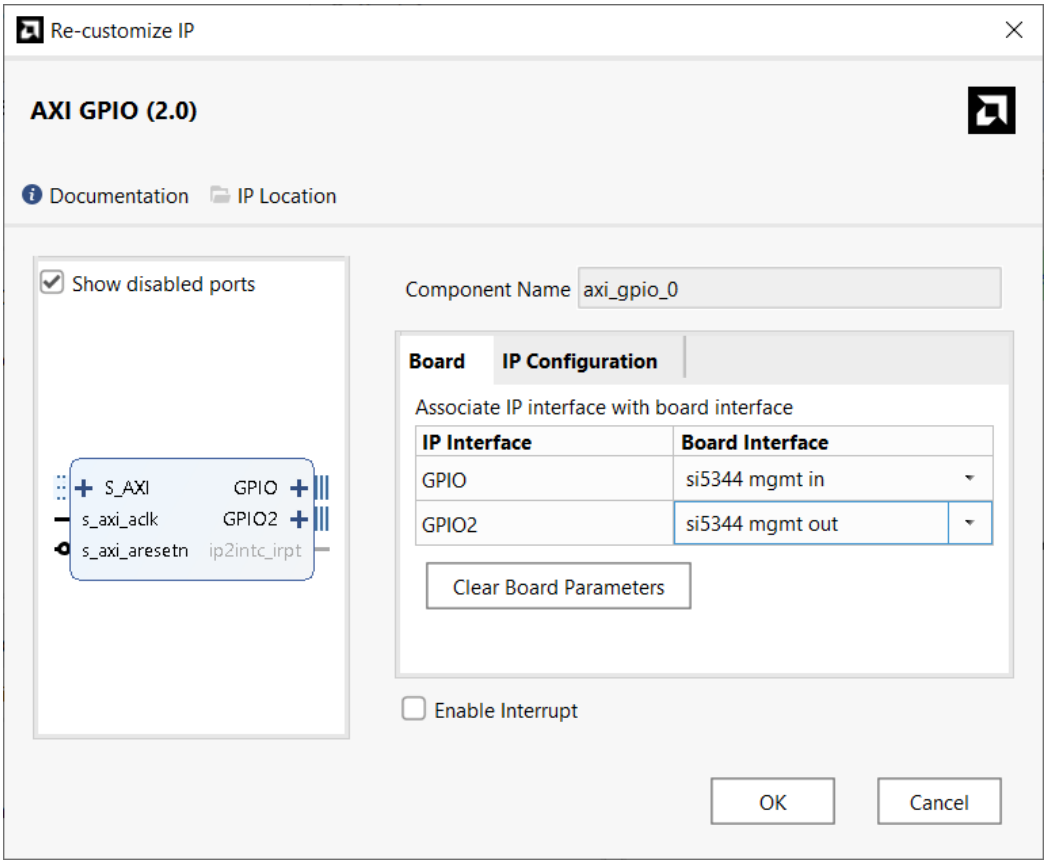


Figure 13: Configuring IP Interfaces of an AXI GPIO IP instance

3.7 I²C interfaces

The board file defines the following I²C interfaces:

Interface name	Description
`qsfp0_i2c`	QSFP-DD 0 module management interface; use I²C address 0x50.
`qsfp1_i2c`	QSFP-DD 1 module management interface; use I²C address 0x50.
`qsfp2_i2c`	QSFP-DD 2 module management interface; use I²C address 0x50.
`si5344_i2c`	Management interface for SI5344C jitter attenuator; use I²C address 0x68. This interface is also connected to the user-definable I2C EEPROM at I2C address 0x50.
`si5402_i2c`	Management interface for SI5402B network synchronizer; use I²C address 0x58.

Table 7: Available I²C interfaces

The I²C interfaces can be used in a Block Diagram design by right-clicking on an item and selecting ***Connect Board Component***. For an example of this method of connecting a board interface to an IP instance, see Section 3.4.

Alternatively, a compatible IP may be instantiated (for example, AXI GPIO) and customized to set its IP interfaces. For example, Figure 14 shows an AXI IIC IP instance configured to use the `qsfpdd0_i2c` board interface:

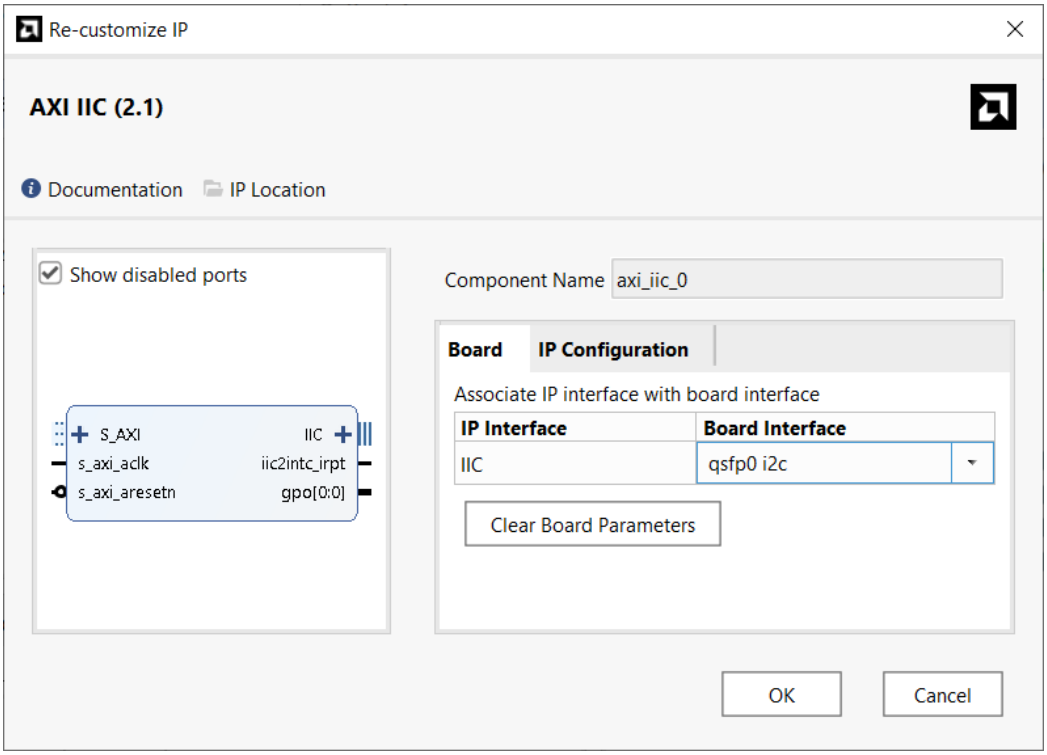


Figure 14: Configuring the IP Interface of an AXI IIC IP instance

4 Board features without board file support

4.1 PCIe connector

No board interfaces or IP presets are defined for the CPM PCIe blocks within the CIPS IP because (as of Vivado 2024.2) the CIPS IP does not have any board interface properties for its CPM PCIe module.

5 Known issues

This section describes issues that impact use of an ADM-PA120 board file in Vivado.

5.1 Single-ended clock board interfaces cannot be connected in IPI

In Vivado 2024.1 and 2024.2, IP Integrator does not permit single-ended clock interfaces defined in the board file to be connected to IPs, as in the example given in Figure 11; upon right-clicking such a clock interface, no IP choices are offered. It is unclear whether this is a bug introduced in Vivado 2024.1 or a permanent intentional change in Vivado.

However, it remains possible to set the board interface properties `CONFIG.CONFIG.CLK_IN1_BOARD_INTERFACE` & `CONFIG.CONFIG.CLK_IN2_BOARD_INTERFACE` for the Clock Wizard IP, either via Tcl scripting or using the Clock Wizard's IP configuration GUI.

5.2 Incorrect board constraints for Utility Buffer IP configured as `IBUFDS_GTME5`

In Vivado versions up to and including 2024.2, when the Utility Buffer IP is configured to use a differential clock board interface whose IP preset configures the Utility Buffer to be of type `IBUFDS_GTME5`, incorrect board con-

straints are generated when the IP or block diagram is synthesized. This results in the PACKAGE_PIN properties of top-level ports remaining undefined, with implementation failing at the Placement step.

The main symptom of this issue, besides failure of the Placement step, is that each implementation step, beginning with Design Initialization, results in two critical warnings of the form

```
[Netlist 29-160] Cannot set property 'BOARD_PART_PIN', because the property does
    ↪  not exist for objects of type 'pin'. ["/path/to/my_proj/my_proj.gen/
    ↪ sources_1/bd/top/ip/top_mgt_progclk_0_0/top_mgt_progclk_0_0_board.xdc
    ↪ ":<3>]
```

for each affected differential clock interface used. The cause is the constraints generated by the Utility Buffer IP, for example in the file top_mgt_progclk_0_0_board.xdc:

```
set_property BOARD_PART_PIN {mgt_progclk_0_n} [get_ports IBUF_DS_N]
set_property BOARD_PART_PIN {mgt_progclk_0_p} [get_ports IBUF_DS_P]
```

The IBUF_DS_N & IBUF_DS_P ports do not exist in a Utility Buffer IP instance that is configured as buffer type IBUFDS_GTME5.

The differential clock board interfaces affected by this issue are all of the clocks listed in Table 5 where CONFIG.C_BUF_TYPE is IBUFDS_GTME5. The solution is to include a user-created constraints (.xdc) file in the Vivado project that provides PACKAGE_PIN values for the affected top-level ports.

5.3 u-boot 2023.2 / 2024.1 (at least) fails to boot Petalinux from QSPI Flash in "x4 Dual Parallel" mode

The u-boot binary generated by Petalinux 2023.2 & 2024.1 (at least) fails to retrieve boot.scr, the kernel image etc. from QSPI Flash when the PS QSPI Flash controller is configured to operate in "x4 Dual Parallel" mode. A symptom of the problem is the following error message issued by u-boot:

```
U-Boot 2024.01 (May 14 2024 - 03:31:48 +0000)

CPU:   Versal
Silicon: v2
Chip:  v2
Model: Xilinx Versal
DRAM:  2 GiB (effective 16 GiB)
EL Level:       EL2
Core:  31 devices, 19 uclasses, devicetree: board
MMC:   mmc@f1050000: 0
Loading Environment from SPIFlash...  zynqmp_qspi spi@f1030000:  Invalid chip
    select 0:0 (err=-19)
*** Warning - spi_flash_probe_bus_cs() failed, using default environment
```

This issue is currently under investigation to determine the range of Petalinux versions affected and possible workarounds.

A verified workaround is to use the ps_pmc_fixed_io_linux_qspis CIPS IP preset, as described in Section 3.2.

6 Related documents

1. UG895 – Vivado Design Suite User Guide: System-Level Design Entry, AMD Inc. (formerly Xilinx Inc.)
2. UG994 – Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator, AMD Inc. (formerly Xilinx Inc.)
3. PG313 – LogiCORE IP Product Guide "Versal Adaptive SoC Programmable Network on Chip and Integrated Memory Controller", AMD Inc. (formerly Xilinx Inc.)

4. PG352 – LogiCORE IP Product Guide “Control Interfaces and Processing System”, AMD Inc. (formerly Xilinx Inc.)
5. AD-UG-1492 – ADM-PA120 User Manual, 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