



Arm[®] MPS4 FPGA Prototyping Board

Technical Reference Manual

Non-Confidential

Copyright © 2022, 2024 Arm Limited (or its affiliates).
All rights reserved.

Issue 02

102577_0000_02_en



Arm® MPS4 FPGA Prototyping Board Technical Reference Manual

This document is Non-Confidential.

Copyright © 2022, 2024 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

The product described in this document complies with the requirements set out in the included [conformance notice](#).

This document (102577_0000_02_en) was issued on 2024-07-19. There might be a later issue at <http://developer.arm.com/documentation/102577>

See also: [Proprietary Notice](#) | [Conformance notices](#) | [Product and document information](#) | [Useful resources](#)

Start Reading

If you prefer, you can skip to [the start of the content](#).

Intended audience

This document is written for experienced hardware and software developers to enable early software development and prototyping using the MPS4 board.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

Contents

1. The MPS4 FPGA Prototyping Board.....	6
1.1 Measures to ensure safety and prevent damage.....	7
1.2 MPS4 board layout.....	8
1.3 MPS4 board features.....	9
2. Getting started.....	11
2.1 Encryption battery warning.....	12
2.2 Install the SODIMM memory module.....	12
2.3 Power up into the operating state.....	13
2.4 Install an FPGA image.....	16
2.5 Install an executable software image.....	17
2.6 Install a custom FPGA bitfile.....	18
3. Hardware description.....	20
3.1 FPGA.....	23
3.1.1 Design settings for correct board operation with a minimal design.....	23
3.1.2 FPGA image packages.....	24
3.2 Memory and storage.....	24
3.2.1 FPGA DDR4 memory interface.....	25
3.2.2 User non-volatile memory.....	26
3.3 Clocks, resets, and power.....	27
3.3.1 Clocks.....	28
3.3.2 Reset, powerup, and configuration.....	30
3.3.3 Power.....	32
3.4 Motherboard Configuration Controller.....	33
3.4.1 Serial Configuration Controller interface.....	33
3.4.2 Static Memory Controller interface.....	35
3.5 Display, camera, and audio.....	39
3.5.1 HDMI display interface.....	39
3.5.2 Audio codec interface.....	40
3.5.3 Display Serial Interface.....	42
3.5.4 Camera Serial Interfaces.....	42
3.6 I/O connectivity.....	43

3.6.1	Arduino Shield, Pmod, and HAT interfaces.....	44
3.6.2	FMC+ HPC interface.....	48
3.6.3	On-board user components.....	49
3.6.4	Board-to-board interface and QSFP interface.....	50
3.6.5	Remote board farm interface.....	51
3.7	PCI Express systems.....	52
3.8	Interrupts.....	54
3.9	Debug.....	55
3.9.1	CoreSight™ debug and trace.....	55
3.9.2	Debug over USB.....	57
4.	Configuration.....	59
4.1	Using the reset switch and push buttons.....	60
4.2	Configuration files.....	61
4.2.1	config.txt board configuration file.....	63
4.2.2	Contents of the MB directory.....	64
4.2.3	Contents of the SOFTWARE directory.....	68
4.3	MCC command-line interface.....	68
4.3.1	MCC main command menu.....	68
4.3.2	MCC debug menu.....	70
4.4	Configuring remote farm access.....	70
4.5	Troubleshooting.....	74
A.	Connector and signal descriptions.....	75
A.1	Debug connectors.....	76
A.1.1	60-pin MIPI debug connector.....	76
A.1.2	20-pin CoreSight™ debug and ETM connector.....	78
A.1.3	10-pin CoreSight™ debug connector.....	80
A.1.4	14-pin F-JTAG ILA debug connector.....	81
A.2	I/O connectors.....	82
A.2.1	Arduino Shield, Pmod, and HAT shared SHx_IOx signals.....	82
A.2.2	Arduino Shield connectors.....	84
A.2.3	Pmod connectors.....	87
A.2.4	Raspberry Pi HAT and board farm connectors.....	90
A.2.5	FMC+ HPC connector.....	92
A.2.6	Samtec AcceleRate ARC6 connectors.....	93
A.3	Display, camera, and audio connectors.....	94

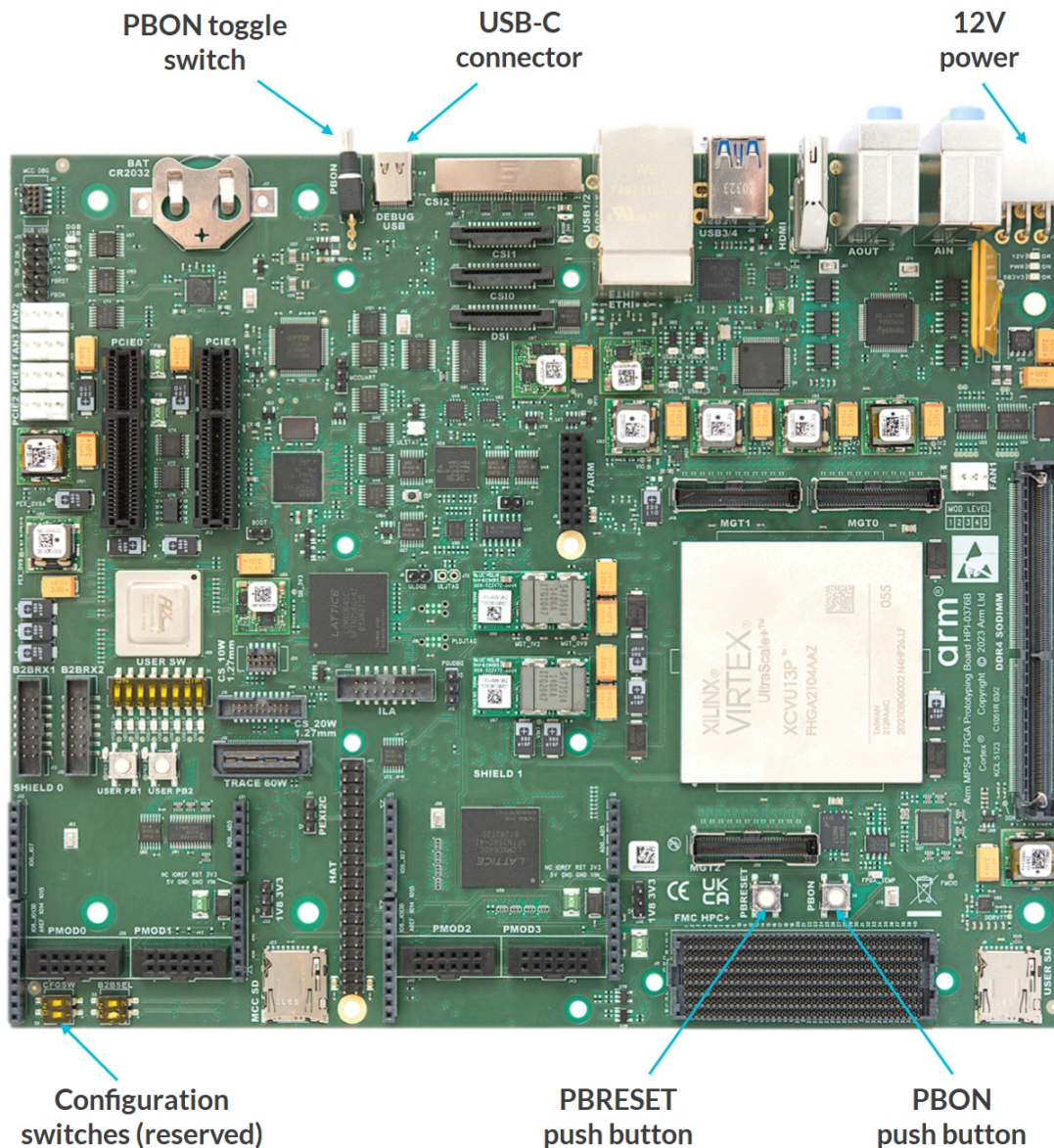
A.3.1 HDMI connector.....	95
A.3.2 Audio connectors, stacked stereo jacks.....	95
A.3.3 MIPI DSI connector.....	96
A.3.4 MIPI CSI connectors.....	97
A.4 PCIe Switch connectors.....	100
A.4.1 PCI Express connectors.....	100
A.4.2 Ethernet and USB 3.0 connectors.....	103
A.5 12V power connector.....	104
B. Available power for expansion boards.....	105
C. Powerup and configuration sequence.....	108
D. Extra hardware required for selftests.....	111
Proprietary Notice.....	113
Conformance notices.....	115
Product and document information.....	117
Product status.....	117
Revision history.....	117
Conventions.....	118
Useful resources.....	120

1. The MPS4 FPGA Prototyping Board

The MPS4 board is an FPGA development platform with increased capacity and updated peripherals over its predecessors. The board is designed to support Arm® Cortex®-A, Cortex®-R, and Cortex®-M processors, machine learning processors, and various subsystems, including Arm® Corstone™ reference systems.

The following figure shows the MPS4 board.

Figure 1-1: MPS4 FPGA Prototyping Board



To locate the main components on the MPS4 board, see [1.2 MPS4 board layout](#) on page 7.

1.1 Measures to ensure safety and prevent damage

You can take certain measures to ensure safety and to prevent damage to your product.

Ensuring safety

The onboard connector of the MPS4 board receives power from an external 12V DC power supply.



Do not use the MPS4 board near equipment that is sensitive to electromagnetic emissions, for example, medical equipment.

Operating temperature



The MPS4 board has been tested in the temperature range 0°C to 40°C.

We recommend that you do not operate the MPS4 board in temperatures outside this range.

Preventing damage

The MPS4 board is intended for use within a laboratory or engineering development environment.



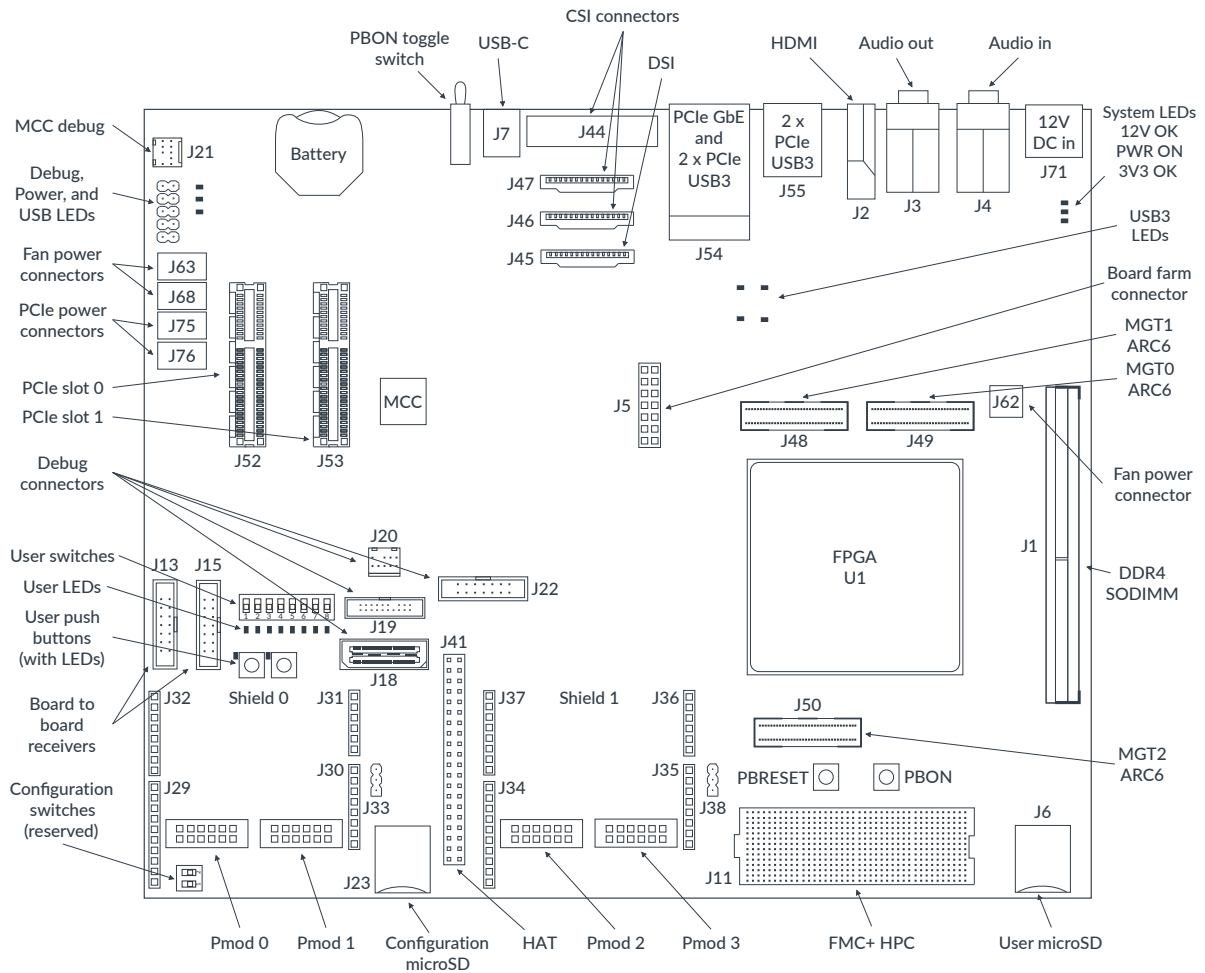
To avoid damage to the MPS4 board, observe the following measures:

- Connect the external power supply to the board before starting the powerup process.
 - Never subject the board to high electrostatic potentials. Observe *ElectroStatic Discharge* (ESD) precautions when handling any board.
 - Always wear a grounding strap when handling the board.
 - Only hold the board by the edges, and avoid bending or flexing the board.
 - Avoid touching the component pins or any other metallic element.
 - Do not connect any kind of board or storage device to the MPS4 board while it is powered.
-

1.2 MPS4 board layout

The following figure shows the physical layout of the MPS4 board and the location of the main components and connectors.

Figure 1-2: MPS4 board layout



The configuration switches, labeled CFGSW S9 (also shown in [Figure 1-1: MPS4 FPGA Prototyping Board](#) on page 6), are reserved. For correct operation, you must ensure that both switches are in the default, OFF position.

For a list of the MPS4 board connectors, see [A. Connector and signal descriptions](#) on page 75.

To learn more about the Shield, Pmod, and HAT power and I/O reference voltage user-links (J33 and J38), see [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47.

1.3 MPS4 board features

The MPS4 board contains an AMD Virtex Ultrascale+ VU13P FPGA, support logic, and peripheral interfaces that provide access to the FPGA and I/O interfaces.

MPS4 board uses

The MPS4 board enables FPGA prototyping of complex designs:

- Software development and validation:
 - Linux development on Arm® Cortex®-A, Cortex®-R, or Cortex®-M processors
 - Mbed™ OS, *Real-Time Operating System* (RTOS) development on Cortex®-M processors
 - Testing of Arm® Development Studio and DSTREAM family of probes
 - Bare-metal development, that is, without an OS, faster than simulation or emulation
- Software tool development
- IP evaluation

MPS4 board major components and systems

The MPS4 board provides the following components and systems:

- AMD Virtex Ultrascale+ VU13P FPGA with 3,780k logic cells and 1,728k *LookUp Tables* (LUTs)
- *Motherboard Configuration Controller* (MCC) that controls and configures the MPS4 board
- Memory:
 - 8 GB DDR4, with capacity for up to 16 GB
 - 128 MB user *Quad Serial Peripheral Interface* (QSPI) flash
 - Up to 11.8 MB of FPGA *Block RAM* (BRAM), and up to 45 MB *UltraRAM* (URAM)
 - microSD card interface
- *PCI Express* (PCIe) Gen 3 Switch with:
 - 2 PCIe Gen 3 x4 slots
 - 4 USB 3.0 slots
 - 10/100/1000M *Gigabit Ethernet* (GbE) port
- Expansion connectivity:
 - 2 Arduino Shield interfaces for custom peripherals
 - VITA FMC+ HPC expansion for up to 160 I/O, 10 *MultiGigabit Transceivers* (MGT), and clocks
 - 2 Pmod interfaces with 2 connectors each
 - HAT 28 I/O slot
 - 3 Samtec AcceleRate ARC6 24x connectors that you can use for connecting multiple MPS4 boards

- Remote access through board farm interface, compatible with a Raspberry Pi
- HDMI 1.4a port, supporting up to 1080p and I²S audio
- Surround-sound 5.1 I²S audio codec
- User board components:
 - 10 user LEDs
 - 8 user switches
 - 2 user push buttons
- Debug support:
 - 10-pin JTAG debug
 - 20-pin JTAG and ETM debug and trace
 - 60-pin MIPI debug and 32-bit trace
 - ARC6 24x QSFP debug interface
 - FPGA ILA connector for MCC debug software running on your FPGA image, Development Studio or Keil® µVision®
 - Onboard v1.2 Arm® Keil® ULINKplus™ Embedded debug adapter
 - 4 debug UARTs (1 Mbps)
 - High-speed (160 Mbps) 16-bit USB 3.0 FIFO Bridge

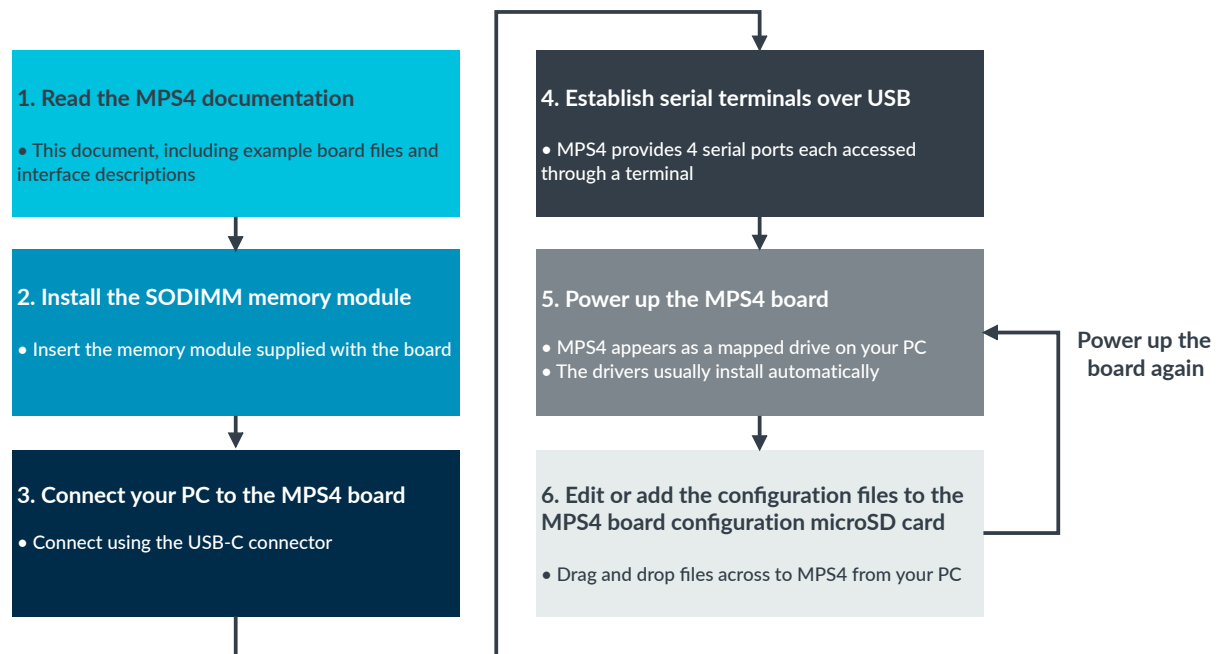
To learn more about the MPS4 board hardware, see [3. Hardware description](#) on page 20.

2. Getting started

You can connect and control the MPS4 board using the USB-C connector. The *Motherboard Configuration Controller* (MCC) configuration microSD card contains a set of files that configures the board. The MCC command-line interface provides user control of MPS4 board functionality.

The following figure summarizes the steps to get started with the MPS4 board.

Figure 2-1: MPS4 board getting started flow



For detailed steps, see:

1. [2.2 Install the SODIMM memory module](#) on page 12
2. [2.3 Power up into the operating state](#) on page 13
3. [2.4 Install an FPGA image](#) on page 16
4. [2.5 Install an executable software image](#) on page 17
5. [2.6 Install a custom FPGA bitfile](#) on page 18

2.1 Encryption battery warning

Arm prebuilt FPGA images are encrypted using a key that matches a preprogrammed decryption key in the FPGA on the MPS4 board. Decryption happens automatically when you load the image into the FPGA. You do not need to take any additional steps.

A battery supplies power to the area where the key is stored in the FPGA on the board.



Do not remove the battery from the board. If battery power is lost or depleted, the FPGA might lose its preprogrammed decryption key. If this happens, return the board to Arm for reprogramming of the key.

2.2 Install the SODIMM memory module

The MPS4 board is supplied with a SODIMM memory module. This module is not installed at the factory to reduce the possibility of damage during transport.

About this task



While installing the memory module, follow anti-static handling precautions to avoid damaging the board or the module with *ElectroStatic Discharge* (ESD). See [1.1 Measures to ensure safety and prevent damage](#) on page 7.

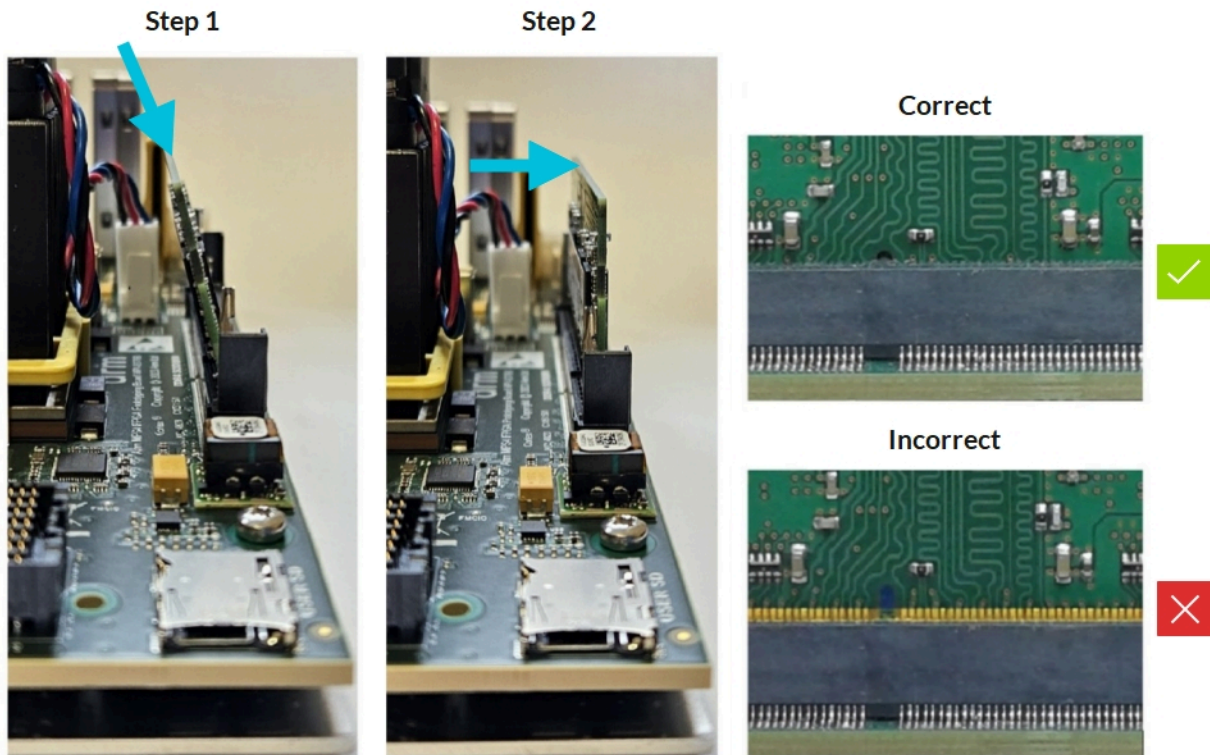
Procedure

1. Insert the memory module into the socket at an angle, oriented so that the slot in the edge-connector of the module aligns with the keyring bar across the opening in the socket. Make sure you push it in completely, otherwise it will not make proper contact with the pins of the socket.
2. Push the memory module into the vertical position. It will engage with the catches at the ends of the socket, making an audible clicking noise.

Example 2-1: Correct and incorrect insertion

The following figure shows how to correctly insert the memory module.

Figure 2-2: SODIMM memory module insertion



Next steps

- [2.3 Power up into the operating state](#) on page 13

Related information

1. [The MPS4 FPGA Prototyping Board](#) on page 6
- 1.2 [MPS4 board layout](#) on page 7
- 3.2.1 [FPGA DDR4 memory interface](#) on page 25

2.3 Power up into the operating state

The following steps describe how to power up the MPS4 board into the basic operating state.

Before you begin

- Familiarize yourself with this manual and its description of the MPS4 board configuration and boot flows
- [2.2 Install the SODIMM memory module](#) on page 12
- Understand how to power up, reset, and establish a serial terminal over a USB connection to a host computer

- Check that both configuration switches (CFGSW S9) are in the default, OFF position. To locate the switches, see [1.2 MPS4 board layout](#) on page 7.

Procedure

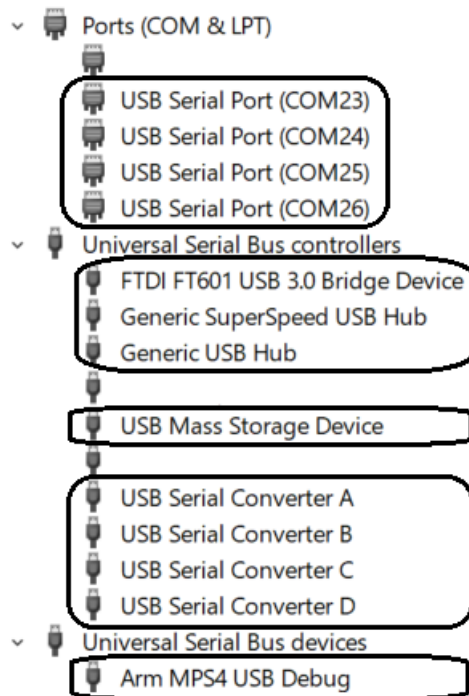
1. Connect your host computer to the MPS4 board through the USB-C connector.
2. Establish serial terminals for each serial port over USB using a terminal emulator:
 - Your host computer virtual serial port settings must be:
 - 115200 baud rate
 - 8N1, that is, 8 data bits, no parity, and 1 stop bit
 - No hardware or software flow control
 - By default, the serial ports are connected as follows:
 - Serial Port 0 is connected to the *Motherboard Configuration Controller* (MCC) UART0 command-line interface, see [4.3 MCC command-line interface](#) on page 68
 - Serial Port 1 is connected to the FPGA UART1
 - Serial Port 2 is connected to the FPGA UART2
 - Serial Port 3 is connected to the FPGA UART3



The serial port numbers vary from one host computer to another. In the Device Manager of a Windows PC, the ports appear in a block of four consecutive numbers [n], such as COM[n], COM[n+1], COM[n+2], and COM[n+3].

The following example figure shows the USB devices that the MPS4 board adds when you connect a Windows PC to the USB-C connector.

Figure 2-3: MPS4 board software drivers



The software drivers normally install automatically. However, for some Windows PCs you might need to install the drivers separately. If the ports are not visible in the Device Manager, you might not have the correct device drivers installed. You can download the USB UART device (FT4232H) Windows software drivers at <https://ftdichip.com>.

3. Connect the supplied 12V power supply adapter to the 12V power connector on the MPS4 board.
The 12VOK light turns on.
4. Press the white *PBON* hardware push button or *PBON* toggle switch.
The *PWRON* light turns on. The board initializes. The MCC command-line interface shows the log.

Results

The MPS4 board is now fully powered and in the operating state. The FPGA is preloaded with the default factory image.

Next steps

- To power down the MPS4 board so that it enters standby mode, perform any one of these actions:
 - Press the white *PBRESET* hardware push button.

- Hold the *PBON* toggle switch or white *PBON* hardware push button down for more than 2 seconds.
- Run the `shutdown` command in the MCC command-line interface.
- To initiate a soft reset, press the white *PBON* hardware push button or *PBON* toggle switch.
- [2.4 Install an FPGA image](#) on page 16
- [2.5 Install an executable software image](#) on page 17
- [2.6 Install a custom FPGA bitfile](#) on page 18

2.4 Install an FPGA image

Arm provides FPGA images for use on the MPS4 board. FPGA image packages contain an Application Note document describing the image, configuration files, and (optionally) demonstration example software. The package includes a `Boardfiles` directory that contains configuration files for you to copy to the board.

Before you begin

[2.3 Power up into the operating state](#) on page 13

About this task

The configuration files are installed on the MPS4 board using the USB-C connector. When the MPS4 board is connected to a host computer, the configuration microSD card appears as a USB mass storage device, `v2M_MPS4`. The configuration microSD card contains pre-installed configuration files.

To learn more about the configuration files, see [4.2 Configuration files](#) on page 61. For more information on a particular FPGA image package, see the Application Note document that accompanies it.

Procedure

1. Connect your host computer to the USB-C connector on the MPS4 board.
The system is now in the standby state. Your host computer recognizes the configuration microSD card as a USB mass storage device that appears as `v2M_MPS4`.



If the USB mass storage device does not appear, run the `usb_on` command in the MCC command-line interface. For more information, see [2.3 Power up into the operating state](#) on page 13.

2. Save a copy of any file on `v2M_MPS4` that you want to retain for future use.
3. Format `v2M_MPS4`. We recommend using FAT16, but alternatively you can use FAT32.
4. Copy the files from the `Boardfiles` directory to `v2M_MPS4`, preserving the directory structure. Keep the file `config.txt` at the root of `v2M_MPS4`.
5. Close all open files and then eject the USB mass storage device, `v2M_MPS4`.

6. Reboot the board. You can perform any one of these actions:

- Press the white *PBRESET* hardware push button and then press the white *PBON* hardware push button.
- Hold the *PBON* toggle switch down for more than 2 seconds and then hold the switch down again briefly.
- Run the `reboot` command in the MCC command-line interface.

Results

The MPS4 board loads with the new FPGA image.

Next steps

- [2.5 Install an executable software image](#) on page 17
- [2.6 Install a custom FPGA bitfile](#) on page 18

Related information

[3.1.2 FPGA image packages](#) on page 24

[4.2 Configuration files](#) on page 61

2.5 Install an executable software image

You can install software images onto the MPS4 board from your host computer using the USB-C connector.

Before you begin

[2.3 Power up into the operating state](#) on page 13

Procedure

1. Connect your host computer to the USB-C connector on the MPS4 board.
The system is now in the standby state. Your host computer recognizes the configuration microSD card as a USB mass storage device that appears as `v2M_MPS4`.



If the USB mass storage device does not appear, run the `usb_on` command in the MCC command-line interface. For more information, see [2.3 Power up into the operating state](#) on page 13.

2. Add the new software binary to the `SOFTWARE` directory of `v2M_MPS4`.
For example:

```
<V2M_MPS4_drive>\SOFTWARE\memtest.axf
```

3. Edit the `images.txt` file to add an entry for the new software binary.

For example:

```
TITLE: Arm MPS4 FPGA prototyping board Images Configuration File

[IMAGES]
TOTALIMAGES:1                ;Number of Images (Max: 32)
IMAGE0ADDRESS: 0X0000000     ;Select the required executable program
IMAGE0UPDATE: FORCE          ;Image update:NONE/AUTO/FORCE
;IMAGE0FILE: \SOFTWARE\selftest.axf ;Selftest image
IMAGE0FILE: \SOFTWARE\memtest.axf ;Add new entry, comment out old entries
```

4. Reboot the board. You can perform any one of these actions:
 - Press the white *PBRESET* hardware push button and then press the white *PBON* hardware push button.
 - Hold the *PBON* toggle switch down for more than 2 seconds and then hold the switch down again briefly.
 - Run the `reboot` command in the MCC command-line interface.

Results

The MCC loads the new software image into the memory regions accessible through FPGA design.

2.6 Install a custom FPGA bitfile

To install a custom FPGA bitfile on the MPS4 board, save it to the correct location in the configuration microSD card directory structure.

Before you begin

[2.3 Power up into the operating state](#) on page 13

About this task



File names and directory names are in 8.3 short file name format:

- File names that you generate must be in lowercase.
- Directory names must be in uppercase.
- All configuration files must use DOS line endings, that is, 0x0D/0x0A, or \n and \r, or LF and CF.

Procedure

1. Connect your host computer to the USB-C debug port on the MPS4 board.
The system is now in the standby state. Your host computer recognizes the configuration microSD card as a USB mass storage device that appears as `v2M_MPS4`.



If the USB mass storage device does not appear, run the `usb_on` command in the MCC command-line interface. For more information, see [2.3 Power up into the operating state](#) on page 13.

2. Copy the custom FPGA bitfile, with the filename extension `.bit`, to the appropriate directory under `MB` on `v2M_MPS4`.
For example:

```
<v2M_MPS4_drive>\MB\HBI0376B\CUSTOM\custom.bit
```

3. Specify the custom FPGA bitfile in the FPGA implementation `.txt` file, usually `fixxx_vx.txt`, which specifies clock frequencies and other configuration information. This file must point to the correct FPGA bitfile, for example:

```
[FPGAS]
TOTALFPGAS: 1                ;Total Number of FPGAs
F0FILE: custom.bit           ;FPGA0 Filename
F0MODE: FPGA                 ;FPGA0 Programming Mode
```

To learn more about this `.txt` file, see [4.2.2.2 FPGA implementation .txt file](#) on page 65.



You can also change the name of the FPGA implementation `.txt` file to match the name of the new FPGA bitfile, for example, `custom.txt`. However, you must then edit the `board.txt` file in the directory above (`<v2M_MPS4_drive>\MB\HBI0376B\board.txt`) so that it has the path to the FPGA implementation `.txt` file. For example:

```
[FPGA IMPLEMENTATION]
APPFILE: CUSTOM\custom.txt    ;Your custom design
```

4. Reboot the board. You can perform any one of these actions:
 - Press the white *PBRESET* hardware push button and then press the white *PBON* hardware push button.
 - Hold the *PBON* toggle switch down for more than 2 seconds and then hold the switch down again briefly.
 - Run the `reboot` command in the MCC command-line interface.

Results

The FPGA on the MPS4 board loads with the custom FPGA bitfile.

Related information

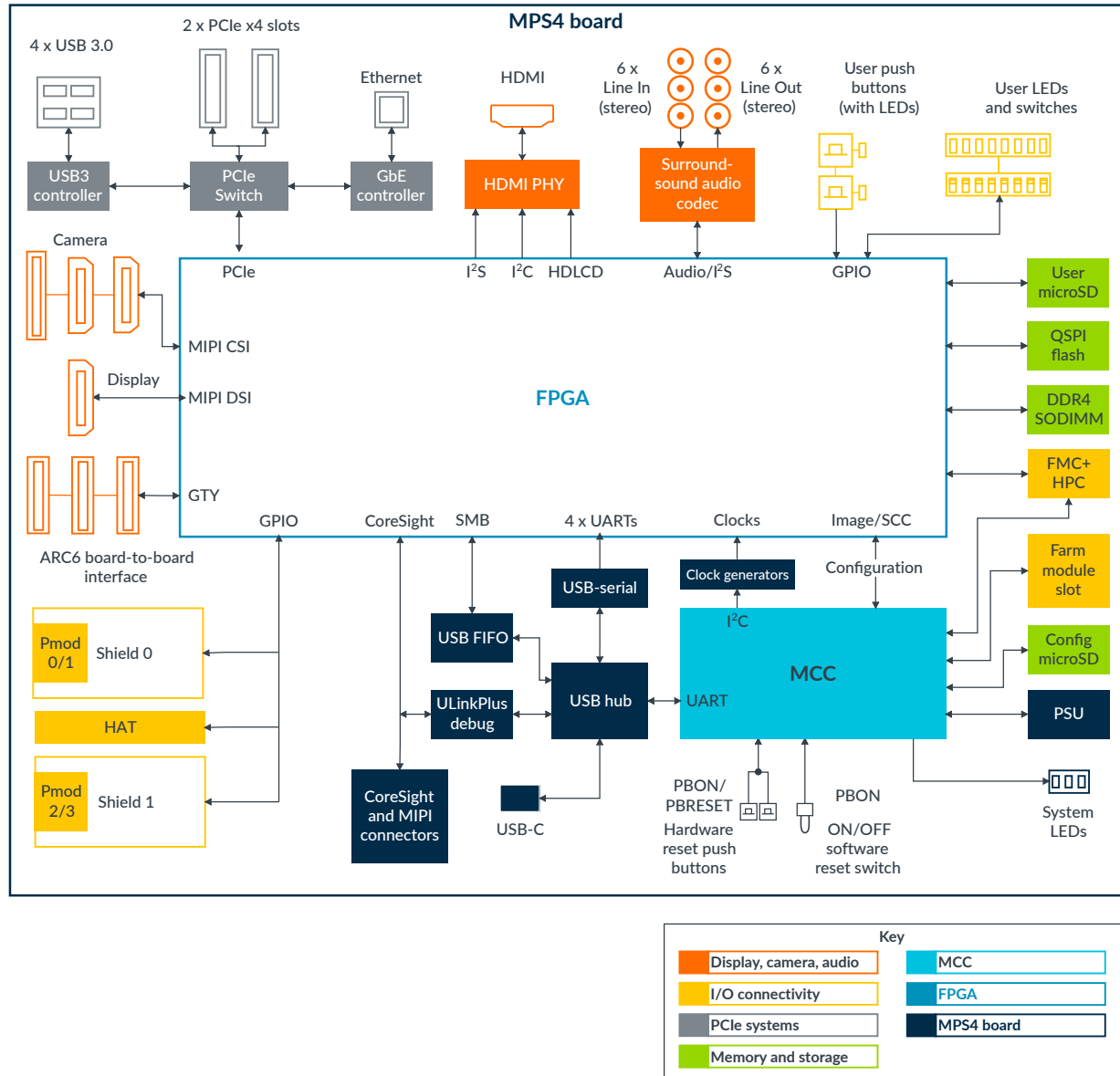
[4.2 Configuration files](#) on page 61

3. Hardware description

The MPS4 board provides access to the AMD Virtex Ultrascale+ VU13P FPGA and peripherals to enable FPGA prototyping and software development.

The following figure shows an overview of the MPS4 board.

Figure 3-1: MPS4 board system overview



The MPS4 board contains the following components.

FPGA, see [3.1 FPGA](#) on page 23

- AMD Virtex Ultrascale+ VU13P FPGA with 3,780k logic cells, 1,728k *LookUp Tables* (LUTs), and 6 programmable clocks
- Up to 11.8 MB of FPGA *Block RAM* (BRAM) and up to 45 MB *UltraRAM* (URAM)

Motherboard Configuration Controller, see [3.4 Motherboard Configuration Controller](#) on page 33

- Controls the MPS4 board, and supports board configuration at powerup or reset
- Uses an Arm® Cortex®-M7 processor clocked at 216 MHz
- Configures the FPGA
- Configures the board
- Supervises user update of the board configuration files in the configuration microSD card
- Includes 1 UART connected to a USB to serial hub for connection to the host computer, and multiplexed to the other 4 UARTs on the FPGA
- Supports 2 hardware reset buttons, one labeled *PBRESET* and one labeled *PBON*, and an ON/OFF software reset switch that is labeled *PBON*, see [3.3.2 Reset, powerup, and configuration](#) on page 30

User memory system, see [3.2 Memory and storage](#) on page 24

- 8 GB, 64-bit DDR4 SODIMM (with support for up to 16 GB single-rank or dual-rank RAM)
- microSD card interface
- 128 MB QSPI flash

PCI Express systems, see [3.7 PCI Express systems](#) on page 52

- PCI Express Gen 3 Switch
- 2 PCIe Gen 3 x4 slots
- 4-port USB 3.0 controller
- *Gigabit Ethernet* (GbE) controller with 10/100/1000M Ethernet port

1080p HDMI 1.4a port and HDMI controller

- Inputs 24-bit RGB data from the HDLCD controller in the FPGA
- Configured over I²C directly from FPGA
- Supports an I²S audio connection to the FPGA

MIPI display and camera ports

- 1 DSI port, supporting up to 2 D-PHY channels with up to 1.25 Gbps per channel, see [3.5.3 Display Serial Interface](#) on page 41
- 3 CSI ports, supporting multiple D-PHY channels with up to 1.25 Gbps per channel, see [3.5.4 Camera Serial Interfaces](#) on page 42

Audio codec, see [3.5.2 Audio codec interface](#) on page 40

- Provides a 6-channel surround-sound 5.1 line-in and line-out stacked jack audio interface

- Configured over I²C from the I²C controller in the FPGA
- I²S audio connection to the I²S in the FPGA

Arduino Shield expansion, see [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43

- 2 Arduino Shield expansion interfaces for custom peripherals or off-the-shelf sensor interfaces, for example, WiFi, Bluetooth, proximity detectors, or gyro sensors
- Each interface connects 16 × digital 3V3 I/O or 16 × digital 1V8 I/O, voltage references, and 6 analog inputs from each Shield. The Shield interfaces share I/O pins with the Pmod and HAT interfaces.

Peripheral module (Pmod) interface expansion, see [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43

- 4 Pmod expansion connectors that form 2 Pmod interfaces, each with 16 digital I/O
- Share I/O pins with the Shield and HAT interfaces
- Type 2A/3/4 support
- Requires I/O voltage level to be set to 3V3

Raspberry Pi HAT interface, see [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43

- Shares pins with the Shield and Pmod interfaces to give 28 I/O
- Supports HAT boards and peripherals, for example, an LCD, sensor, or a QSFP board
- Requires I/O voltage level to be set to 3V3

FMC+ HPC expansion, see [3.6.2 FMC+ HPC interface](#) on page 48

- Connector enables you to fit a 560-pin *FPGA Mezzanine Card Plus* (FMC+) or *FMC High Pin Count* (HPC) expansion card to the MPS4 board. However, only the 400 pins from the FMC standard are actually connected.
- 160 LVDS I/O supporting 1V2, 1V5, and 1V8, with 10 multigigabit links

Board-to-board interface and QSFP interface, see [3.6.4 Board-to-board interface and QSFP interface](#) on page 50

- 3 12-lane multigigabit links supporting connections up to 25 Gbps per lane for board-to-board links (J48 and J49) and up to 12.5 Gbps per lane for the QSFP interface (J50)
 - 3 Samtec AcceleRate ARC6 24x connectors for connecting multiple MPS4 boards

User switches, LEDs, and push buttons that connect directly to GPIO in the FPGA, see [3.6.3 On-board user components](#) on page 49

- 8-way user DIP switch
- 8 user LEDs
- 2 user push buttons each with an LED

System LEDs

- Green 12V OK LED
- Orange 3V3 OK LED
- Green PWR ON LED

- Green *MCC active* LED
- Orange *Debug USB active* LED
- Green LED. Ethernet speed indicator-incorporated into combined Ethernet and dual-USB connector.
- Yellow LED. Ethernet link and activity indicator-incorporated into combined Ethernet and dual-USB connector.

3.1 FPGA

The MPS4 board uses the AMD Virtex Ultrascale+ VU13P FPGA. The Ultrascale+ VU13P FPGA has enough capacity for small GPUs, can run Android, and supports encrypted FPGA images from Arm.

The following table shows the key features of the Ultrascale+ VU13P FPGA (part number: XCVU13P-1FHGA2104E).

Table 3-1: Ultrascale+ VU13P FPGA features

Feature	Description
Logic cells	3,780k
LookUp Tables (LUTs)	1,728k
Clock inputs	6 programmable clocks (OSCCLKs) and 12 low-jitter MGT reference clocks, see 3.3.1 Clocks on page 27
Block RAM (BRAM) and UltraRAM (URAM)	11.8 MB BRAM and 45 MB URAM, which are internal to the FPGA. BRAM and URAM have lower latency than external memory like DDR or QSPI.
Support for encrypted FPGA images	256-bit AES-GCM hardware encryption with battery-backed keys

Related information

- Learn more about the Ultrascale+ VU13P FPGA at www.amd.com
- Download the FPGA pinout as an .xlsx spreadsheet at <https://developer.arm.com/mps4>

3.1.1 Design settings for correct board operation with a minimal design

For correct operation with an FPGA implementation for a minimal design, the MPS4 board requires some design settings. You must have a minimum amount of RTL in the FPGA, and certain parameter settings in the `config.txt` file.

Minimum RTL

The following table shows the signals that you must tie off in the FPGA to generate the minimum RTL for correct operation of the MPS4 board.

Table 3-2: Minimum RTL for correct operation of the MPS4 board

FPGA signal	Minimum RTL
MMB_IDCLK	Tie LOW
MMB_SCK	Tie LOW
QSPI_nCS	Tie HIGH
QSPI_SCLK	Tie LOW
IOFPGA_SYSWDT	Tie LOW
WDOG_RREQ	Tie LOW
SMBM_nWAIT	Tie HIGH
CFG_DATAOUT	Tie LOW
AUD_RST	Tie LOW
USD_CLK	Tie LOW
SH_nRST	Tie LOW
PEX_nPERST	Tie LOW

Configuration file settings

If your FPGA implementation does not implement the *Serial Configuration Controller* (SCC), you must set the `FPGA_scc` parameter to `FALSE` in the board `config.txt` file.

If your FPGA implementation does not implement the MCC-SMC interface, you must set the `FPGA_SMB` parameter to `FALSE` in the board `config.txt` file.

To learn more about the `config.txt` file, see [4.2.1 config.txt board configuration file](#) on page 63.

3.1.2 FPGA image packages

Arm provides FPGA images developed for the MPS4 board. These FPGA images model Arm hardware subsystems.

You can download the FPGA image packages from <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>.

Each package contains an explanatory Application Note document and the necessary board files required to implement the image on the MPS4 board. For more information about the configuration files that each package contains, see [4.2 Configuration files](#) on page 61.

3.2 Memory and storage

The MPS4 board provides:

- 8 GB DDR4 SODIMM, with support for up to 16 GB single-rank or dual-rank RAM
- 128 MB QSPI flash

- A user microSD card that supports 1-bit SPI or 4-bit QSPI controllers

In addition, the Ultrascale+ VU13P FPGA has 11.8 MB of *Block RAM* (BRAM) and 45 MB of *Ultra RAM* (URAM), which you can implement in the design if required.

3.2.1 FPGA DDR4 memory interface

The MPS4 board provides 8 GB of DDR4 SODIMM and a DDR4 interface to the FPGA. Up to 16 GB is supported.

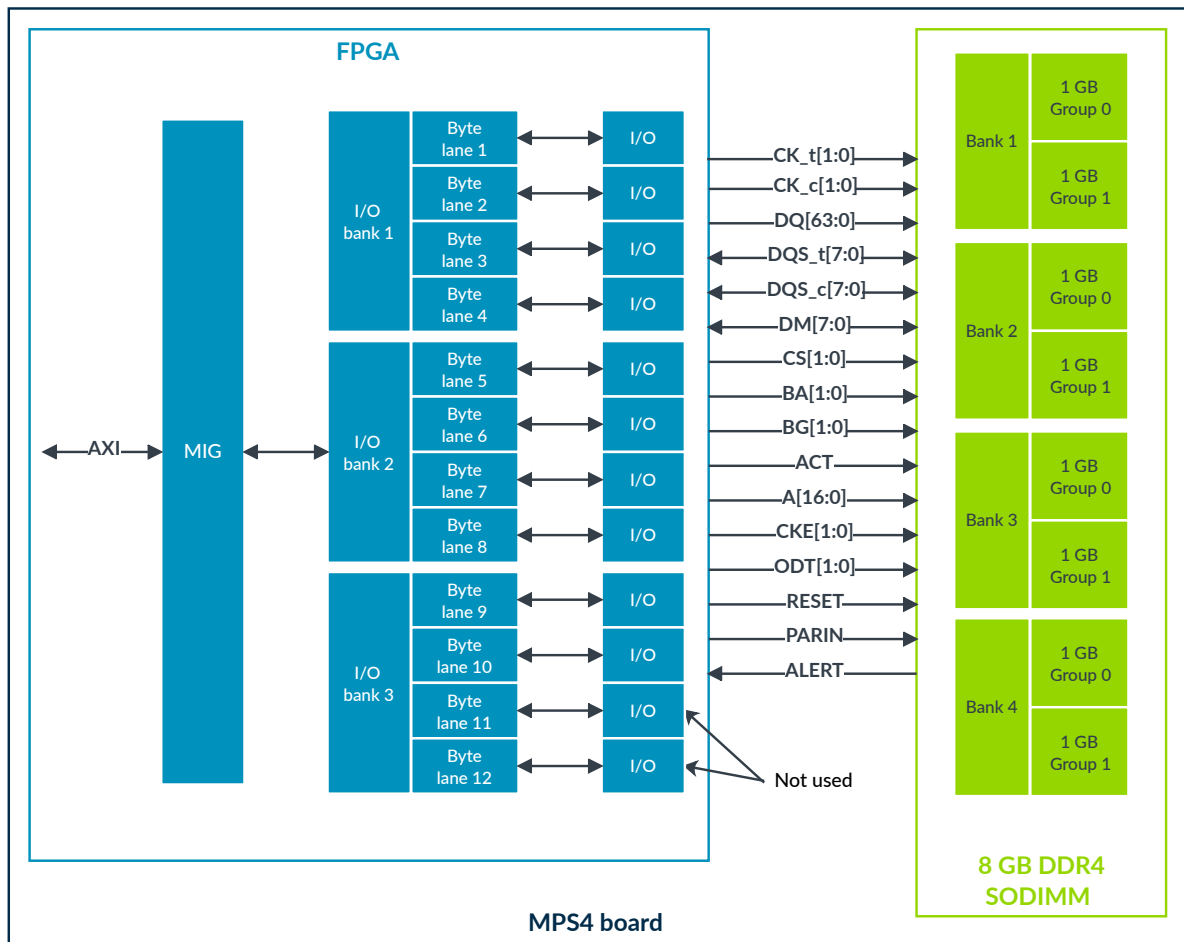
The DDR4 controller and 64-bit PHY interface use the *AMD Memory Interface Generator* (MIG).

The interface supports up to 900 MHz, 1800 MT/s, with the supplied SODIMM. The default 8 GB SODIMM module is Micron MTA8ATF1G64HZ-2G6E1. The following SODIMM modules have been tested with the MPS4 board:

- 8 GB single-rank MTA8ATF1G64HZ-2G6
- 16 GB single-rank MTA8ATF2G64HZ-3G2
- 16 GB dual-rank MTA16ATF2G64HZ-2G6

The following figure shows the FPGA DDR4 memory interface.

Figure 3-2: FPGA DDR4 memory interface



3.2.2 User non-volatile memory

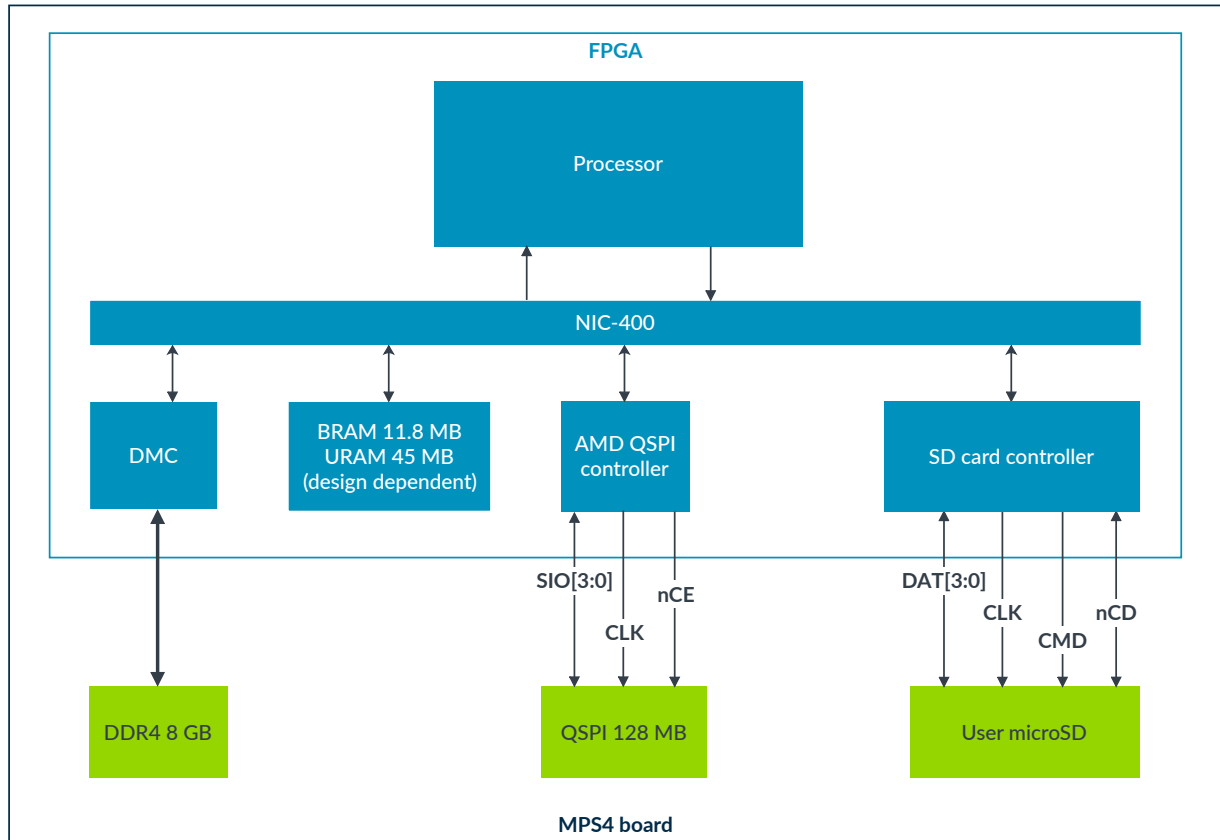
The MPS4 board provides on-board user non-volatile memory, 128 MB QSPI flash (Micron MT25QU01G) and a user microSD card interface.

A typical use of the QSPI flash is as boot memory for a soft processor implemented in the FPGA.

You can use the user microSD card to store user data. Do not confuse the user microSD card (J6) with the configuration microSD card (J23).

The following figure shows a non-volatile memory system example design. For completeness, the figure includes the DDR4 volatile memory, *Block RAM* (BRAM), and *UltraRAM* (URAM).

Figure 3-3: MPS4 board non-volatile memory system example design



The simplest boot method is to use BRAM in the FPGA that the MCC can preload before resets are released. The use of BRAM requires the FPGA design to implement the MCC-SMC interface. See [3.4.2 Static Memory Controller interface](#) on page 35.

3.3 Clocks, resets, and power

The MPS4 board provides:

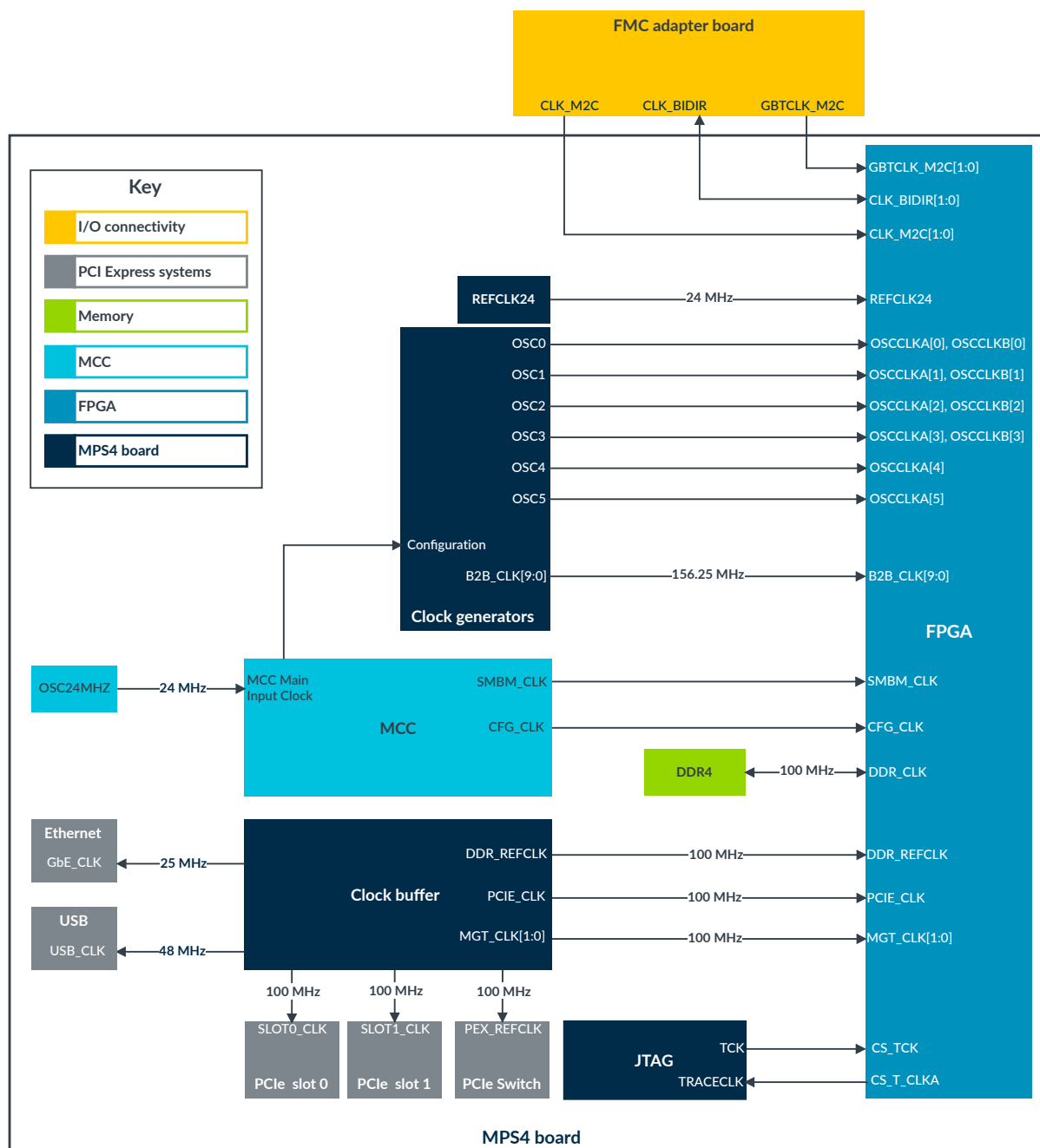
- Programmable and fixed clocks for the FPGA and board peripherals
- External resets to the FPGA
- 252W power using an external power unit

3.3.1 Clocks

The MPS4 board provides fixed and programmable clocks to drive the FPGA and board peripheral interfaces. The programmable clock generators on the MPS4 board generate clocks for the FPGA and board peripherals.

The following figure shows a functional overview of the clock systems of the MPS4 board.

Figure 3-4: MPS4 board clocks overview



The *Motherboard Configuration Controller* (MCC) configures the programmable *OSCillator* (OSC) clocks at powerup using the default frequencies. You can set the frequencies in the MPS4 board configuration FPGA implementation .txt file, usually `F1xxx_vx.txt`, in the configuration microSD card. To learn more about the configuration files, see [4.2 Configuration files](#) on page 61.

The system register interface can implement runtime control of the OSCs. The system register interface requires you to implement the `SYS_CFG` register in the *Serial Configuration Controller* (SCC) interface. OSC clocks `OSCCLK[0]` to `OSCCLK[3]` each have 2 outputs that are the same frequency but connect to different banks on the FPGA to help FPGA routing.

PLLs within the FPGA can optionally use the fixed reference 24 MHz clock to generate other fixed internal frequencies.

The following table describes the MPS4 board clocks and their characteristics.

Table 3-3: MPS4 board clocks

Clock source	Programmable frequency range	Destination	Description
REFCLK24	24 MHz fixed	FPGA REFCLK24]	Typically, source clock for these signals: <ul style="list-style-type: none"> • SYSPLL • TIMCLK • CLK100Hz • UARTCLK
OSC0	2-230 MHz	FPGA OSCCLKA[0], FPGA OSCCLKB[0]	System clock (optional)
OSC1	2-230 MHz	FPGA OSCCLKA[1], FPGA OSCCLKB[1]	CPU0 (optional) clock
OSC2	2-230 MHz	FPGA OSCCLKA[2], FPGA OSCCLKB[2]	CPU1 (optional) clock
OSC3	2-230 MHz	FPGA OSCCLKA[3], FPGA OSCCLKB[3]	Audio (optional) clock
OSC4	2-230 MHz	FPGA OSCCLKA[4]	QSPI (optional) clock
OSC5	2-230 MHz	FPGA OSCCLKA[5]	HDMI (optional) clock, used by HDCLD controller. Note: OSC5 is powered from the FPGA FMC bank, which must be powered to use OSC5. See the <code>FMC_FORCE</code> parameter in <code>config.txt</code> .
GTCLK	100 MHz fixed	DDR PHY	DDR PHY requires a fixed 100 MHz frequency.
TCK	25 MHz fixed	CS_TCK	Source clock for JTAG debug system. Its frequency depends on the debugger setting.
TRACECLK	Up to 150 MHz	CS_T_CLKA	CoreSight™ clock used for JTAG
CFG_CLK	Up to 10 MHz	SCC	FPGA configuration clock from the MCC
PCIE_CLK	100 MHz fixed	PCIE	PCIE links from the PCIE Switch
MGT_CLK[1:0]	100 MHz fixed	FPGA	Reference clocks for the <i>FPGA Mezzanine Card</i> (FMC) and <i>MultiGigabit Transceiver</i> (MGT) links

Clock source	Programmable frequency range	Destination	Description
B2B_CLK[9:0]	156.25 MHz	FPGA	10 board-to-board MGT clocks, provided by the board-to-board clock generator on the MPS4 board. These clocks default to 156.25 MHz.
DDR_REFCLK	100 MHz fixed	FPGA	DDR4 PHY reference clock
CLK_M2C[1:0]	0-100 MHz	FPGA	FMC clock, Mezzanine to Carrier
CLK_BIDIR[1:0]	0-100 MHz	FMC/FPGA	FMC clock, Carrier to Mezzanine
GBTCLK_M2C[1:0]	0-100 MHz	FPGA	FMC <i>GigaBit Transceiver</i> (GBT) clock, Mezzanine to Carrier

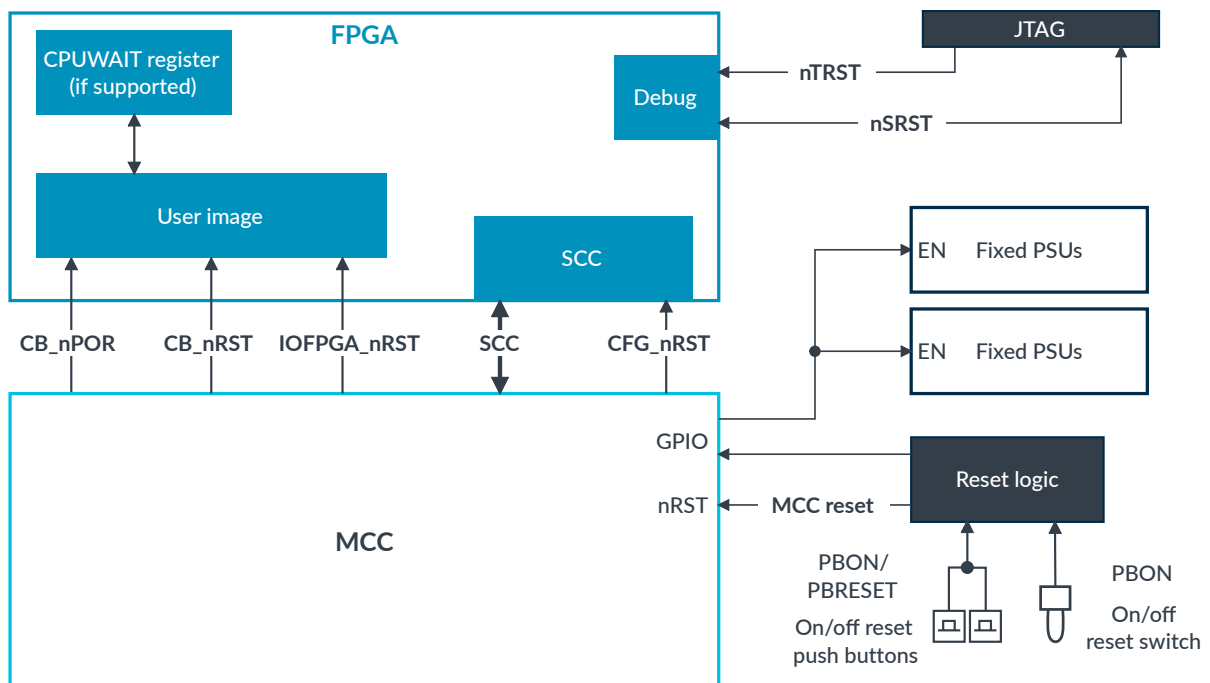
3.3.2 Reset, powerup, and configuration

The MPS4 board provides external resets to the FPGA, 2 reset push buttons at the front of the board and an ON/OFF reset switch at the back of the board.

For more information about the reset push buttons and reset switch, see [4.1 Using the reset switch and push buttons](#) on page 60. To locate the buttons and switch, see [1.2 MPS4 board layout](#) on page 7.

The following figure shows the MPS4 board reset system, where the FPGA contains a user image.

Figure 3-5: MPS4 board reset system



FPGA resets

The FPGA resets are as follows:

IOFPGA_nRST

Resets the board peripherals and FPGA, including the FPGA PLLs

CB_nPOR

Main Cold reset for the FPGA image logic. If a *System Control Processor* (SCP) is present in the design, releasing this reset might also trigger the Cold reset sequencing.

nTRST

Resets the CoreSight™ DAP and TAP controllers for boundary scan and *Memory Built-In Self Test* (MBIST)

CB_nRST, nSRST

Warm resets. These inputs are typically ANDed together in the FPGA. They initiate operation of the processors and enable the debug tools to debug the processors before they leave reset.



You can also configure your FPGA design to have a CPUWAIT reset.

CB_RUN

Indicates the system is out of reset and running. This is an input to the FPGA.

CFG_nRST

Reset signal for the serial interface of the *Serial Configuration Controller* (SCC).

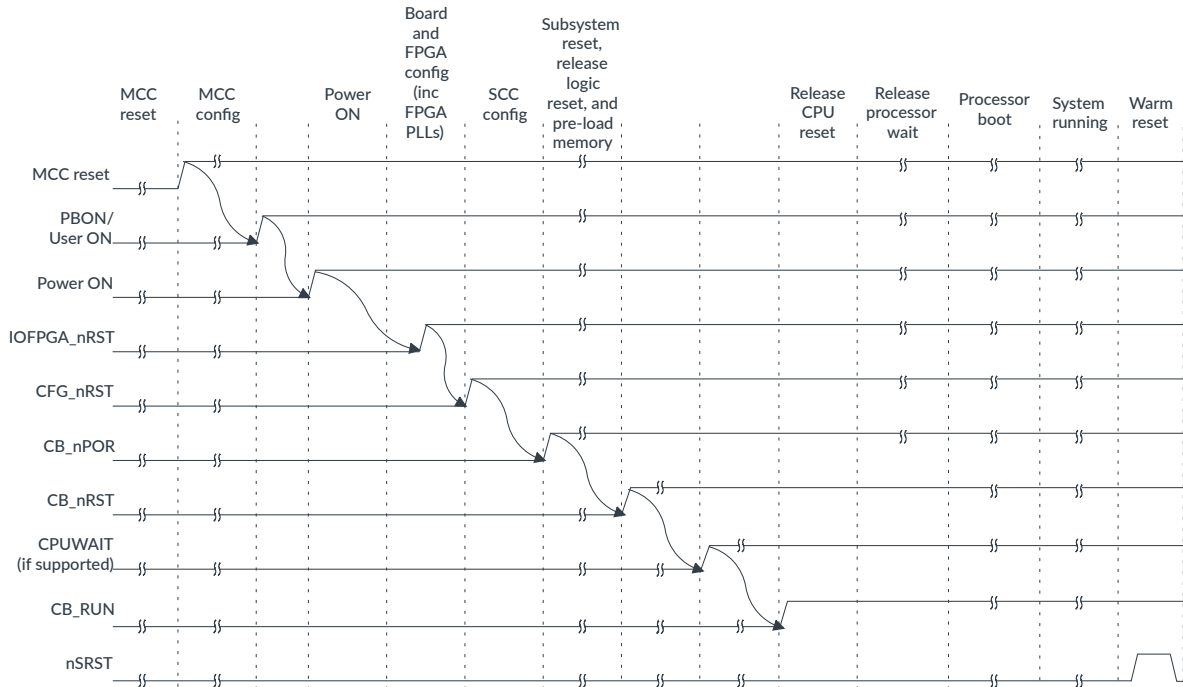
CPUWAIT (design dependent)

Core register that releases processor core or cores from reset, if supported in your FPGA design. This register is internal and not present at top level.

Reset sequence

The following figure shows the MPS4 board reset and powerup timing cycle including board configuration.

Figure 3-6: MPS4 board reset and configuration timing



3.3.3 Power

You supply power to the board from the mains supply using the on-board power connector and the power supply unit that Arm supplies with the board.

Arm supplies a 252W external power supply unit that converts mains power to 12V 21A DC and connects to the 12V 6-pin CP6 power connector on the board. The unit accepts mains power in the range 110V AC to 240V AC.

On-board regulators supply power to the board power domains and to the FPGA power domains.

The following on-board LEDs illuminate when power is applied:

12V OK

External 12V DC connected

PWR ON

On-board 5V supply asserted

SB_3V3 OK

On-board 3V3 supply asserted

Related information

[A.5 12V power connector](#) on page 103

[B. Available power for expansion boards](#) on page 105

3.4 Motherboard Configuration Controller

The *Motherboard Configuration Controller* (MCC) controls the MPS4 board and supports board configuration at powerup or reset. The MCC loads FPGA images into the FPGA, configuring the FPGA using the relevant FPGA implementation configuration files.

The MCC uses an Arm® Cortex®-M7 processor clocked at 216 MHz and includes these interfaces:

Serial Configuration Controller (SCC) interface

Reads and sets configuration values, typically used for prereset initialization of FPGA logic, see [3.4.1 Serial Configuration Controller interface](#) on page 33

Static Memory Controller (SMC) interface

Reads and writes memory areas with the FPGA logical design, and pre-loads user software images into memory, see [3.4.2 Static Memory Controller interface](#) on page 35

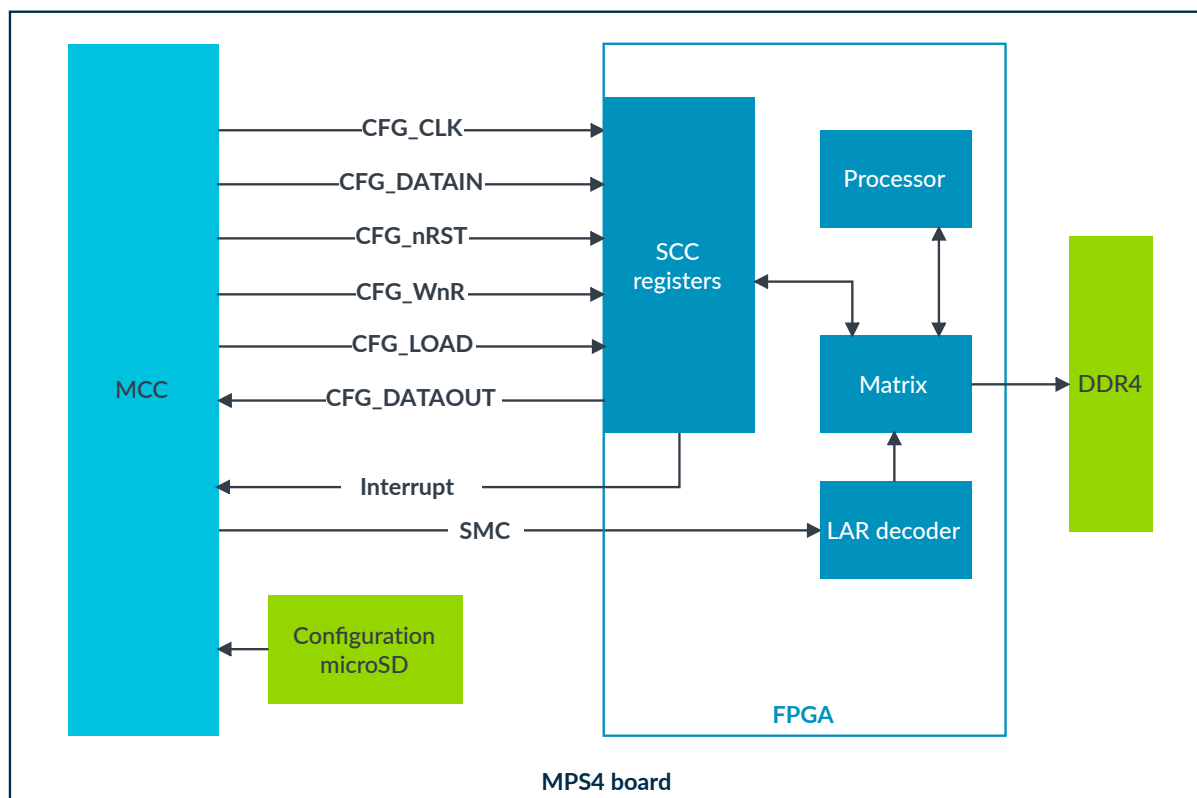
3.4.1 Serial Configuration Controller interface

Your FPGA implementation can include the *Serial Configuration Controller* (SCC) interface to set configuration values. It is typically used for pre-reset initialization (tie-offs) of FPGA logic.

After FPGA configuration, the *Motherboard Configuration Controller* (MCC) sets default values in the SCC registers, using values from the `board.txt` file in the configuration microSD card. For more information about the `board.txt` file, see [4.2.2.1 board.txt file](#) on page 64. The MCC configures the SCC through the SCC serial interface on the MCC.

The following figure shows the SCC interface connections to the FPGA.

Figure 3-7: Serial Configuration Controller interface



For the SCC registers, see the relevant documentation for the FPGA image package.

The MCC SCC interface operates at up to 10 MHz. Therefore, the FPGA image must be constrained to support up to 10 MHz on this interface.

The following figures show the read and write cycle timing of the SCC interface. The exact frequency and duty cycle of the SCC clock might vary depending on MCC use.

Figure 3-8: Serial Configuration Controller interface read cycle timing

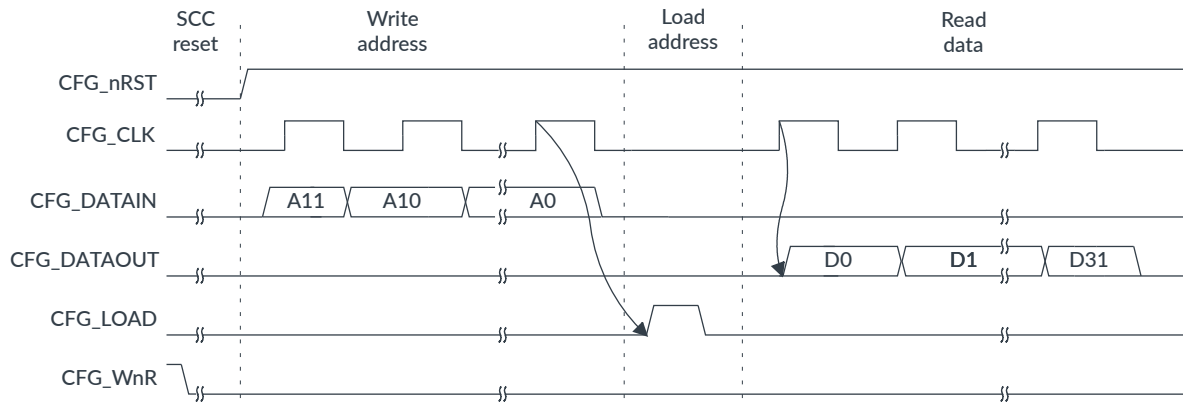
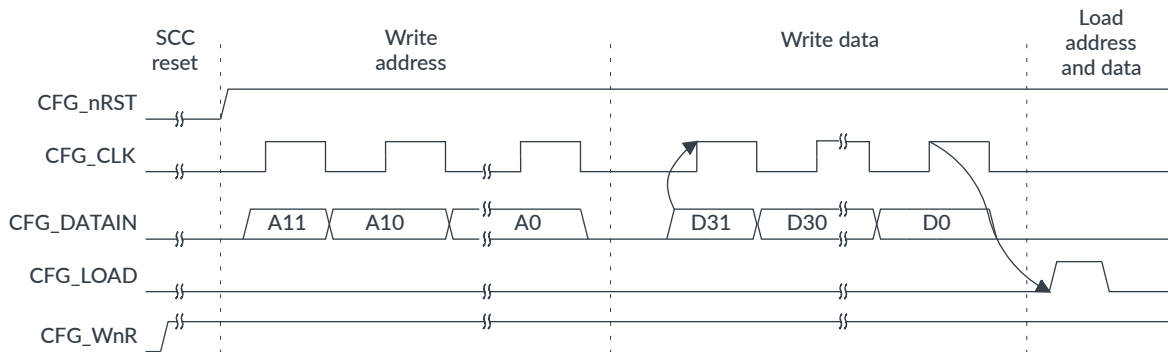


Figure 3-9: Serial Configuration Controller interface write cycle timing



Serial Configuration Controller interface not implemented

If your FPGA implementation does not implement the SCC, set the parameter `FPGA_scc` to `FALSE` in the board `config.txt` file. See [4.2.1 config.txt board configuration file](#) on page 63.

3.4.2 Static Memory Controller interface

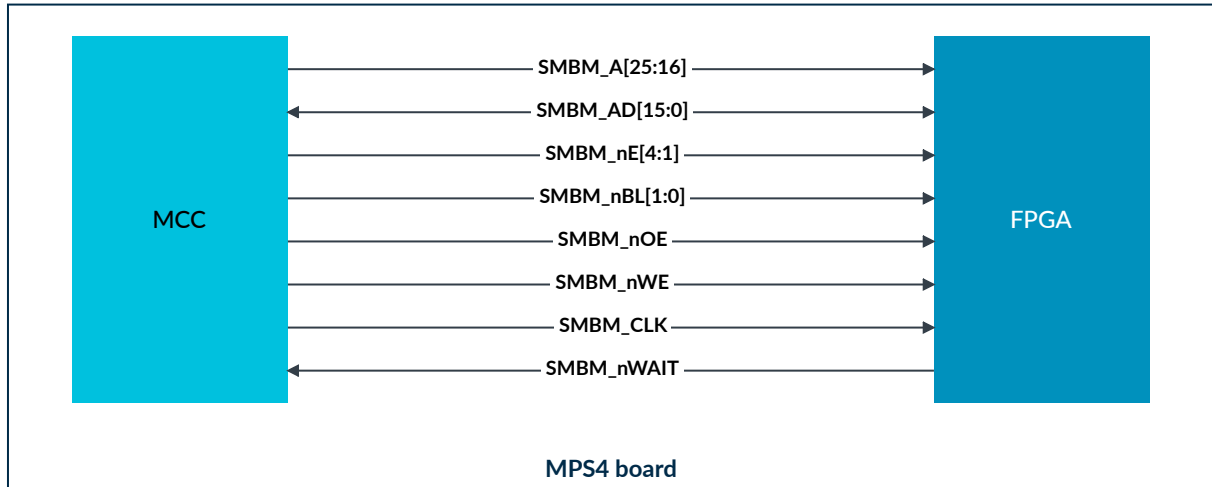
The *Static Memory Controller* (SMC) interface in the *Motherboard Configuration Controller* (MCC) supports read and write transactions that enable communication with the internal system bus of the FPGA.

If enabled, your FPGA implementation must convert these read and write transactions to the type of transactions that are used in your FPGA implementation, usually AHB-type or AXI-type transactions.

The SMC interface uses a block of registers in the FPGA that supports 40-bit address, 32-bit data *Long Address Range* (LAR) with multiple address regions.

The following figure shows the SMC interface. The SMBM_nBL[1:0] signal is reserved.

Figure 3-10: SMC interface



The SMC interface enables the MCC to access the peripheral space of the FPGA system. Typical uses are to:

- Preload boot images
- Read and write to System registers
- Configure peripherals
- Run memory transactions during runtime

Implementing the SMC interface

Chip selects

The SMBM_nE[4:1] signals operate as active-LOW chip selects:

- SMBM_nE[1] is used as a 64 MB window, which is offset by the LAR register.
- SMBM_nE[2] sets the SMC LAR register to allow support for up to 40-bit addressing and 16 chip selects (ports).

Address and data transfer out of the SMC interface

The MCC provides 25 of the 40 address bits that the FPGA implementation uses for AHB-type transactions. The FPGA SMC LAR register generates the 14 top bits. The MCC sets this register using the SMBM_nE[2] access window. Address bit[0] is always set to 0 by the FPGA implementation.

During the address phase, the 25 address bits are shared between the SMBM_D[15:0] signal and the SMBM_A[24:16] signal:

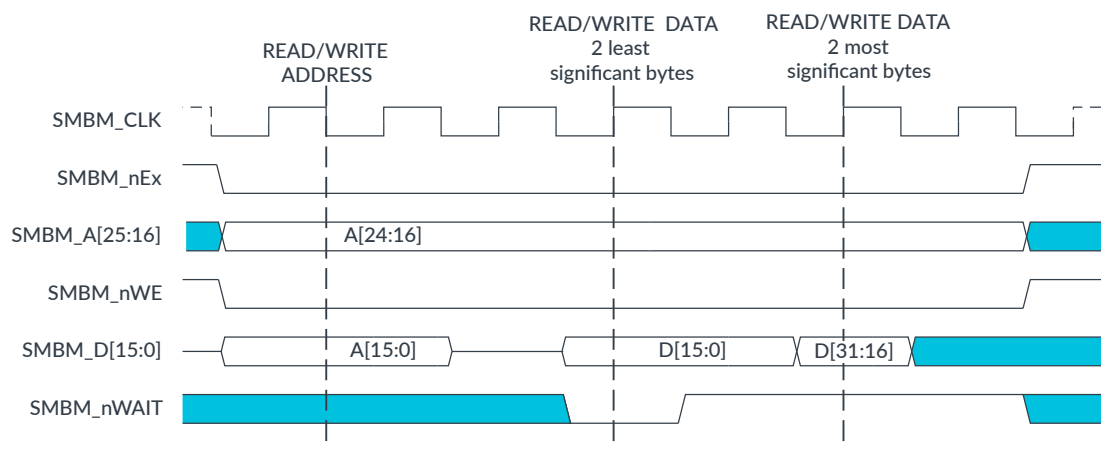
- SMBM_D[15:0] carries the 16 least significant address bits of the interface.
- SMBM_A[24:16] carries the 9 most significant address bits of the interface.

The SMBM_A[25] signal is reserved.

During the data transfer phase, the SMBM_D[15:0] signal carries the 4 data bytes in 2 stages. It carries the 2 least significant bytes, and then the 2 most significant data bytes.

The following figure shows an SMC interface address and data transfer.

Figure 3-11: SMC interface address and data transfer timing



The SMC interface supports only 32-bit wide data read and write transfers. 8-bit or 16-bit data transfers are not supported.

Forming the 40-bit address and 16 chip selects in the FPGA

The FPGA implementation must form the 40-bit address, for AHB-type transfers inside the FPGA, from the following bits:

- The least significant bit generated by the FPGA implementation, which is always 0b0



The SMC interface supports only 4-byte-aligned address mode transactions. To support 4-byte address mode, the 2 LSBs generated by the FPGA implementation must be 0b00 and the 2 LSBs of the SMC interface are always 0b00.

- 25 address bits from the MCC
- 14 upper address user bits, generated by the FPGA implementation (LAR SMC register value)

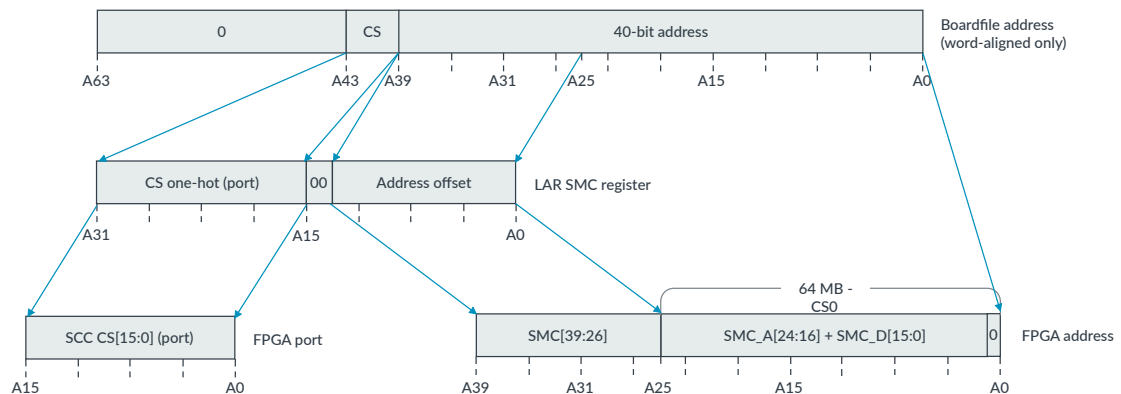
- 16 chip-select user bits (LAR SMC register value)

The MCC can address up to 16 targets with a 40-bit address window, if supported by your FPGA implementation.

The 14 address bits generated by the FPGA implementation, and if necessary, the chip-select bits, define what parts of the user memory space are accessed.

The following figure shows the formation of the AHB 40-bit address in the FPGA.

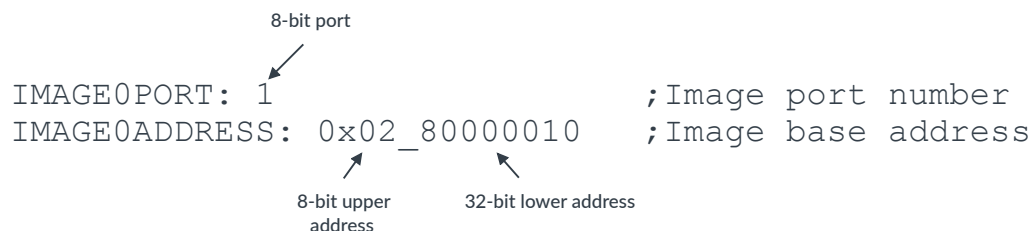
Figure 3-12: Formation of 40-bit address in FPGA



images.txt configuration file example

You can set up the `images.txt` configuration file to load software images or binary images. The parameters `IMAGEEXPORT` and `IMAGEADDRESS` in the `images.txt` configuration file define the port number and base address in the MCC memory. For example, setting `IMAGEEXPORT` to 1 and `IMAGEADDRESS` to `0x02_80000010` selects a port address of 1 and a base address of `0x02_80000010` in the FPGA user memory. The following figure shows this example for image number 0.

Figure 3-13: Example IMAGE0PORT and IMAGE0ADDRESS parameters



To learn more about the `images.txt` file, see [4.2.2.3 images.txt file](#) on page 67.

SMC interface not implemented

If your FPGA implementation does not implement the MCC-SMC interface, you must set the parameter `FPGA_SMB` to `FALSE` in the board `config.txt` file. See [4.2.1 config.txt board configuration file](#) on page 63.

3.5 Display, camera, and audio

The MPS4 board provides a flexible set of display, camera, and audio interfaces to enable you to develop a wide range of applications.

The MPS4 board includes:

- RGB to HDMI video (1080p) transmitter
- 5.1 surround-sound I²S audio codec interface
- LCD interface that uses a Raspberry Pi HAT connector
- 1 DSI port, supporting up to 2 channels with up to 1.25 Gbps per channel
- 2 CSI ports, supporting up to 2 channels with up to 1.25 Gbps per channel
- 1 CSI port, supporting up to 4 channels with up to 1.25 Gbps per channel

3.5.1 HDMI display interface

The HDMI transmitter and HDMI connector on the MPS4 board enable you to implement an HD video interface.

You can include the HDMI controller IP in the FPGA design. It is a framebuffer device that can display up to 1920 × 1080p resolution at 60 fps.

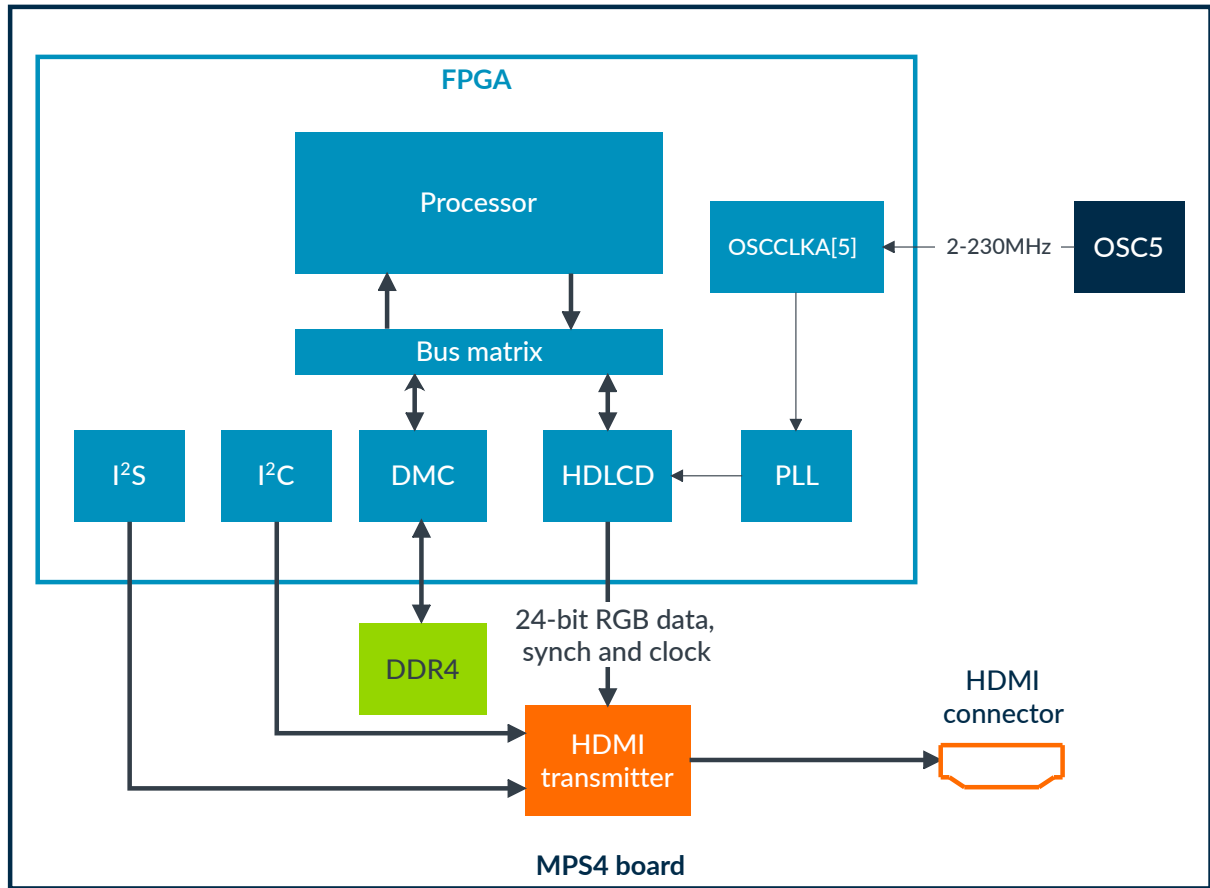


Support for higher resolutions up to 1080p depends on the timing performance of your FPGA image.

The HDMI transmitter supports 4-lane I²S audio from the FPGA.

The following figure shows an HDMI example MPS4 board design.

Figure 3-14: MPS4 board HDMI example design



OSC5 is powered from the FPGA FMC bank, which must be powered to use OSC5. See the `FMC_FORCE` parameter in `config.txt`.

Related information

[A.3.1 HDMI connector](#) on page 95

3.5.2 Audio codec interface

An AACI audio codec on the MPS4 board provides a 5.1 surround-sound audio interface with 6 channels in and 6 channels out.

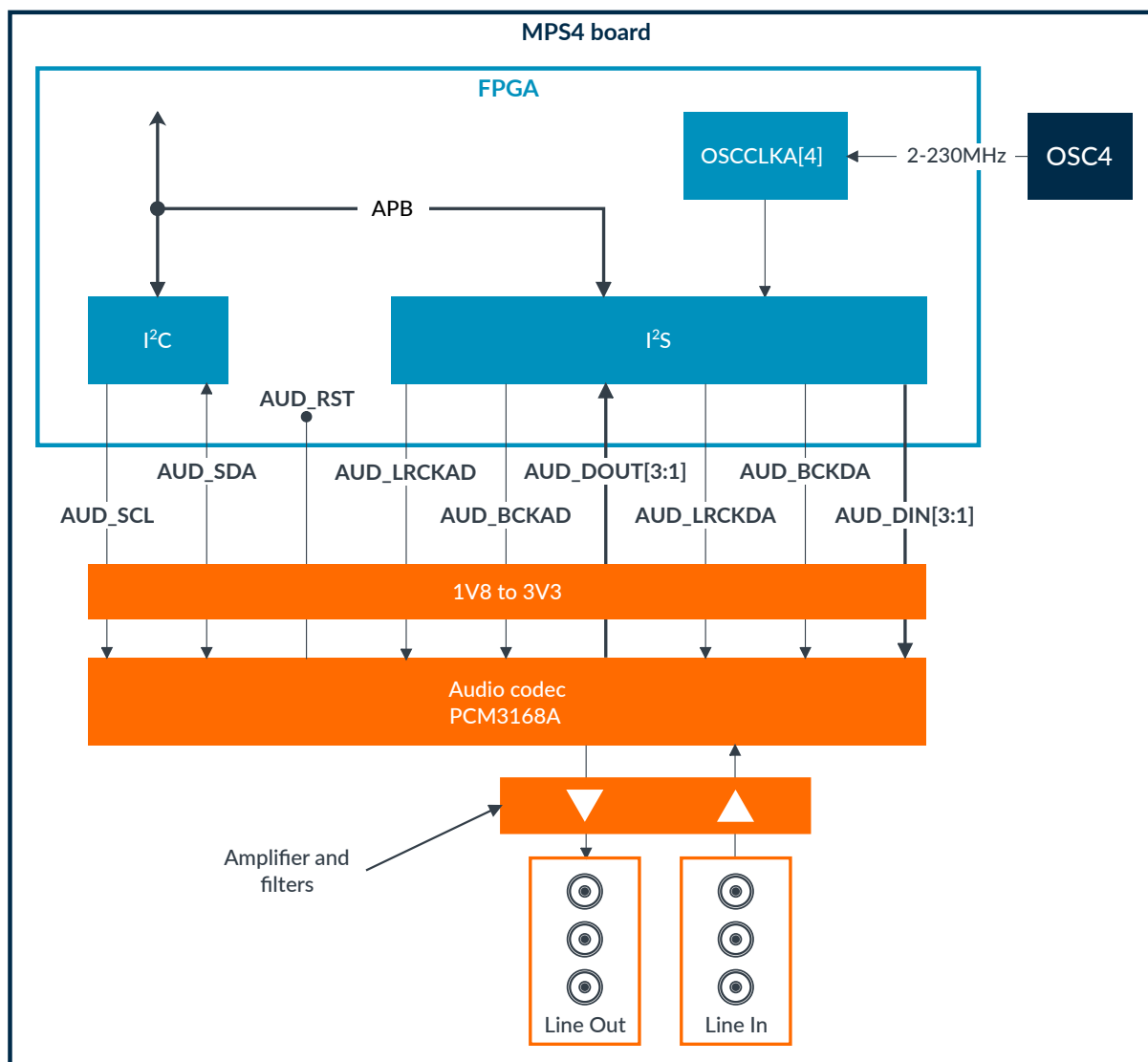
The FPGA configures the codec over I²C and has a 3-lane I²S audio interface.

The interface supports audio sample rates up to 96 kHz. The on-board amplifiers and filters provide line-level input and outputs.

The audio codec drives 6 stereo jack connectors on the MPS4 board.

The following figure shows an audio codec interface example design.

Figure 3-15: MPS4 board audio codec interface example design



Related information

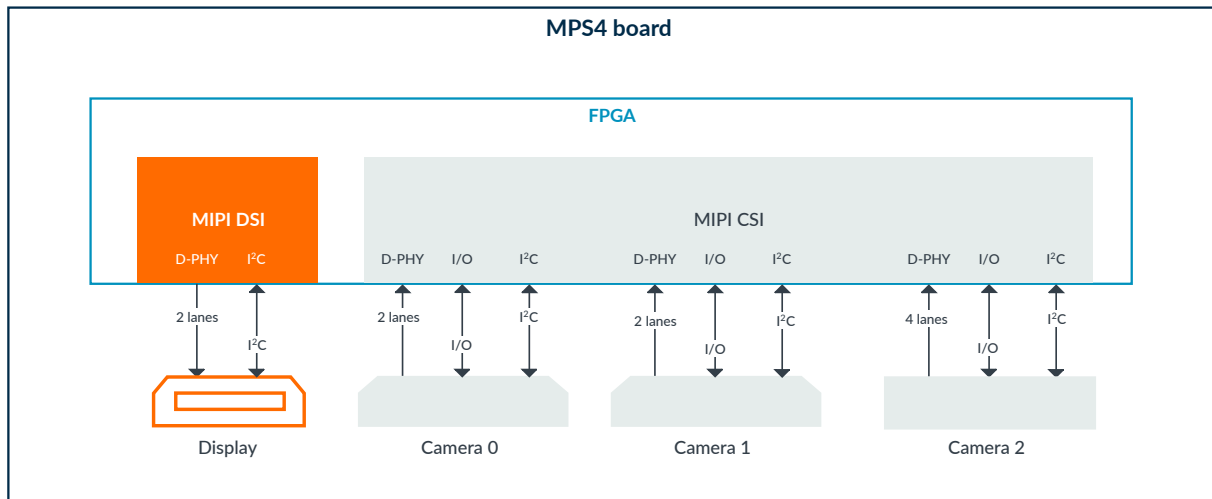
[A.3.2 Audio connectors, stacked stereo jacks](#) on page 95

3.5.3 Display Serial Interface

The MPS4 board provides a 2-lane 15-way MIPI *Display Serial Interface* (DSI) port (DSI-1), which is a more power-efficient display option than HDMI. The DSI D-PHY connector supports high-speed DSI display connections of up to 1.25 Gbps per lane.

The following figure shows an overview of the MPS4 board DSI.

Figure 3-16: MPS4 board DSI overview



The DSI connector has the part number TE 1-1734248-5. The connector is pin-compatible with Raspberry Pi DSI displays.

Related information

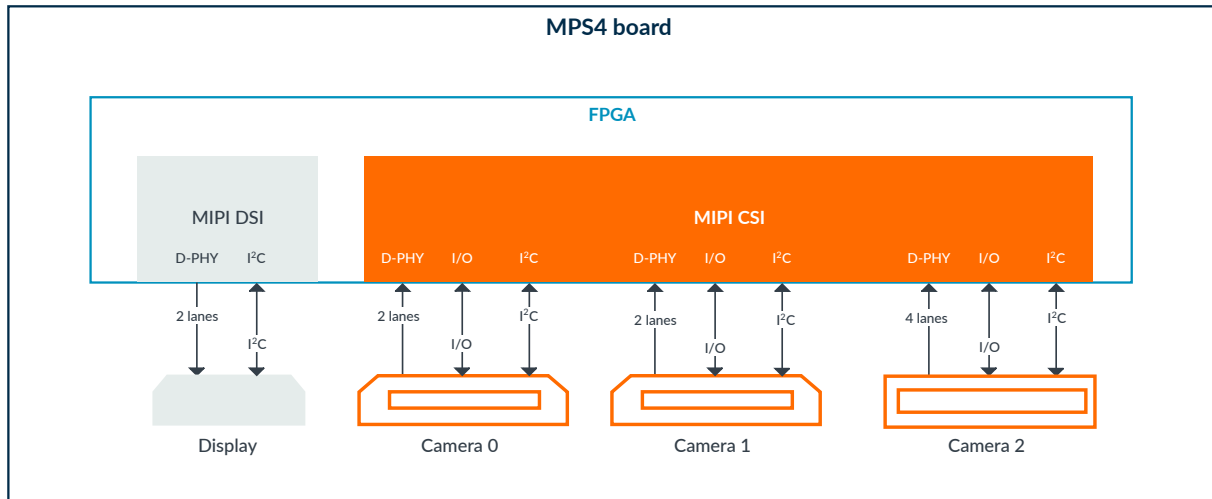
[A.3.3 MIPI DSI connector](#) on page 96

3.5.4 Camera Serial Interfaces

The MPS4 board provides 3 MIPI *Camera Serial Interface* (CSI) ports (CSI-1a), which support higher resolutions than SPI. The CSI D-PHY connectors support high-speed CSI camera connections of up to 1.25 Gbps per lane.

The following figure shows an overview of the MPS4 board CSI.

Figure 3-17: MPS4 board CSI overview



The MPS4 board has 3 CSI connectors:

Camera 0

Camera 1

- 2-lane MIPI D-PHY differential connectors for lower-bandwidth cameras
- Both connectors are pin-compatible with Raspberry Pi CSI connectors
- 15-pin CSI connectors have the part number TE 1-1734248-5

Camera 2

- FCS8 4-lane MIPI D-PHY differential connector for higher-bandwidth cameras
- 30-pin CSI connector has the part number Samtec FCS8-30-01-L-S-A-TR

Related information

[A.3.4 MIPI CSI connectors](#) on page 97

3.6 I/O connectivity

The MPS4 board provides a flexible set of interfaces and components to support Arduino Shields, board-to-board connections, trace, debug, and other I/O connectivity. This I/O connectivity enables you to add various sensors and peripherals to help develop and test your custom design.

3.6.1 Arduino Shield, Pmod, and HAT interfaces

The MPS4 board supports peripheral development with 2 Arduino Shield interfaces, 4 *Peripheral module* (Pmod) connectors (Type 2A/3/6 support), and a Raspberry Pi *Hardware Attached on Top* (HAT) interface.

The Shield, Pmod, and HAT interfaces share signals, including the digital I/O. Each Shield interface has 16 digital I/O, the 2 Pmod interfaces each have 8 digital I/O, and the HAT interface has 28 I/O. This means that you cannot use all of the Shield, Pmod, and HAT interfaces simultaneously. The following combinations are supported (Pmod 0/1 and Pmod 2/3 are supported as either 1 or 2 interfaces):



- Shield 0 and Shield 1
- Shield 0 and Pmod 2/3
- Shield 1 and Pmod 0/1
- Pmod 0/1 and Pmod 2/3
- HAT

To learn more about how the Shield, Pmod, and HAT interfaces share the digital I/O signals, see [A.2.1 Arduino Shield, Pmod, and HAT shared SHx_I/Ox signals](#) on page 82.

The following figure shows the 2 Shield interfaces, the HAT interface, and the 4 Pmod connectors. In this example, the FPGA design implements an SPI 12-bit ADC and supports 1V8 I/O, 3V3 I/O, and HAT connections.

The diagram illustrates the internal architecture of the MPS4 board. On the left, the **FPGA** is connected to an **SPI controller** and **System registers**. The SPI controller interfaces with a **12-bit ADC AD7490**. The ADC's output is sent to the FPGA via **SH0_IO[15:0]** and **SH0_EN[15:0]** signals, which pass through a **Level shifter CPLD 1V8/3V3**. The ADC also receives control signals (**ADC_CS**, **ADC_CLK**, **ADC_DIN**, **ADC_DOUT**) from the SPI controller. Power supply connections include **1V8**, **3V3**, and **5V**, with two **1V8/3V3 link** blocks managing voltage levels. The board features multiple interface headers on the right, corresponding to each connector: **Pmod 0**, **Pmod 1**, **Shield 0**, **HAT**, **Shield 1**, **Pmod 2**, and **Pmod 3**. These headers provide access to various pins including **IOREF0**, **IOREF1**, **VINO**, **AREF0**, **AD0[5:0]**, **Reset0**, **Reset1**, **SH1_IO[15:0]**, and **AD1[5:0]**.

Copyright © 2022, 2024 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

All interface pins are connected to the FPGA in the same way. Therefore, you can configure the pins for different standards, such as SPI, I²C, and UART, depending on your FPGA design and the requirements of each interface.



The Arduino Shield, Pmod, and HAT interfaces use the same I/O reference voltage and power select links that select digital I/O references and power inputs.

For more information about the user-links, see [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47.

Arduino Shield interfaces

Each Shield interface has 16 I/O and supports the choice of 1V8 or 3V3 I/O, independently selectable for each Shield using 2 user-links.

The outputs from the digital I/O user-links act as the digital I/O voltage references for the Shields.

A 12-bit ADC supports six analog channels on each Shield up to 5V.

Pmod interfaces

The 2 Pmod interfaces are an alternative to the Shield interfaces:

- Pmod connectors 0 and 1 form Pmod interface Pmod 0/1:
 - You can use Pmod connectors 0 and 1 independently or combine them to make a dual connector interface
 - Pmod 0/1 use the same 16 I/O pins as Shield 0
- Pmod connectors 2 and 3 form Pmod interface Pmod 2/3:
 - You can use Pmod connectors 2 and 3 independently or combine them to make a dual connector interface
 - Pmod 2/3 use the same 16 I/O pins as Shield 1

Each Pmod expansion connector has 8 digital I/O and supports the choice of 1V8 or 3V3 digital I/O, although typically Pmod is used at 3V3.

The outputs from the digital I/O user-links also act as the digital I/O voltage references for the Pmod expansion boards.

The locations of the Pmod connectors support the use of a dual-connector board, if required.

The Pmod interfaces do not support analog I/O.

HAT interfaces

You can use the Raspberry Pi HAT slot as a generic interface for plug-on boards, such as for LCD and touch screen interfaces. The HAT slot supports 28 general-purpose 1V8 or 3V3 I/O, although typically only 3V3 is used for HAT peripherals.

Support for HAT peripherals depends on what your FPGA design supports.

Selecting I/O voltage levels and reference voltages for expansion connectors

You can select I/O voltage levels and reference voltages for expansion connectors using 2 user-links on the board:

- A user-link selects 1V8 or 3V3 as the I/O voltage level and reference voltage for the Pmod 0/1, Shield 0 and the SH0_ pins of the HAT connector.
- A user-link selects 1V8 or 3V3 as the I/O voltage level and reference voltage for the Pmod 2/3, Shield 1 and the SH1_ pins of the HAT connector.

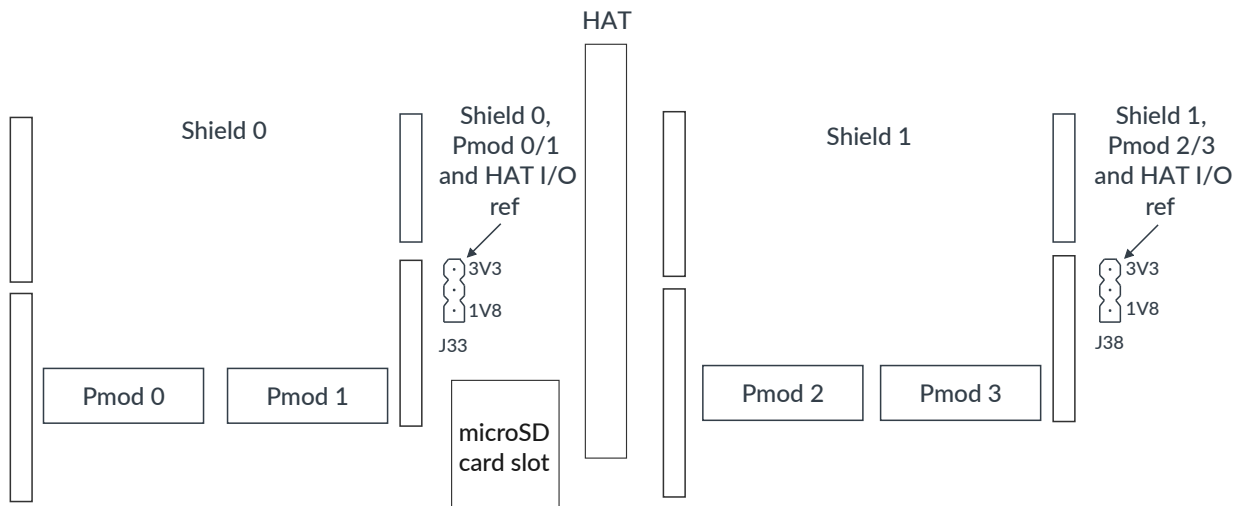
By default, the voltage is set to 3V3.



Because both user-links select the I/O and reference voltages for pins on the HAT connector, you must set them both to the same voltage if the HAT connector is in use.

The following figure shows the user-links that select 1V8 or 3V3 I/O references and voltage levels for the Shield, Pmod, and HAT interfaces.

Figure 3-19: Shield, Pmod, and HAT power and I/O reference voltage user-links



The maximum currents available at the power and reference pins are:

1V8/IOREF

2A maximum available for both Shields, all Pmod interfaces, and the HAT interface.



3V3/IOREF

2A maximum available for both Shields, all Pmod interfaces, and the HAT interface.

Related information

[1.2 MPS4 board layout](#) on page 7

[A.2.2 Arduino Shield connectors](#) on page 83

[A.2.3 Pmod connectors](#) on page 87

[A.2.4 Raspberry Pi HAT and board farm connectors](#) on page 89

3.6.2 FMC+ HPC interface

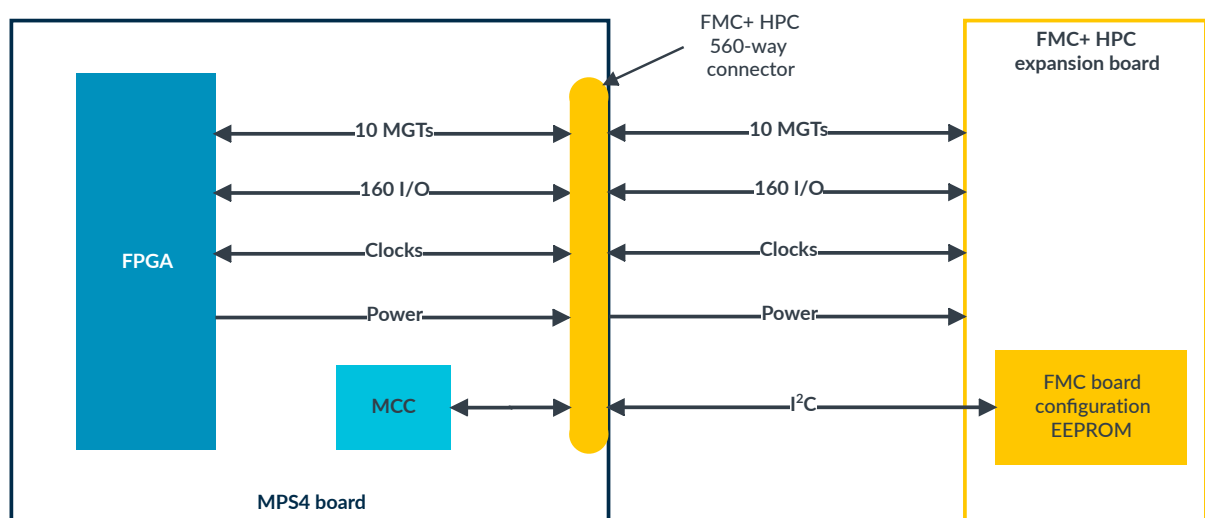
The MPS4 board supports high speed, high pin count expansion using the *FPGA Mezzanine Card Plus* (FMC+) standard. FMC+ is compatible with the *FPGA Mezzanine Card* (FMC) standard.

The MPS4 board uses the FMC+ *High Pin Count* (HPC) variant and provides:

- 560-way Samtec SEARAY connector with 400 pins connected, supporting either FMC+ or FMC expansion boards
- 160-way single-ended general-purpose I/O or 80-way differential general-purpose I/O. Utilizes AMD SelectIO high-performance banks.
- 10 high-speed differential transceivers, *MultiGigabit Transceivers* (MGTs). FMC+ provides 24 MGTs, however, only 10 MGTs are supported. This link has been tested up to 10 Gbps.
- 2 high-speed differential MGT clocks
- 4 differential clocks
- Power to attached FMC+ and FMC boards

The following figure shows the MPS4 board FMC+ HPC interface.

Figure 3-20: MPS4 board FMC+ HPC interface



The FMC board configuration EEPROM is part of the FMC standard.

FPGA-FMC+ pin connectivity

You can download the FPGA-FMC+ pinout as a .xlsx spreadsheet at <https://developer.arm.com/mps4>. For more information on the FPGA packages, see [3.1.2 FPGA image packages](#) on page 24.

FMC+ board sizes

The MPS4 board supports FMC and FMC+ single-width modules that are 69mm wide.



To avoid damage when you insert a module with an FMC connector into the MPS4 board using an FMC+ connector, ensure that both connectors are correctly aligned before you push them together. Using the 2 side locating holes can make this easier.

Related information

[A.2.5 FMC+ HPC connector](#) on page 92

3.6.3 On-board user components

The MPS4 board provides these on-board user-programmable components:

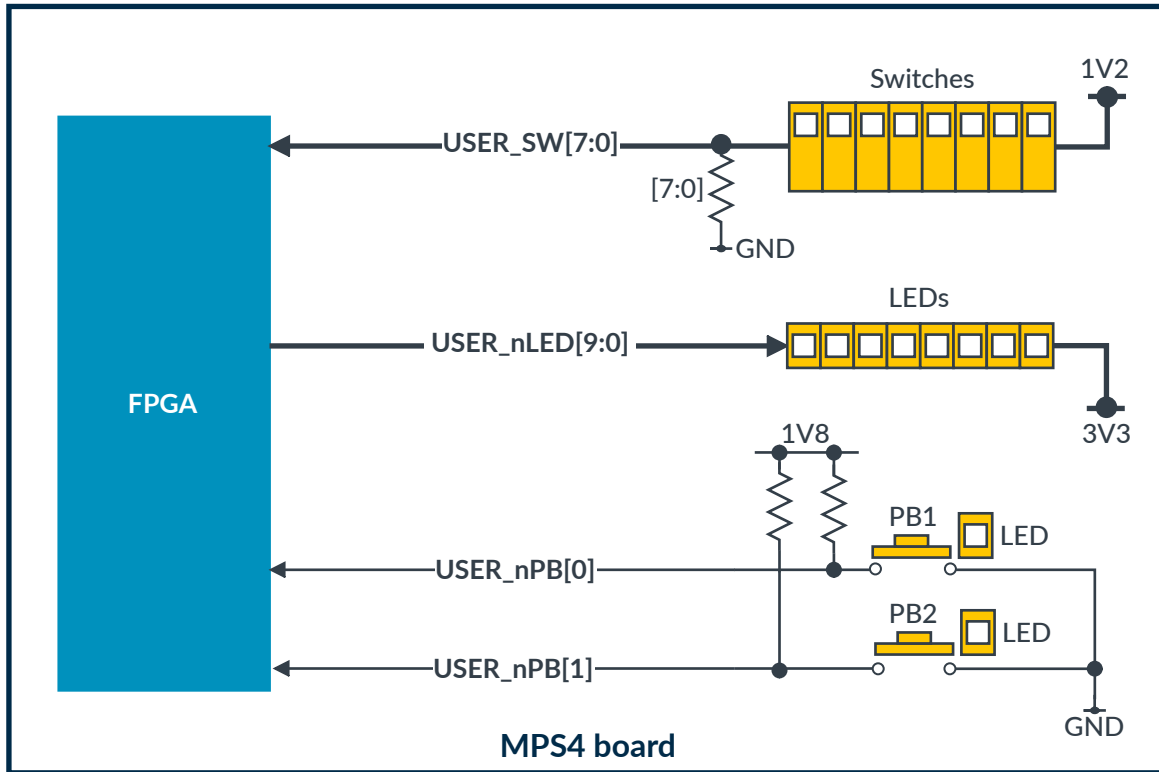
Table 3-4: On-board user components

Components	Polarity
8 user LEDs	Active-LOW
8 user switches	Active-HIGH
2 user push buttons with LEDs	Active-LOW

The LEDs, switches, and push buttons connect directly to the FPGA, meaning you can use them for debug.

The following figure shows the user components on the MPS4 board.

Figure 3-21: MPS4 board user components



3.6.4 Board-to-board interface and QSFP interface

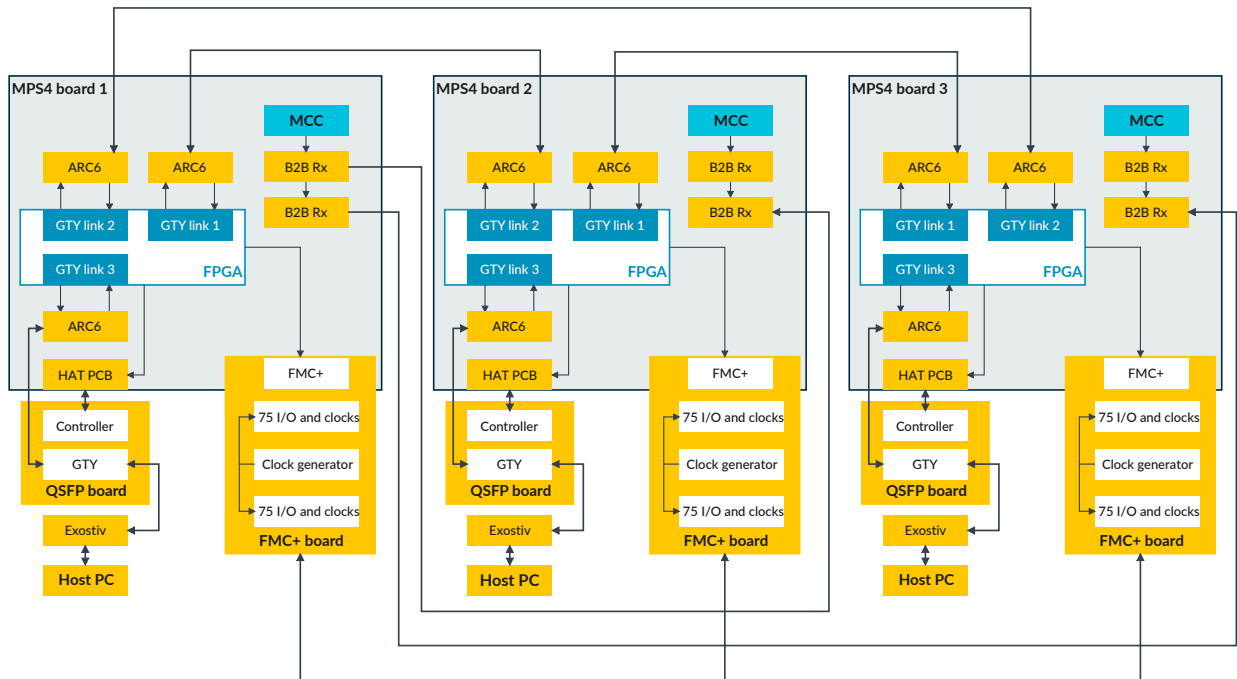
The MPS4 board supports high-speed board-to-board connections through a wide multigigabit link supporting 36 GTY transceivers over 3 Samtec AcceleRate ARC6 24-lane connectors.

The interface supports up to 25 Gbps per lane for board-to-board links (J48 and J49) and up to 12.5 Gbps per lane for the QSFP interface (J50).

When only 2 connectors are used for board-to-board links, the third connector can be used as a QSFP interface. The QSFP interface supports QSFP peripherals and debug tools, such as EXOSTIV. Learn more about EXOSTIV at <https://www.exostivlabs.com>.

The following figure shows an example board-to-board setup linking 3 MPS4 boards together and using EXOSTIV connectivity. EXOSTIV connectivity requires an external QSFP board to link the MPS4 board to the EXOSTIV debug interface. A custom board that connects to the MPS4 board HAT interface provides power and control for the QSFP board.

Figure 3-22: MPS4 board-to-board example



Related information

[A.2.6 Samtec AcceleRate ARC6 connectors](#) on page 93

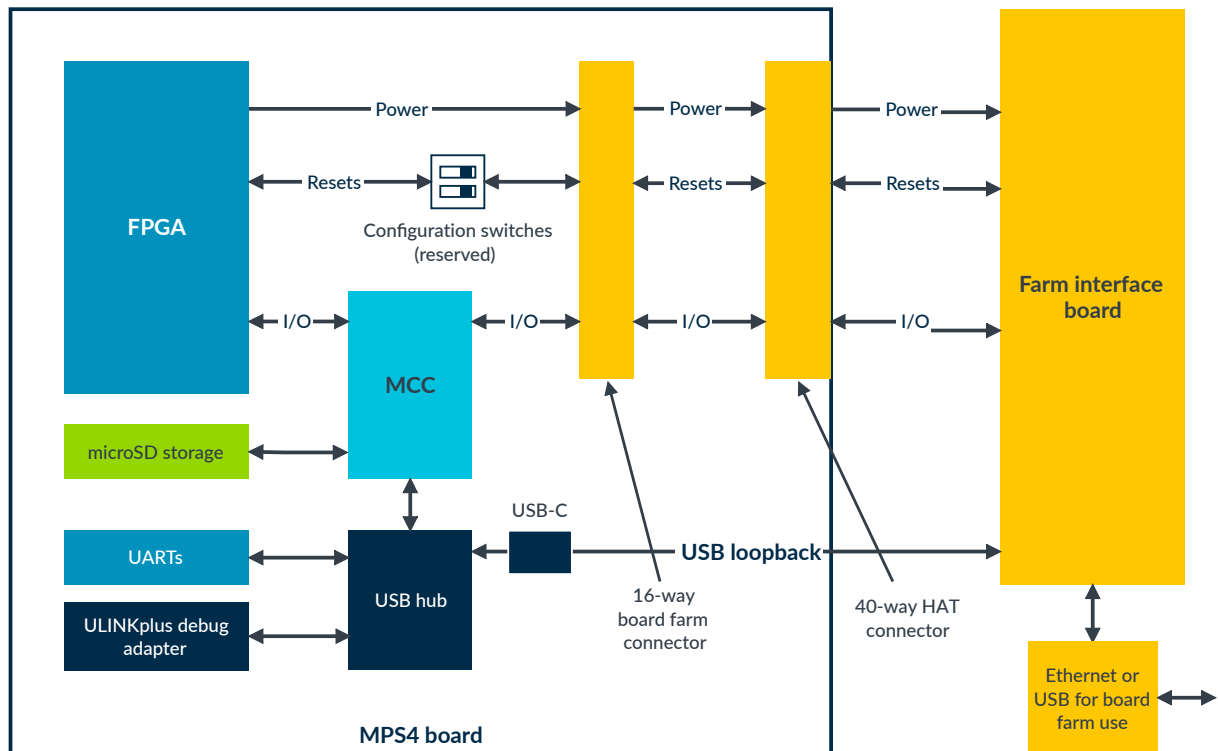
3.6.5 Remote board farm interface

The MPS4 board supports remote USB and Ethernet connectivity through a board farm interface, compatible with a Raspberry Pi 4. The board farm interface allows MPS4 boards to be accessed remotely rather than on your desk. This allows you to share boards for different projects.

You can connect your farm interface board to the MPS4 board using the USB-C connector. This USB loopback connection provides access to the file system and debug tools. The farm interface can also control and read the status of resets of the MPS4 board.

The following diagram shows an example of the MPS4 board farm support.

Figure 3-23: MPS4 board farm support example



This example shows a 16-way Raspberry Pi board farm connector on the MPS4 board that gives access to power, board resets, the MCC, and FPGA I/O. This pin header connects the farm interface to the Raspberry Pi HAT 40-pin I/O connector and powers the farm interface board with up to 5V 3A. To see how the 16-way board farm connector maps to the 40-way HAT connector, see [A.2.4 Raspberry Pi HAT and board farm connectors](#) on page 89.

For information on how to configure remote farm access, see [4.4 Configuring remote farm access](#) on page 70.

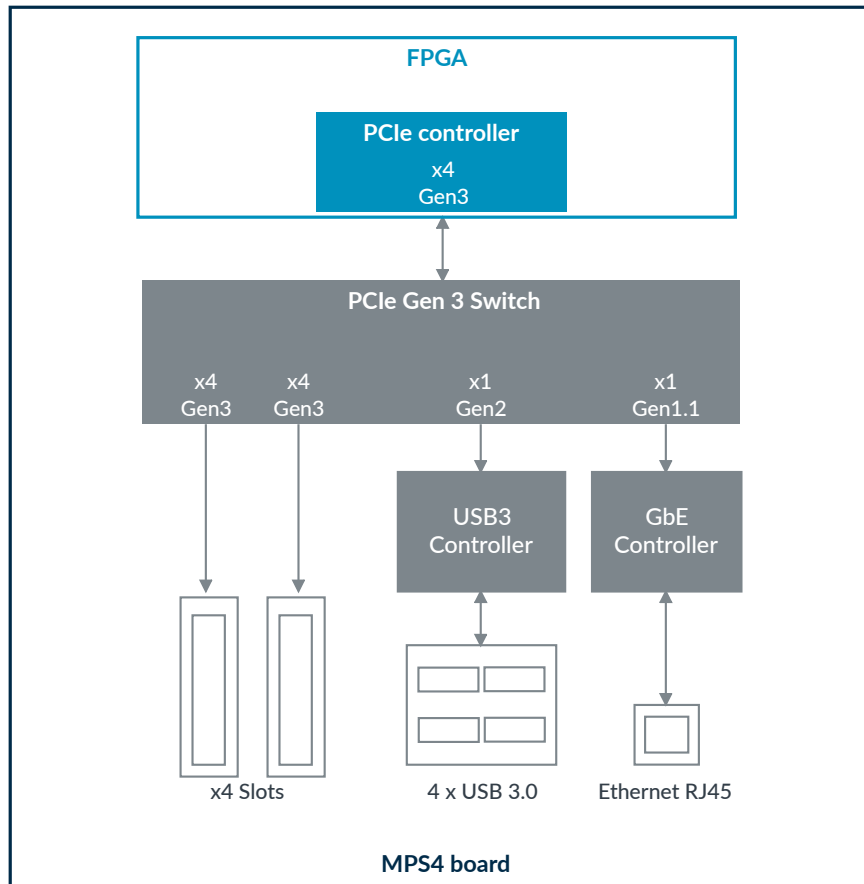
3.7 PCI Express systems

The MPS4 board provides PCI Express (PCIe) interfaces to support high-bandwidth peripherals. On-board PCIe peripherals include a USB 3.0 host controller and a *Gigabit Ethernet* (GbE) controller.

The MPS4 board supports PCIe plug-in cards through 2 PCIe x4 Gen 3 sockets, which you can use for M.2 mass storage or extra peripherals. The MPS4 board PCIe Switch, USB 3.0, and GbE controllers provide high-speed peripheral interfaces to support Linux-based designs.

The following diagram shows the MPS4 board PCIe systems.

Figure 3-24: MPS4 board PCIe systems overview



The MPS4 board provides the following systems and PCIe components.

PCIe card slots

The MPS4 board provides 2 PCIe x4 card slots. You can use these slots for mass storage devices, such as M.2 storage expansion cards, which can plug in using an M.2-to-PCIe card adapter. Each slot supports a maximum 25 W output load. The total system power must not exceed the power supply rating, see [PCIe power](#) on page 105.

For more information on the PCIe card slots, see [A.4.1 PCI Express connectors](#) on page 100.

PCIe controller

The MPS4 board implements the AMD DMA/Bridge Subsystem for PCI Express v4.1 IP in a root complex configuration. This controller enables software configuration and control of the PCIe link to the PCIe Switch on the MPS4 board.

PCIe Gen 3 Switch

All upstream and downstream traffic between the PCIe slots, USB 3.0 controller, GbE controller, and the MPS4 board goes through the PCIe Gen 3 Switch:

- Broadcom ExpressLane PEX8724-CA device

- 6 ports and 24 lanes

USB 3.0 controller

The PCIe Switch connects to a USB 3.0 host controller over an x1 PCIe Gen 2 link. The USB 3.0 host controller drives 4 USB 3.0 ports on the back panel:

- Texas Instruments TUSB7340IRKMT device with a Gen 2 link to the PCIe Switch
- Operates at USB 3.0 (5 Gbps), and is compatible with USB 2.0 (480 Mbps)

GbE controller

The PCIe Switch connects over a x1 Gen 1.1 link to the GbE controller:

- Microchip LAN7430 device
- Drives an RJ45 GbE port
- Provides a 10/100/1000M connection to the GbE port

Related information

[A.4.1 PCI Express connectors](#) on page 100

[A.4.2 Ethernet and USB 3.0 connectors](#) on page 103

3.8 Interrupts

Interrupt signals from peripherals on the MPS4 board connect to input pins on the FPGA. The FPGA provides 2 interrupt signals to the *Motherboard Configuration Controller* (MCC) and 1 for the USB FIFO.

The FPGA image that you implement connects the peripheral interrupts to systems in the FPGA.

The following table shows the interrupt signals that connect to input pins on the FPGA and the interrupt signals that the FPGA provides. The PB_IRQ signal is a general-purpose interrupt.

Table 3-5: Interrupts

Interrupt signal	Source	Destination
DVI_INT	HDMI	FPGA
PEX_nINTA	PCIe Switch	FPGA
PB_IRQ	MCC (Reserved)	FPGA
DDR_nEVENT	DDR	FPGA
USER_nPB[1:0]	User push buttons	FPGA
PCIE_nWAKE	PCIe Switch	FPGA
FIFO_nWAKE	FPGA	USB FIFO
WDOG_RREQ	FPGA	MCC
IOFPGA_SYSWDT	FPGA	MCC

3.9 Debug

The MPS4 board provides several methods of performing debug. You can debug using Arm® CoreSight™ on several connectors. You can debug over USB using the USB-C connector.

For FPGA and RTL debug, you can use the ILA connector with ChipScope or similar tools. For more information on the ILA connector, see [A.1.4 14-pin F-JTAG ILA debug connector](#) on page 80.

The MPS4 board also provides a QSFP interface (requiring a separate plug-in board available from Arm), which uses one of the board-to-board ARC6 connectors, if available. You can use the QSFP interface to connect to QSFP peripherals and debug tools, including EXOSTIV. Contact Arm for more information about using the QSFP interface for debug. For more information about the QSFP interface generally, see [3.6.4 Board-to-board interface and QSFP interface](#) on page 50.

Related information

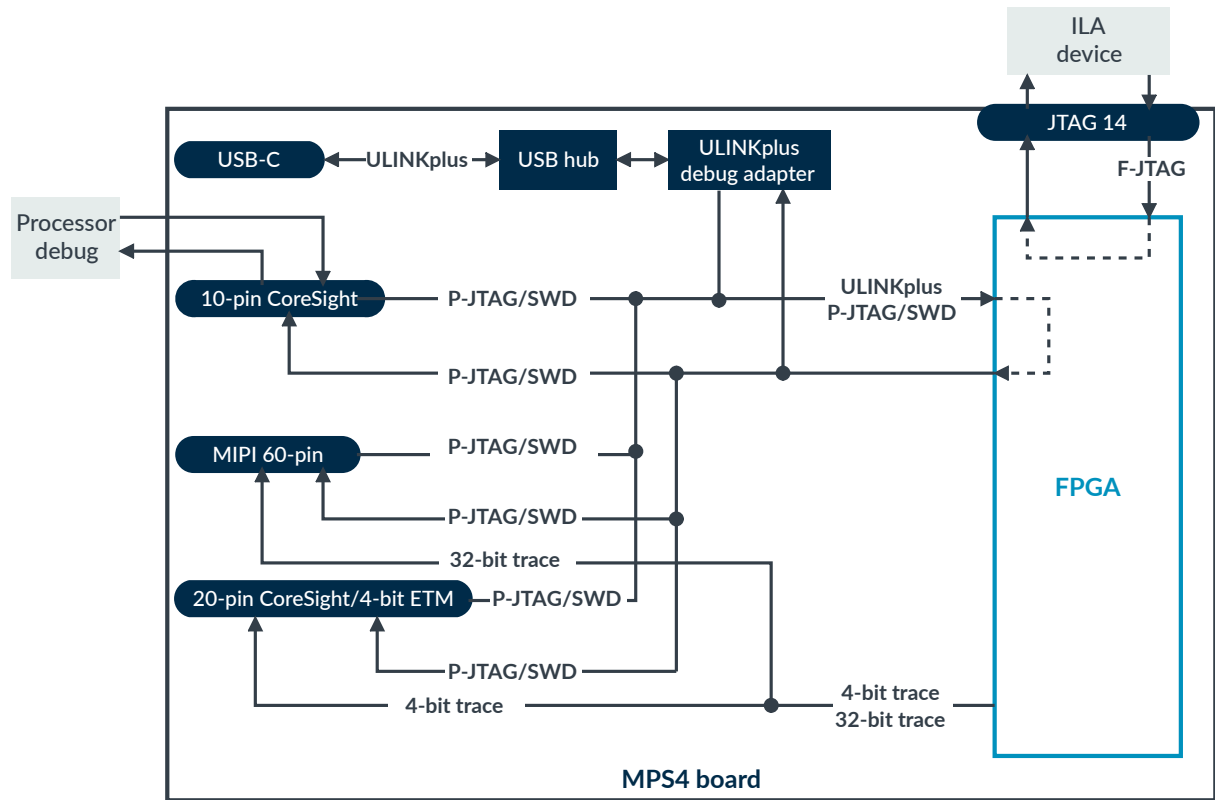
[A.1 Debug connectors](#) on page 76

3.9.1 CoreSight™ debug and trace

The MPS4 board supports connectivity from the FPGA to Arm® CoreSight™ debug connectors. If necessary for your FPGA design, you can use these connectors for CoreSight™ debug and trace.

The following figure shows the MPS4 board debug and trace system.

Figure 3-25: MPS4 board CoreSight™ debug and trace



The following table shows the components you can use for CoreSight™ debug and trace.

Table 3-6: Components that support CoreSight™ debug and trace

Connector	Support
20-pin CoreSight™ debug and ETM connector	<ul style="list-style-type: none"> P-JTAG processor debug Serial Wire Debug (SWD) 4-bit trace
10-pin CoreSight™ debug connector	<ul style="list-style-type: none"> P-JTAG processor debug SWD
60-pin MIPI connector	<ul style="list-style-type: none"> P-JTAG processor debug SWD 32-bit trace
Arm® Keil® ULINKplus™ Embedded debug over USB using the USB-C connector, see 3.9.2 Debug over USB on page 57	<ul style="list-style-type: none"> P-JTAG processor debug SWD
14-pin ILA connector	FPGA debug



The availability of P-JTAG, SWD, 32-bit trace, and 4-bit trace depends on the design that you implement in the FPGA.

Related information

[A.1 Debug connectors](#) on page 76

3.9.2 Debug over USB

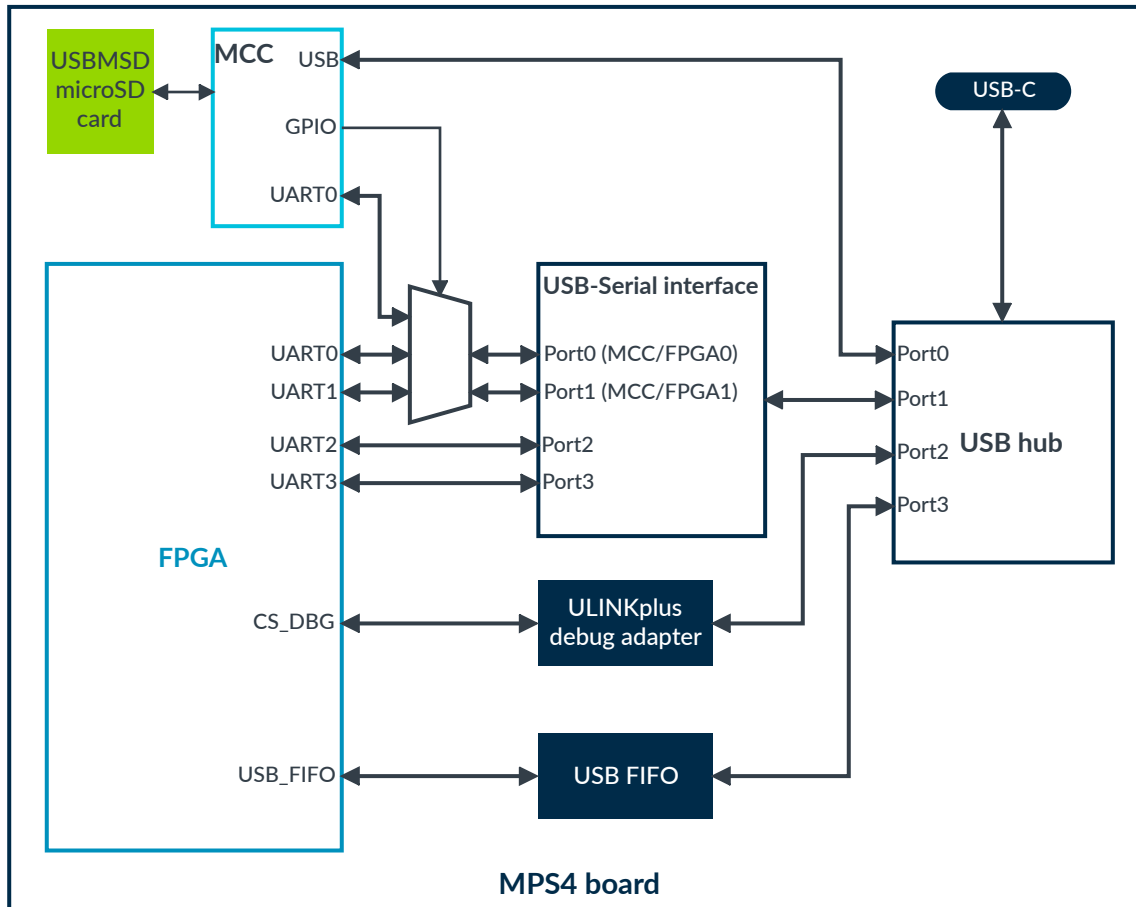
You can debug the MPS4 board over USB using the USB-C connector.

Debug over USB supports:

- 4 FPGA UARTs
- MCC debug
- Arm® Keil® ULINKplus™ Embedded debug adapter
- USB FIFO bridge, high-speed 160-Mbps 16-bit bus clocked at 100 MHz

The following figure shows the architecture of the debug over USB system.

Figure 3-26: MPS4 board debug over USB system architecture



Serial ports

The Ultrascale+ VU13P FPGA connects to 4 user serial ports, if required in your implementation. The MCC serial port (selectable as Port 0 or Port 1) indicates the status of the system during boot time. After the system has booted, you can use the serial ports for debug or status information, depending on your implementation. The 4 serial ports are concentrated into a USB-serial interface that connects to a 4-port USB hub, which is connected to the USB-C connector at the rear of the MPS4 board. This single USB-C connection gives a host computer access to the serial ports, the ULINKplus™ Embedded debug adapter, and the USB FIFO controller.

Debug using ULINKplus™ Embedded debug adapter

The onboard ULINKplus™ Embedded debug adapter enables debug over USB to the FPGA CoreSight™ components.

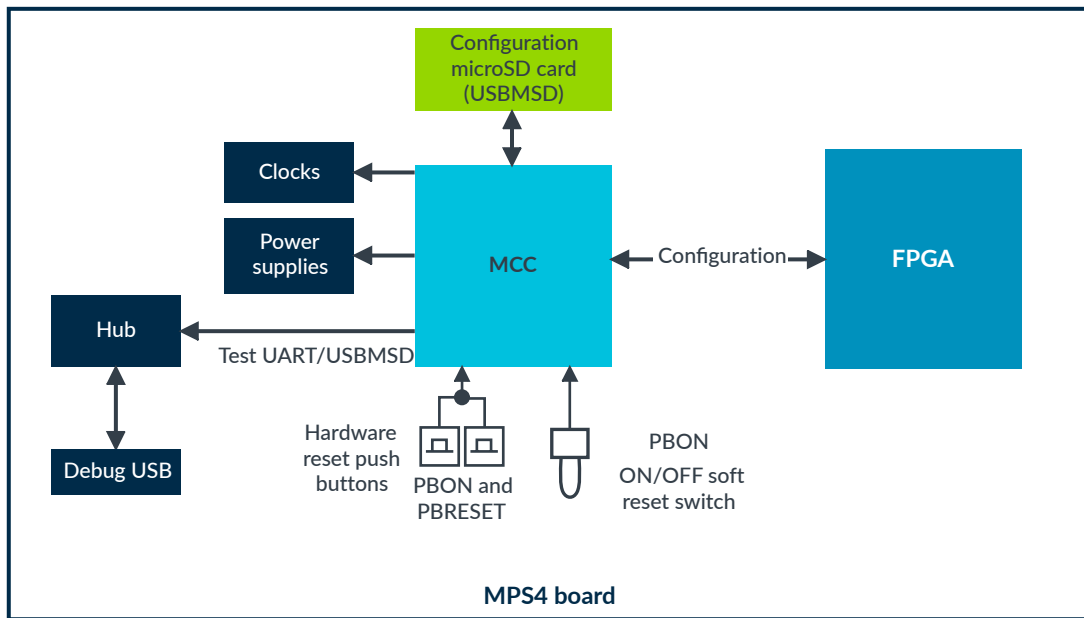
For information on setting up your host PC to connect through USB to the MPS4 board, see [2.3 Power up into the operating state](#) on page 13.

4. Configuration

The *Motherboard Configuration Controller* (MCC) controls the configuration process of the MPS4 board during powerup or reset. After powerup, and toggling up the ON/OFF soft reset switch (labeled *PBON*) or pressing the *PBON* push button, the configuration process begins and completes without further user intervention.

The following figure shows the board configuration system.

Figure 4-1: MPS4 board configuration system



The microSD card stores the board configuration files, including the `board.txt` and `config.txt` files. To learn more about the different files, see [4.2 Configuration files](#) on page 61.



You can access the configuration microSD card as a USB mass storage device.

Configuration flow

The MCC:

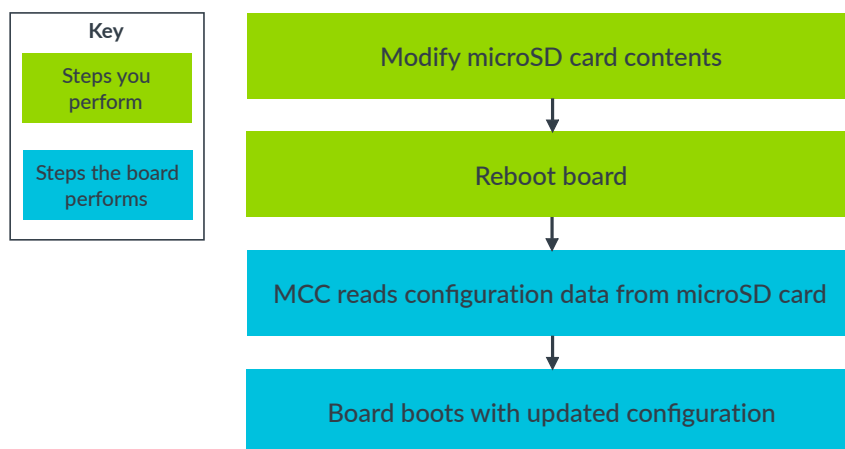
- Reads the FPGA image from the configuration microSD card and loads it into the FPGA.
- Sets the board oscillator frequencies using values from the MPS4 board FPGA implementation `.txt` file, usually `F1xxx_vx.txt`.

- If enabled, configures the FPGA *Serial Configuration Control* (SCC) registers using values from the `board.txt` file with the FPGA image.
- If enabled, loads the boot memory, QSPI, DDR4, or *Block RAM* (BRAM), with the boot image that the `images.txt` file defines.

For a more detailed description of the powerup and configuration sequence, see [C. Powerup and configuration sequence](#) on page 108.

The following figure shows the simple steps for you to configure the MPS4 board for your FPGA design.

Figure 4-2: MPS4 board configuration process



For more information on the steps to get started, see [2. Getting started](#) on page 11.

4.1 Using the reset switch and push buttons

The MPS4 board provides a toggle switch and push buttons that initiate reset and configuration. You can initiate a software reset or a hardware reset of the board.

ON/OFF reset switch

The MPS4 board provides an ON/OFF reset switch located at the back of the board near the USB-C connector, labeled *PBON*:

- The default position is middle.
- Toggling up initiates a hardware reset (*PBRESET*).
- Toggling down powers the board up or down, or initiates a software reset if pressed for more than 2 seconds (*PBON*).



Holding *PBON* in a down position during runtime for more than 2 seconds performs a software reset and the system enters the standby state. This action is similar to pressing the hardware reset push button *PBRESET* and resets the devices on the board, except that the *Motherboard Configuration Controller* (MCC) is not reset by this process.

If the board is already in the standby state, then toggling *PBON* down powers up the system.

To locate the ON/OFF reset switch on the MPS4 board, see [1.2 MPS4 board layout](#) on page 7.

ON/OFF reset push buttons

The MPS4 board provides 2 reset buttons located next to the FMC+ connector:

- A white button labeled *PBON* that powers the board up or down, or initiates a software reset if pressed for more than 2 seconds.
- A white button labeled *PBRESET* that initiates a hardware reset.

You can change the operation of the board from ON to standby by briefly pressing one of the reset buttons. Briefly pressing the button switches off the power to the board and resets the system to the default values. If you then toggle the reset switch, the system performs a full configuration and enters the run state.

To locate the ON/OFF reset push buttons on the MPS4 board, see [1.2 MPS4 board layout](#) on page 7.

Related information

[3.3.2 Reset, powerup, and configuration](#) on page 30

[C. Powerup and configuration sequence](#) on page 108

4.2 Configuration files

The configuration microSD card on the MPS4 board contains configuration files that control the board powerup and configuration process.

The microSD card that is supplied with the MPS4 board contains default configuration files. Because the board microSD card is non-volatile memory, it is only necessary to load new configuration files if you change the system configuration.

If you connect a host computer to the USB-C connector, the microSD card appears as a USB mass storage device. You can then add, edit, or delete files.

You can use a standard text editor that produces DOS line endings to read and edit the board configuration files.



File names and directory names are in 8.3 short file name format:

- File names that you generate must be in lowercase.
- Directory names must be in uppercase.
- All configuration files must use DOS line endings, that is, 0x0D/0x0A, or \n and \r, or LF and CF.

The following figure shows a typical example of the directory structure on the configuration microSD card memory.

Figure 4-3: Example USB mass storage device directory structure

MPS4 (F:)

```

|_ help.txt
|_ config.txt
|_ LOG.txt
|_ MB/
    |_ HBI0376B/
        |_ mbb_v1.ebf
        |_ board.txt
        |_ FI100/
            |_ fi100_v1.bit
            |_ fi100_v1.txt
            |_ images.txt
|_ SOFTWARE/
    |_ selftest.axf

```



The HBI number is a unique code that identifies the board. The MB directory contains subdirectories in the form HBI<BoardNumber><Boardrevision>, for example HBI0376B.

The directory structure and file name format ensure that each image is matched to the correct target device defined in the board configuration:

config.txt file

Generic configuration file for all MPS4 boards. This file applies to all Arm development boards including the MPS4 board. Some settings are specific to the MPS4 board.

See [4.2.1 config.txt board configuration file](#) on page 63.

MB directory

Contains subdirectories for any board variants that might be present in the system. The subdirectory names match the HBI codes for the specific board variants. The files in this directory contain clock, register, and other settings for the board.

See [4.2.2 Contents of the MB directory](#) on page 64.

SOFTWARE directory

Can contain user application files that the MCC can load into the QSPI, BRAM, or DDR4 on the board. The memory that is loaded depends on the image that you implement in the FPGA. The `images.txt` file defines the files that the MCC loads.

See [4.2.3 Contents of the SOFTWARE directory](#) on page 68.

Related information

[4. Configuration](#) on page 59

[C. Powerup and configuration sequence](#) on page 108

4.2.1 config.txt board configuration file

The board configuration file, `config.txt`, contains the parameters that the *Motherboard Configuration Controller* (MCC) reads to modify board behavior.

Syntax

- Colons (:) indicate the end of commands and must be separated by a space character (0x20) from the value fields.
- Semicolons (;) indicate comments. Any text after a semicolon and before the end of a line is a non-executable comment.

Parameters

For the list of `config.txt` parameters that the latest MCC firmware version supports, see the `help.txt` file on the configuration microSD card. You can also type `help configfile` from the MCC command-line interface.



The parameters available depend on what version of the MCC firmware your system uses. You can download the latest version of the firmware with the FPGA image packages at <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>.

Example: config.txt file

The following example shows a `config.txt` file that you can load into the configuration microSD card. This example might not be suitable for your particular design.

```
TITLE: Arm MPS4 FPGA prototyping board configuration file

[CONFIGURATION]
AUTORUN: FALSE           ;Auto Run from power up
AUTORUNDELAY: 3          ;Delay in seconds to wait for key press to stop boot up
UARTMODE: 0              ;0-MCC:FPGA0, 1-MCC:FPGA1, 2-MCC/FPGA0:FPGA1
FMC_FORCE: 1V8           ;Force FMC power ON
DVI_MODE: VGA            ;VGA/SVGA/XGA/SXGA/UXGA or HD1080 (MCC sets OSCCLK5)
FARM_MODE: FALSE         ;Enables Farm support for nPBON
```

4.2.2 Contents of the MB directory

The MPS4 board `MB` directory contains a configuration `HBI` subdirectory that matches the HBI code of the board.

The `HBI` subdirectory contains:

- A file in the format `mbb_vxxx.ebf`. This file is an MCC BIOS image.
- A `board.txt` file that defines the MCC BIOS image
- An subdirectory specific for the FPGA implementation that contains the following board configuration files:
 - Image files for the FPGA that have `.bit` extensions
 - An `images.txt` file that defines the `.axf` files that the MCC loads into external memory during configuration
 - An FPGA implementation `.txt` file, usually in the format `FIxxx_vx.txt`

4.2.2.1 board.txt file

The *Motherboard Configuration Controller* (MCC) BIOS board file, `board.txt`, defines:

- The MCC BIOS image
- The FPGA implementation `.txt` file, usually at `MB/HBI<BoardNumber><BoardRevision>/FIxxx/FIxxx_vx.txt`

Syntax

- Colons (:) indicate the end of commands and must be separated by a space character (0x20) from the value fields.
- Semicolons (;) indicate comments. Any text after a semicolon and before the end of a line is a non-executable comment.

Parameters

For the list of `board.txt` parameters that the latest MCC firmware version supports, see the `help.txt` file on the configuration microSD card. You can also type `help boardfile` from the MCC command-line interface.



The parameters available depend on what version of the MCC firmware your system uses. You can download the latest version of the firmware with the FPGA image packages at <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>.

Example: board.txt file

The following example shows a `board.txt` file that you can load into the configuration microSD card. This example might not be suitable for your particular design.

```
BOARD: HBI0376B
TITLE: Example board configuration file

[MCCS]
MBBIOS: mbb_v206.ebf      ;MB BIOS IMAGE
CPLD: pld_v101.dat        ;CPLD slim vme data file, .alg file should accompany this

[FPGA IMPLEMENTATION]
APPFILE: CA5DS\ca5ds.txt  ;Cortex-A5 Designstart FPGA
```

4.2.2.2 FPGA implementation .txt file

Each FPGA image package contains an FPGA implementation `.txt` file, usually `FIxxx_vx.txt`, which contains specific configuration information, including:

- The number of FPGAs on the board
- The number of oscillators and their frequencies
- The FPGA image file
- Details of the *Serial Configuration Control* (SCC) registers
- Configuration of the supported FPGA peripherals

Syntax

- Colons (:) indicate the end of commands and must be separated by a space character (0x20) from the value fields.
- Semicolons (;) indicate comments. Any text after a semicolon and before the end of a line is a non-executable comment.

Parameters

For the list of the `FIxxx_vx.txt` file parameters that the latest *Motherboard Configuration Controller* (MCC) firmware version supports, see the `help.txt` file on the configuration microSD card.



The parameters available depend on what version of the MCC firmware your system uses. You can download the latest version of the firmware with the FPGA image packages at <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>.

Example FPGA implementation txt file

The following example shows an example FPGA implementation `.txt` file that you can load into the configuration microSD card. This example might not be suitable for your particular design.

```
BOARD: HBI0376
```

```

TITLE: Example Arm Cortex-A5 DesignStart FPGA configuration file

[FPGAS]
TOTALFPGAS: 1                ;Total Number of FPGAs

; Top bit file
F0FILE: mps4_top.bit         ;FPGA0 Filename
F0MODE: FPGA                  ;FPGA0 Programming Mode

[OSCCLKS]
TOTALOSCCLKS: 7

; Clock generators OSC[5:0] connect to FPGA top level signals OSCCLKA[5:0]
; respectively. OSCCLKA[3:0] have a second identical output OSCCLKB[3:0],
; which connect to a different FPGA BANK to improve routing.
; FPGA top level signal REFCLK24 is driven by a fixed 24 MHz reference.
; Clockgen OSC6 drives the DDR reference clock c0_sys_clk_p/n.

OSC0: 200.0 ;OSCCLK[0] - CSI dphy_clk_200M_int video clocks
OSC1: 30.0 ;OSCCLK[1] - SSE-500 sysclk (UART fix - was 33.3/30.0)
OSC2: 100.0 ;OSCCLK[2] - Reserved
OSC3: 30.0 ;OSCCLK[3] - SSE-500 cpucclk
OSC4: 40.0 ;OSCCLK[4] - QSPI CLK (generated from this as OSCCLK4 divide by 2)
OSC5: 23.75 ;OSCCLK[5] - HDMI clock (VGA default)
;Note: OSC5 is powered from the FPGA FMC bank, which must be powered
;to use OSC5. See FMC_FORCE in config.txt.
OSC6: 100.0 ;c0_sys_clk_p/n - DDR ref clock (100MHz x9, 900MHz default)

[HARDWARE CONTROL]
ASSERTNPOR: TRUE             ;External resets assert nPOR
LEGACYRST: FALSE             ;Legacy CB_nPOR/CB_nRST reset mode
CPUWAIT: 0xFFFFFFFF          ;CPUWAIT value, set to 0xFFFFFFFF when using CB_nRST

[PERIPHERAL SUPPORT]
FPGA_SMB: TRUE               ;SMB interface is supported (MCC_SMC<>FPGA_SMB)
SMB_LAR: TRUE                ;SMB LAR interface is supported (MCC_SMC<>FPGA_SMB)

FPGA_SCC: TRUE               ;SCC interface is supported
SCCREG: 0x05300000           ;SCC registers base address

FPGA_DDR: TRUE               ;DDR interface is supported
DDRBASE: 0x41208000          ;DDR I2C register address
DDRTTEST: 0x80000000         ;DDR Memory address for system boot test

FPGA_SYSREG: FALSE           ;System register interface is supported
FPGAREG: 0x05302000          ;System registers base address

FPGA_REMAP: FALSE            ;REMAP interface is supported
REMAPREG: 0x05300000         ;REMAP register address
REMAP: BRAM                  ;REMAP boot device BRAM/QSPI. Must match REMAPVAL.
REMAPVAL: 0                  ;REMAP register value, for example, 0-BRAM. 1-QSPI

FPGA_HDMI: TRUE              ;HDMI interface is supported
HDMIBASE: 0x41313000         ;HDMI I2C register address

FPGA_RTC: FALSE              ;RTC PL031 interface is supported
RTCBASE: 0x0530B000          ;RTC PL031 base address

FPGA_PCIE: TRUE              ;PCIe interface is supported
PCIECFGBASE: 0x50000000      ;PCIe CFG base address
PCIEMEMBASE: 0x60000000      ;PCIe MEMORY MAPPED base address

FPGA_QSPI: TRUE              ;QSPI interface is supported
;QSPITEST: TRUE              ;QSPI RW and XIP test
QSPIBASE: 0x40000000         ;QSPI RW controller base address
QSPIDATA: 0x40000000         ;QSPI RW data address
XIPBASE: 0x08000000          ;QSPI XIP controller base address
QSPISCC: 0x80000000          ;QSPI SCC register (0x00000xxx=SMB access/0x80000xxx=SCC
access)

[SCC REGISTERS]

```

```
TOTALSYSCONS: 0 ;Total Number of SYSCON registers defined
SYSCON: 0x000 0x00000001
```

If you implement a peripheral in your FPGA image, set the peripheral flag to `TRUE`. Set the base address for the peripheral correctly to match the base address defined in the FPGA image.

If you do not implement a peripheral in your FPGA image, set the peripheral flag to `FALSE` in the [PERIPHERAL SUPPORT] section, for example, `FPGA_RTC: FALSE`. In this case, the base address is not used and you can set it to 0.

4.2.2.3 images.txt file

The `images.txt` file defines the `.axf` and `.bin` files that the *Motherboard Configuration Controller* (MCC) loads into external memory during configuration.

Syntax

- Colons (:) indicate the end of commands and must be separated by a space character (0x20) from the value fields.
- Semicolons (;) indicate comments. Any text after a semicolon and before the end of a line is a non-executable comment.

Parameters

For the list of `images.txt` parameters that the latest MCC firmware version supports, see the `help.txt` file on the configuration microSD card. You can also type `help imagefile` from the MCC command-line interface.



The parameters available depend on what version of the MCC firmware your system uses. You can download the latest version of the firmware with the FPGA image packages at <https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images>.

Example: images.txt file

The following example shows `images.txt` that you can load into the configuration microSD card. This example might not be suitable for your particular design.

```
;TITLE: Example Arm MPS4 FPGA prototyping board images configuration File
;
;NONE      - No action
;AUTO      - Auto legacy QSPI (if CS0 and REMAP==2) or RAM update
;FORCE     - Force legacy QSPI (if CS0 and REMAP==2) or RAM update
;RAM       - RAM update
;AUTOQSPI  - Auto QSPI update
;FORCEQSPI - Force QSPI update

[IMAGES]
;Selftest booting from SRAM
TOTALIMAGES: 1 ;Number of Images (Max: 32)
IMAGE0ADDRESS: 0x00000000 ;Select the required executable program
IMAGE0UPDATE: FORCE ;Image Update:NONE/AUTO/FORCE/RAM/AUTOQSPI/
FORCEQSPI
```

```
IMAGE0FILE: \SOFTWARE\selftest.axf
```

For information on the `IMAGEEXPORT` and `IMAGEADDRESS` parameters, see [3.4.2 Static Memory Controller interface](#) on page 35.

`.axf` and `.elf` files are treated as ELF files. All other files are treated as binary.

4.2.3 Contents of the SOFTWARE directory

The `SOFTWARE` directory can contain applications that you can load into the QSPI flash, BRAM, or DDR4 memory. The memory that is loaded depends on the image that you implement in the FPGA.

You can create applications and load them into the memory on the board. Application images are typically boot images or demo programs and have a `.axf` extension. For example, `selftest.axf` board test software.



File names and directory names are in 8.3 short file name format:

- File names that you generate must be in lowercase.
- Directory names must be in uppercase.
- All configuration files must use DOS line endings, that is, `0x0D/0x0A`, or `\n` and `\r`, or `LF` and `CF`.

4.3 MCC command-line interface

The MPS4 board command-line interface sends commands to the *Motherboard Configuration Controller* (MCC). You must connect a host computer to UART0 to enter command-line input.

Your terminal emulator settings must be:

- 115200 baud rate
- 8N1, that is, 8 data bits, no parity, and 1 stop bit
- No hardware or software flow control

4.3.1 MCC main command menu

In the MPS4 board *Motherboard Configuration Controller* (MCC) main command menu, enter the `HELP` or `?` command. This help shows the commands and their descriptions for the main menu.

The following table shows the MCC main menu system commands.

Table 4-1: MPS4 board MCC main command menu

Command	Description
CAP "fname" [/A]	Capture serial data to a file /A option appends data to a file
FILL "fname" [nnnn]	Create a file filled with text nnnn specifies the number of lines to create. The default is 1000.
TYPE "fname"	Display the contents of a text file
REN "fname1" "fname2"	Rename a file from fname1 to fname2
COPY "fin" ["fin2"] "fout"	Copy a file fin to fout file fin2 option merges fin and fin2
DEL "fname"	Delete a file
DIR "[mask]"	Display a list of files in the directory
FORMAT [label]	Format Flash memory card
USB_ON	Enable USB
USB_OFF	Disable USB
USB_INIT	Re-initialize USB
SHUTDOWN	Shut down the power supply but leave the MCC running. The board returns to Standby mode
REBOOT	Cycle system power and reboot
RESET	Reset board using CB_nRST,
RESET_ON	Reset CPU and set CPUWAIT
RESET_OFF	Release CPUWAIT
DEBUG	Enter debug menu
BOARD_ID	Display the unique board ID
REVISION	Display the board revision
HELP or ?	Display help
HELP "flag"	Display help on a flag
HELP CONFIGFILE	Display the configuration file flags
HELP BOARDFILE	Display the board file flags
HELP IMAGEFILE	Display the image file flags
HELP VERSION	Display the MCC version history

The following commands are only available in run mode.

Table 4-2: MPS4 board MCC run mode commands

READ_AXI "fname" "address" "end_address"	Read system memory to file fname from address to end address
WRITE_AXI "fname" "address"	Write file fname to system memory at address (RAM only)

4.3.2 MCC debug menu

To switch to the debug menu, enter `DEBUG` at the *Motherboard Configuration Controller* (MCC) main menu. The debug menu is valid only in the run state. In the MPS4 board debug menu, enter the `HELP` or `?` command. This help shows the commands and their descriptions for the debug menu.

Table 4-3: MPS4 board MCC debug command menu

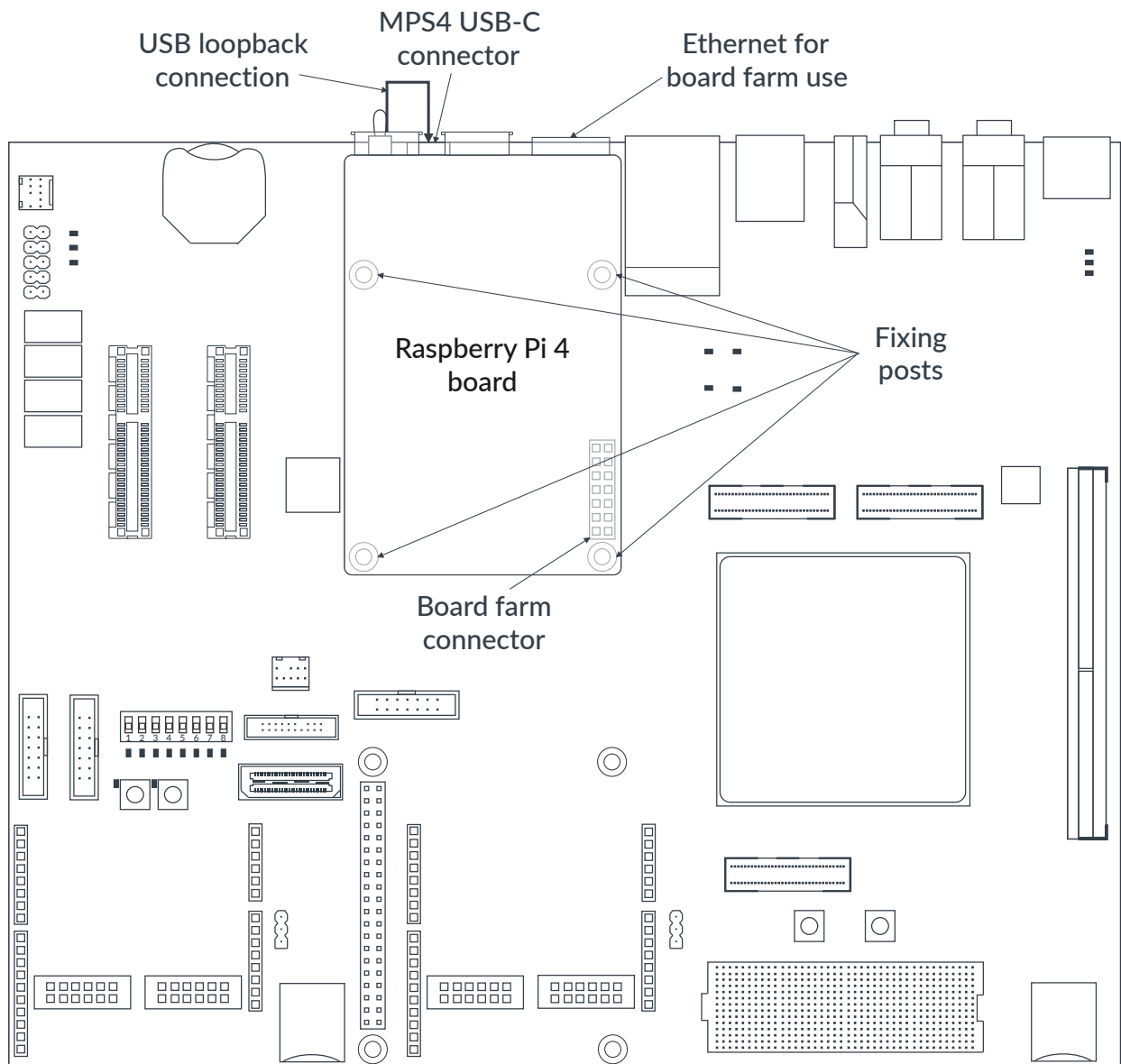
Command	Description
DEBUG "0 1"	0 Disable debug printing 1 Enable debug printing
USERIO "6 7" "0 1"	0 Clear user I/O (CB_RSVD6/7) 1 Set user I/O (CB_RSVD6/7)
WRITEFMC "fname"	Write FMC text file to EEPROM
READFMC	Read FMC EEPROM and display info
SAVEFMC "fname"	Read FMC EEPROM and save to binary file
FMCPWR	Enable power for FMC EEPROM
EXIT or QUIT	Return to main menu
HELP or ?	Display the help

4.4 Configuring remote farm access

You can configure remote farm access to the MPS4 board through the board farm interface, which is compatible with a Raspberry Pi 4 board. This interface provides connectivity and power. The farm interface board is loaded with a Python boot script that controls the MPS4 board resets.

The following figure shows a Raspberry Pi 4 board mounted upside down onto the MPS4 board. The boards share a USB loopback connection using the USB-C connector. This USB loopback connection provides access to the file system and debug tools. Fixing posts are also available on the MPS4 board that you can use to screw the farm interface board into position.

Figure 4-4: Raspberry Pi 4 board mounted upside down onto the MPS4 board



To enable farm access, you must modify the `config.txt` file on the board configuration microSD card to include this line:

```
FARM_MODE: TRUE           ;Farm slot nPBON enable
```

For more information about the `config.txt` file, see [4.2.1 config.txt board configuration file](#) on page 63.

The following example Python script enables remote farm access from the Raspberry Pi 4 used as a farm interface board. Running this script means that the farm interface can control and read the status of resets of the MPS4 board.

```
import RPi.GPIO as GPIO
import time

#Set pin mapping mode
GPIO.setmode(GPIO.BCM)

# MPS4 RPi pin mapping
CB_FARM0 = 2
CB_FARM1 = 3
CB_FARM2 = 4
CB_FARM3 = 14
CB_FARM4 = 15
CB_FARM5 = 17
CB_FARM6 = 18
CB_FARM7 = 22
CB_FARM8 = 23
CB_FARM9 = 27

# Used pins
CB_nPOR = CB_FARM5
CB_nRST = CB_FARM6
CB_nRUN = CB_FARM7
NPBON = CB_FARM8
NPBRESET = CB_FARM9

# Note: to avoid unwanted resets when program starts we only configure NPBON/
NPBRESET when used

# IO Direction
GPIO.setup(CB_nPOR, GPIO.IN)
GPIO.setup(CB_nRST, GPIO.IN)
GPIO.setup(CB_nRUN, GPIO.IN)
#GPIO.setup(NPBON, GPIO.OUT)
#GPIO.setup(NPBRESET, GPIO.OUT)

# Default IO levels using open drain drive
#GPIO.output(NPBON, GPIO.LOW)
#GPIO.output(NPBRESET, GPIO.LOW)

# IO Direction
#GPIO.setup(NPBON, GPIO.IN)
#GPIO.setup(NPBRESET, GPIO.IN)

print("-----")
print("MPS4 RPi Farm GPIO test ")
print("PBRESET and PBON simulator")
print("-----")
print("")

while True:
    print("Options: 0-Status, 1-MCCreset, 2-Boot, 3-Reboot, 4-FPGAreset or 5-
Shutdown")
    print("")
    print("Cmd> ")
    choice = input()

    if (choice == 'Status') or (choice == '0'):
        # Display board status
        print("Board status:")
        if (GPIO.input(CB_nPOR)):
            print("CB_nPOR = 1/High")
        else:
            print("CB_nPOR = 0/Low")
```



```

        if(GPIO.input(CB_nRST)):
            print("CB_nRST = 1/High")
        else:
            print("CB_nRST = 0/Low")

        # Note: MCC always drives CB_RUN low but in standby CPLD if OFF so line is
        # pulled up
        if(GPIO.input(CB_nRUN)):
            print("CB_nRUN = 1/High")
        else:
            print("CB_nRUN = 0/Low")
        print("")

    if (choice == 'MCCreset') or (choice == '1'):
        # Apply nPBRESET
        print("Assert MCC")
        GPIO.setup(NPBRESET, GPIO.OUT)
        GPIO.output(NPBRESET, GPIO.LOW)
        time.sleep(0.5)
        GPIO.setup(NPBRESET, GPIO.IN)
        time.sleep(5.0)
        print("MCC reset")
        print("")

    if (choice == 'Reboot') or (choice == '3'):
        # Apply nPBON for 3.0sec
        print("Assert nPBON (for 3sec)")
        GPIO.setup(NPBON, GPIO.OUT)
        GPIO.output(NPBON, GPIO.LOW)
        time.sleep(3.0)
        GPIO.setup(NPBON, GPIO.IN)
        time.sleep(5.0)
        print("Board shutdown")
        print("")
        # Apply nPBON
        print("Assert nPBON")
        GPIO.setup(NPBON, GPIO.OUT)
        time.sleep(0.5)
        GPIO.setup(NPBON, GPIO.IN)
        time.sleep(0.5)
        print("Board booting...")
        print("")

    if (choice == 'Boot') or (choice == 'FPGAreset') or (choice == '2') or (choice
    == '4'):
        # Apply nPBON for 0.5sec
        print("Assert nPBON")
        GPIO.setup(NPBON, GPIO.OUT)
        GPIO.output(NPBON, GPIO.LOW)
        time.sleep(0.5)
        GPIO.setup(NPBON, GPIO.IN)
        time.sleep(0.5)
        print("Board boot or FPGA reset")
        print("")

    if (choice == 'Shutdown') or (choice == '5'):
        # Apply nPBON for 3.0sec
        print("Assert nPBON (for 3sec)")
        GPIO.setup(NPBON, GPIO.OUT)
        GPIO.output(NPBON, GPIO.LOW)
        time.sleep(3.0)
        GPIO.setup(NPBON, GPIO.IN)
        time.sleep(0.5)
        print("Board shutdown")
        print("")

```

Related information

[3.6.5 Remote board farm interface](#) on page 51

[A.2.4 Raspberry Pi HAT and board farm connectors](#) on page 89

[A.4.2 Ethernet and USB 3.0 connectors](#) on page 103

4.5 Troubleshooting

You can find solutions to common troubleshooting issues online and in our documentation:

- For common troubleshooting issues that might affect the MPS4 board, see <https://community.arm.com/oss-platforms/w/docs/560/fpga-prototyping-boards>.
- For specific troubleshooting issues with particular FPGA images, see the relevant documentation for the respective FPGA image package.
- For any other issues with the MPS4 board, contact Arm at <https://developer.arm.com/support>.

Appendix A Connector and signal descriptions

This appendix describes the MPS4 board interface connectors and lists the signals for each connector.

The following table shows a list of the main MPS4 board connectors that you can use to connect peripherals to develop your design.

Table A-1: MPS4 board connectors

Usage	Connectors	J numbers
Debug	60-pin MIPI debug connector	J18
Debug	20-pin CoreSight™ debug and ETM connector	J19
Debug	10-pin CoreSight™ debug connector	J20
Debug	14-pin F-JTAG ILA debug connector	J22
Expansion connectivity	Arduino Shield 0 interface digital I/O connectors	J29, J32
Expansion connectivity	Arduino Shield 0 interface analog I/O connector	J31
Expansion connectivity	Arduino Shield 0 interface power and voltage reference	J30
Expansion connectivity	Arduino Shield 1 interface digital I/O connectors	J34, J37
Expansion connectivity	Arduino Shield 1 interface analog I/O connector	J36
Expansion connectivity	Arduino Shield 1 interface power and voltage reference	J35
Expansion connectivity	Pmod 0 connector	J39
Expansion connectivity	Pmod 1 connector	J42
Expansion connectivity	Pmod 2 connector	J40
Expansion connectivity	Pmod 3 connector	J43
Expansion connectivity	Raspberry Pi HAT connector	J41
Expansion connectivity, remote board farm interface	Raspberry Pi board farm connector	J5
Expansion connectivity	FMC+ HPC connector	J11
Board-to-board connectivity, QSFP interface	Samtec AcceleRate ARC6 connectors	J48, J49, J50
Display	HDMI connector	J2
Audio	3.5m line-out stacked audio connectors	J3
Audio	3.5m line-in stacked audio connectors	J4
Display	MIPI DSI connector	J45
Camera	MIPI 2-lane CSI connectors	J46, J47
Camera	MIPI 4-lane CSI connector	J44

Usage	Connectors	J numbers
Expansion connectivity	PCIe expansion slots	J52, J53
Expansion power	PCIe power connectors	J75, J76
General	Gigabit Ethernet and USB 3.0 connectors	J54
General	USB 3.0 connectors	J55
Power	12V power connector	J71

To locate the MPS4 board connectors, see the layout diagram in [1.2 MPS4 board layout](#) on page 7.

For the FPGA pin assignment spreadsheet and I/O timing information, see <https://developer.arm.com/mps4>.

A.1 Debug connectors

The MPS4 board provides connectors that support P-JTAG processor debug, F-JTAG FPGA debug, 16-bit and 4-bit trace, and *Serial Wire Debug* (SWD):

- 60-pin MIPI debug connector (J18)
- 20-pin CoreSight™ debug and ETM connector (J19)
- 10-pin CoreSight™ debug connector (J20)
- 14-pin FPGA ILA debug connector (J22)



All debug signals operate at 1V8.

You can also use the ARC6 connectors for debug, see [A.2.6 Samtec AcceleRate ARC6 connectors](#) on page 93.

To locate the debug connectors, including the USB-C connector (J7), see [1.2 MPS4 board layout](#) on page 7.

A.1.1 60-pin MIPI debug connector

The 60-pin MIPI connector is a 60-pin 0.5mm pitch socket that supports JTAG debug, *Serial Wire Debug* (SWD), and 32-bit trace.



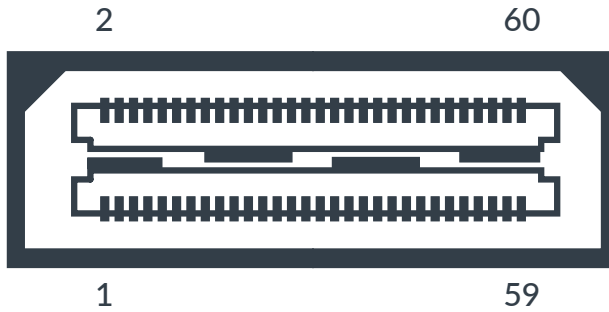
Only use one of the following connectors at a time because they share some signals:

- 60-pin MIPI debug connector
- 20-pin CoreSight™ debug and ETM connector

- 10-pin CoreSight™ debug connector

The following figure shows the 60-pin MIPI connector, J18.

Figure A-1: 60-pin MIPI debug connector



The following table shows the pin mapping for the 60-pin MIPI debug connector, J18.



The 60-pin MIPI debug connector is 1V8 only.

Table A-2: 60-pin MIPI debug connector pin mapping

Pin	Signal	Pin	Signal
1	VTREF (1V8)	2	TMS/SWDIO
3	TCK	4	TDO
5	TDI	6	nSRST
7	RTCK	8	TRST_PD
9	nTRST	10	DBGREQ
11	DBGACK	12	TRACE_VTREF
13	TRACE_CLKA	14	TRACE_CLKB
15	nDET	16	GND
17	TRACECTL	18	TRACEDATA[19]
19	TRACEDATA[0]	20	TRACEDATA[20]
21	TRACEDATA[1]	22	TRACEDATA[21]
23	TRACEDATA[2]	24	TRACEDATA[22]
25	TRACEDATA[3]	26	TRACEDATA[23]
27	TRACEDATA[4]	28	TRACEDATA[24]
29	TRACEDATA[5]	30	TRACEDATA[25]
31	TRACEDATA[6]	32	TRACEDATA[26]

Pin	Signal	Pin	Signal
33	TRACEDATA[7]	34	TRACEDATA[27]
35	TRACEDATA[8]	36	TRACEDATA[28]
37	TRACEDATA[9]	38	TRACEDATA[29]
39	TRACEDATA[10]	40	TRACEDATA[30]
41	TRACEDATA[11]	42	TRACEDATA[31]
43	TRACEDATA[12]	44	Reserved
45	TRACEDATA[13]	46	Reserved
47	TRACEDATA[14]	48	Reserved
49	TRACEDATA[15]	50	Reserved
51	TRACEDATA[16]	52	Reserved
53	TRACEDATA[17]	54	Reserved
55	TRACEDATA[18]	56	Reserved
57	GND	58	GND
59	Reserved	60	Reserved



The TRST_PD signal (pin 8) connects to nTRST (pin 9).

A.1.2 20-pin CoreSight™ debug and ETM connector

The 20-pin CoreSight™ connector is a 1V8 1.27mm CoreSight™ debug and *Embedded Trace Macrocell* (ETM) connector that supports P-JTAG debug, *Serial Wire Debug* (SWD), and 4-bit trace.

The 20-pin CoreSight™ debug and ETM connector connects to general-purpose pins on the FPGA. The availability of P-JTAG, SWD, or 4-bit trace depends on the design that you implement in the FPGA.

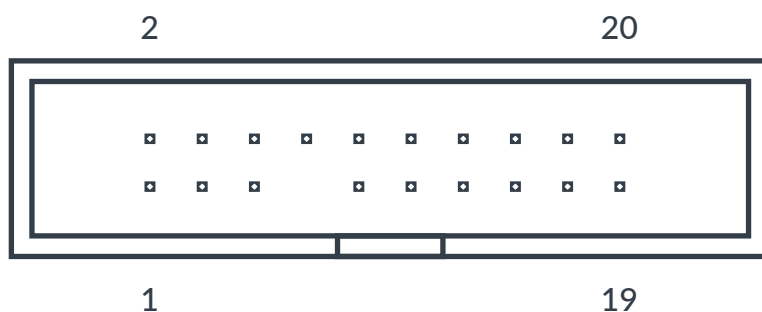


Only use one of the following connectors at a time because they share some signals:

- 60-pin MIPI debug connector
- 20-pin CoreSight™ debug and ETM connector
- 10-pin CoreSight™ debug connector

The following figure shows the 20-pin CoreSight™ debug and ETM connector, J19.

Figure A-2: 20-pin CoreSight™ debug and ETM connector



The following table shows the pin mapping for each P-JTAG, SWD, and 4-bit trace signal on the 20-pin CoreSight™ debug and ETM connector, J19.



The 20-pin CoreSight™ debug and ETM connector is 1V8 only.

Table A-3: 20-pin CoreSight™ debug and ETM connector pin mapping

Pin	Signal	Pin	Signal
1	VTREF(1V8)	2	SWDIO/TMS
3	GND	4	SWDCLK/TCK
5	GND	6	SWO/TDO
7	KEY	8	No connection/TDI
9	GNDDETECT	10	nSRST
11	CS_01	12	TRACECLK
13	CS_02	14	TRACEDATA[0]
15	GND	16	TRACEDATA[1]
17	GND	18	TRACEDATA[2]
19	GND	20	TRACEDATA[3]



- Pins 2, 6, 8, 9, and 10 have pull-up resistors to 1V8.
- Pin 4 has a pull-down resistor to GND.

A.1.3 10-pin CoreSight™ debug connector

The 10-pin CoreSight™ connector is a 1V8 1.27mm CoreSight™ debug connector that supports P-JTAG debug and Serial Wire Debug.

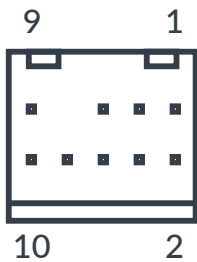


Only use one of the following connectors at a time because they share some signals:

- 60-pin MIPI debug connector
- 20-pin CoreSight™ debug and ETM connector
- 10-pin CoreSight™ debug connector

The following figure shows the 10-pin CoreSight™ debug connector, J20.

Figure A-3: 10-pin CoreSight™ debug connector



The following table shows the pin mapping for each P-JTAG and SWD signal on the 10-pin CoreSight™ debug connector, J20.



The 10-pin CoreSight™ debug connector is 1V8 only.

Table A-4: 10-pin CoreSight™ debug connector pin mapping

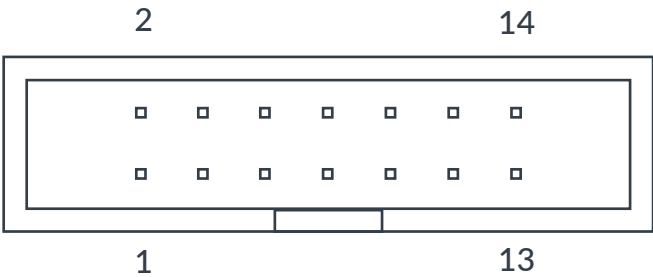
Pin	Signal	Pin	Signal
1	1V8	2	SWDIO/TMS
3	GND	4	SWDCLK/TCK
5	GND	6	SWO/TDO
7	KEY	8	No connection/TDI
9	GNDDTECT	10	nSRST

A.1.4 14-pin F-JTAG ILA debug connector

The MPS4 board provides one 14-pin 2mm ILA debug connector to debug the FPGA. It enables you to connect an ILA device, such as SignalTap II, to a hard FPGA JTAG chain in the FPGA and debug your design.

The following figure shows the 14-pin ILA FPGA debug connector, J22.

Figure A-4: 14-pin ILA debug connector



The following table shows the pin mapping for the ILA FPGA debug connector, J22.



The 14-pin ILA FPGA debug connector is 1V8 only.

Table A-5: 14-pin ILA FPGA debug connector pin mapping

Pin	Signal	Pin	Signal
1	GND	2	VIO (1V8)
3	GND	4	TMS
5	GND	6	TCK
7	GND	8	TDO
9	GND	10	TDI
11	GND	12	No connection
13	GND	14	No connection



- Pins 4, 8, and 10 have pull-up resistors to 1V8.
- Pin 6 has a pull-down resistor to GND.

A.2 I/O connectors

The MPS4 board provides the following I/O connectors:

- 2 Arduino Shield expansion connectors
- 4 Pmod connectors
- 1 Raspberry Pi HAT connector
- 1 FMC+ HPC connector with 560 pins that supports the 400-way FMC HPC standard
- 3 Samtec AcceleRate ARC6 board-to-board connectors, including one connector that you can use for the QSFP interface

A.2.1 Arduino Shield, Pmod, and HAT shared SHx_I/Ox signals

The Arduino Shield, Pmod, and HAT connectors each share some of the same digital I/O signals.



The MPS4 board supports simultaneous use of Pmod, Shield, and HAT expansion. However, you must take care when driving the shared signals to ensure each signal is only used on one expansion interface at a time. The following combinations are supported:

- Shield 0 and Shield 1
- Shield 0 and Pmod 2/3
- Shield 1 and Pmod 0/1
- Pmod 0/1 and Pmod 2/3
- HAT

The SHx_I/Ox signals are level shifted on-board from the FPGA 1V8 I/O voltage to the connector 1V8/3V3 I/O voltage. The level shifter requires a direction control signal for each pin, SHx_nENx. When SHx_nENx is LOW, the associated SHx_I/Ox signal is an output to the connector. When SHx_nENx is HIGH, the associated SHx_I/Ox signal is an input from the connector. For more information, see the diagram in [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43.

The following table shows the Shield 0 signals and what is shared with the Pmod and HAT connectors. The table also shows what FPGA pin each signal uses.

Table A-6: Shield 0 signals shared with the Pmod and HAT connectors

Shield 0 signal	Pmod 0 J39 pin	Pmod 1 J42 pin	HAT J41 pin	FPGA pin
SH0_IO0	-	3	27	M15
SH0_IO1	-	2	28	M13
SH0_IO2	-	8	3	M12
SH0_IO3	-	1	5	M11
SH0_IO4	-	7	7	L15

Shield 0 signal	Pmod 0 J39 pin	Pmod 1 J42 pin	HAT J41 pin	FPGA pin
SH0_IO5	7	-	29	L14
SH0_IO6	8	-	31	L13
SH0_IO7	9	-	26	L11
SH0_IO8	10	-	24	K14
SH0_IO9	-	4	21	V13
SH0_IO10	1	-	19	K13
SH0_IO11	2	-	23	K12
SH0_IO12	3	-	32	K11
SH0_IO13	4	-	33	J14
SH0_IO14	-	10	8	J12
SH0_IO15	-	9	10	J11

The following table shows the Shield 1 signals and what is shared with the Pmod and HAT connectors. The table also shows what FPGA pin each signal uses.

Table A-7: Shield 1 signals shared with the Pmod and HAT interfaces

Shield 1 signal	Pmod 2 J40 pin	Pmod 3 J43 pin	HAT J41 pin	FPGA pin
SH1_IO0	-	3	36	U12
SH1_IO1	-	2	11	U11
SH1_IO2	-	8	12	T11
SH1_IO3	-	1	35	R11
SH1_IO4	-	7	38	R12
SH1_IO5	7	-	40	R13
SH1_IO6	8	-	15	W15
SH1_IO7	9	-	16	P11
SH1_IO8	10	-	18	P12
SH1_IO9	-	4	22	P14
SH1_IO10	1	-	37	R16
SH1_IO11	2	-	13	P15
SH1_IO12	3	-	-	N15
SH1_IO13	4	-	-	N14
SH1_IO14	-	10	-	N13
SH1_IO15	-	9	-	N12

Related information

[A.2.2 Arduino Shield connectors](#) on page 83

[A.2.3 Pmod connectors](#) on page 87

[A.2.4 Raspberry Pi HAT and board form connectors](#) on page 89

A.2.2 Arduino Shield connectors

Connectors on the MPS4 board provide 2 Arduino Shield expansion interfaces. Each interface provides 16 digital I/O and six analog I/O. The digital I/O supports 1V8 or 3V3 signaling.

The *Peripheral module* interface (Pmod) connectors and the HAT connector each share part of the Shield connectors and are wired in parallel with them. Interface Pmod 0/1 shares all signals with Shield 0 interface, Pmod 2/3 shares all signals with Shield 1 interface, and the HAT interface shares signals with both Shield interfaces. For more information on the shared signals, see [A.2.1 Arduino Shield, Pmod, and HAT shared SHx_I/Ox signals](#) on page 82.

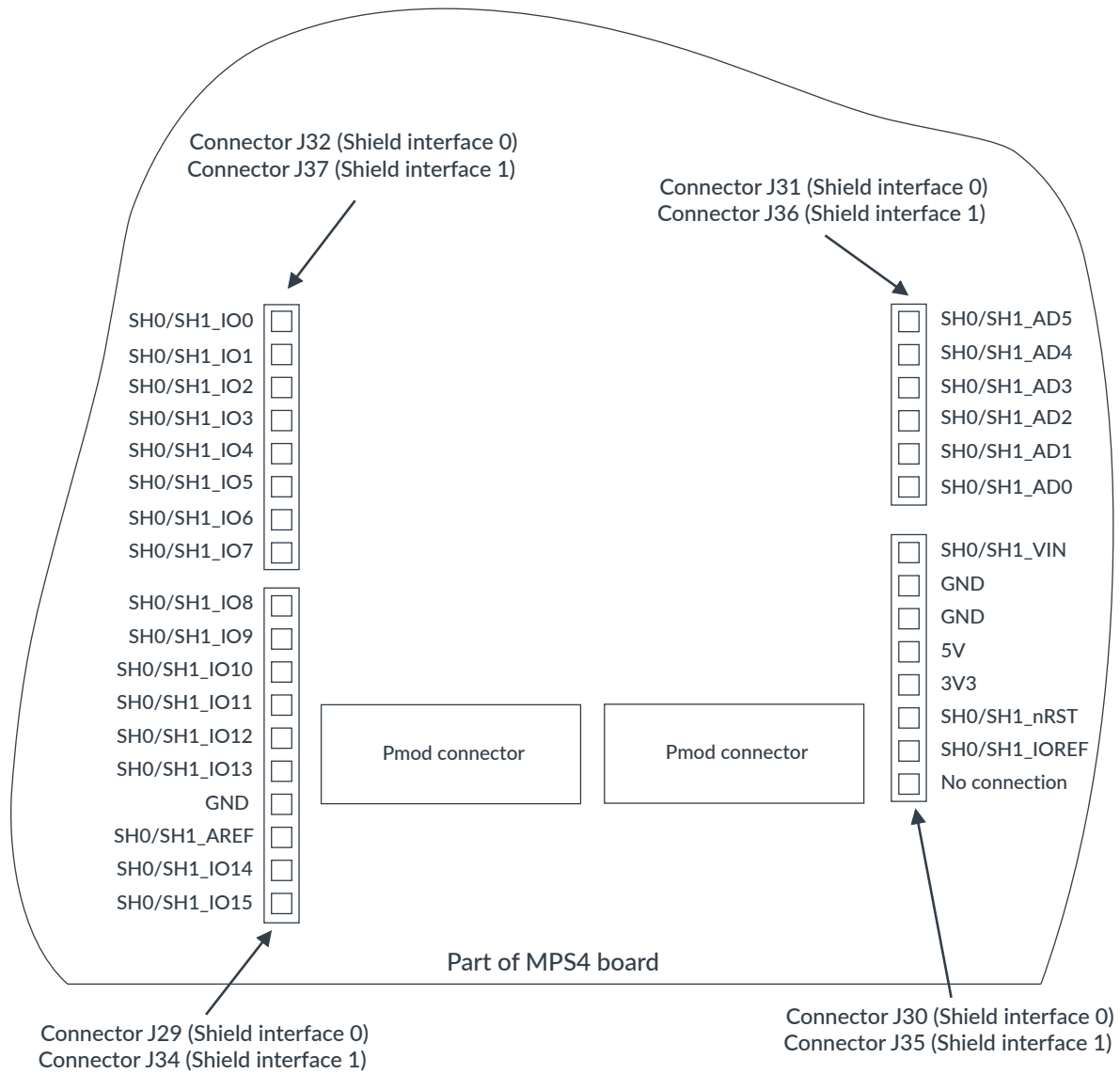


The MPS4 board supports simultaneous use of Pmod, Shield, and HAT expansion. However, you must take care when driving the shared signals to ensure each signal is only used on one expansion interface at a time.

Shield 0 and Shield 1 interface connectors

The following figure shows a combined diagram of the Arduino Shield 0 and Arduino Shield 1 interfaces on the MPS4 board.

Figure A-5: Shield 0 and Shield 1 interface connectors on the MPS4 board



Connectors J29, J30, J31, and J32 form the Shield 0 interface.

Connectors J34, J35, J36, and J37 form the Shield 1 interface.



User-links select digital I/O operating voltages and power inputs. The power inputs and IOREF voltages have maximum current limits available at the board interface pins.

For information on the user-links, and the maximum available IOREF currents, see [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47.

Digital I/O connectors: Connectors J29, J32, J34, and J37

Connector J29 provides Shield 0 I/O[15:8], and connector J34 provides Shield 1 I/O[15:8]. The connectors also provide the analog I/O reference voltages. The following table shows the pin mapping for connectors J29 and J34.

Table A-8: Connectors J29 (Shield 0) and J34 (Shield 1) signal list

Pin	Signal
1	SH0/SH1_IO8
2	SH0/SH1_IO9
3	SH0/SH1_IO10
4	SH0/SH1_IO11
5	SH0/SH1_IO12
6	SH0/SH1_IO13
7	GND
8	SH0/SH1_AREF
9	SH0/SH1_IO14
10	SH0/SH1_IO15

Connector J32 provides Shield 0 I/O[7:0] and connector J37 provides Shield 1 I/O[7:0]. The following table shows the pin mapping for connectors J32 and J37.

Table A-9: Connectors J32 (Shield 0) and J37 (Shield 1) signal list

Pin	Signal
1	SH0/SH1_IO0
2	SH0/SH1_IO1
3	SH0/SH1_IO2
4	SH0/SH1_IO3
5	SH0/SH1_IO4
6	SH0/SH1_IO5
7	SH0/SH1_IO6
8	SH0/SH1_IO7

Analog I/O connectors: Connectors J31 and J36

Connector J31 provides six analog I/O for Shield 0 and connector J36 provides six analog I/O for Shield 1. The following table shows the pin mapping for connectors J31 and J36.

Table A-10: Connectors J31 (Shield 0) and J36 (Shield 1) signal list

Pin	Signal
1	SH0/SH1_ADO

Pin	Signal
2	SH0/SH1_AD1
3	SH0/SH1_AD2
4	SH0/SH1_AD3
5	SH0/SH1_AD4
6	SH0/SH1_AD5

Power and voltage references: Connectors J30 and J35

Connector J30 provides power and voltage references for Shield 0 digital I/O. Connector J35 provides power and voltage references for Shield 1 digital I/O. The following table shows the pin mapping for Shield connectors J30 and J35.

Table A-11: Connectors J30 (Shield 0) and J35 (Shield 1) signal list

Pin	Signal
1	No connection
2	SH0/SH1_IOREF
3	SH0/SH1_nRST
4	3V3
5	5V
6	GND
7	GND
8	SH0/SH1_VIN

Related information

- [1.2 MPS4 board layout](#) on page 7
- [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43
- [A.2.3 Pmod connectors](#) on page 87
- [A.2.4 Raspberry Pi HAT and board farm connectors](#) on page 89
- [Shield expansion, Pmod, and HAT expansion power](#) on page 105

A.2.3 Pmod connectors

The *Peripheral module* (Pmod) connectors on the MPS4 board provide digital I/O expansion capability, an alternative to the Arduino Shield interfaces.

The four Pmod connectors enable fitting of Type 2A (J39/J40) and Type 3/6 (J42/J43) boards:

- Connectors J39 and J42 form interface Pmod 0/1
- Connectors J40 and J43 form interface Pmod 2/3

The Pmod connectors carry a subset of the signals on the Arduino Shield connectors and HAT connector, and are wired in parallel with these other connectors. Interface Pmod 0/1 shares all signals with Shield 0 interface, Pmod 2/3 shares all signals with Shield 1 interface, and the HAT

interface shares signals with both Shield interfaces. For more information on the shared signals, see [A.2.1 Arduino Shield, Pmod, and HAT shared SHx_IOx signals](#) on page 82.



The MPS4 board supports simultaneous use of Pmod, Shield, and HAT expansion. However, you must take care when driving the shared signals to ensure each signal is only used on one expansion interface at a time.

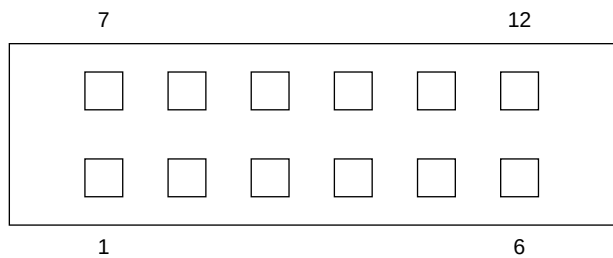


User-links select 1V8 or 3V3 digital I/O operating voltages and the digital IOREF voltages. The IOREF voltages have maximum current limits available at the board interface pins. Pmod boards typically only support 3V3.

For information on the user-links, and the maximum available IOREF currents, see [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47.

The following figure shows the Pmod connectors, J39, J42, J40, and J43.

Figure A-6: Pmod connectors



The following table shows the pin mapping for the Pmod 0 interface connector, J39.

Table A-12: Connector J39 (Pmod 0 interface) signal list

Pin	Signal	Pin	Signal
1	SH0_IO10	7	SH0_IO5
2	SH0_IO11	8	SH0_IO6
3	SH0_IO12	9	SH0_IO7
4	SH0_IO13	10	SH0_IO8
5	GND	11	GND
6	SH0_REF	12	SH0_REF

The following table shows the pin mapping for the Pmod 1 interface connector, J42.

Table A-13: Connector J42 (Pmod 1 interface) signal list

Pin	Signal	Pin	Signal
1	SH0_IO3	7	SH0_IO4
2	SH0_IO1	8	SH0_IO2
3	SH0_IO0	9	SH0_IO15
4	SH0_IO9	10	SH0_IO14
5	GND	11	GND
6	SH0_REF	12	SH0_REF

The following table shows the pin mapping for the Pmod 2 interface connector, J40.

Table A-14: Connectors J40 (Pmod 2 interface) signal list

Pin	Signal	Pin	Signal
1	SH1_IO10	7	SH1_IO5
2	SH1_IO11	8	SH1_IO6
3	SH1_IO12	9	SH1_IO7
4	SH1_IO13	10	SH1_IO8
5	GND	11	GND
6	SH1_REF	12	SH1_REF

The following table shows the pin mapping for the Pmod 3 interface connector, J43.

Table A-15: Connectors J43 (Pmod 3 interface) signal list

Pin	Signal	Pin	Signal
1	SH1_IO3	7	SH1_IO4
2	SH1_IO1	8	SH1_IO2
3	SH1_IO0	9	SH1_IO15
4	SH1_IO9	10	SH1_IO14
5	GND	11	GND
6	SH1_REF	12	SH1_REF

Related information

- [1.2 MPS4 board layout](#) on page 7
- [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43
- [A.2.2 Arduino Shield connectors](#) on page 83
- [A.2.4 Raspberry Pi HAT and board farm connectors](#) on page 89
- [Shield expansion, Pmod, and HAT expansion power](#) on page 105

A.2.4 Raspberry Pi HAT and board farm connectors

The 40-pin Raspberry Pi HAT connector (J41) is located between Shield 0 and Shield 1 on the MPS4 board. The 16-pin Raspberry Pi board farm connector (J5) is located in the middle of the MPS4 board.

Raspberry Pi HAT connector

The HAT interface shares I/O signals with the Shield 0/1 and Pmod0/1/2/3 interfaces. For more information on the shared signals, see [A.2.1 Arduino Shield, Pmod, and HAT shared SHx_IOx signals](#) on page 82.



The MPS4 board supports simultaneous use of Pmod, Shield, and HAT expansion. However, you must take care when driving the shared signals to ensure each signal is only used on one expansion interface at a time.

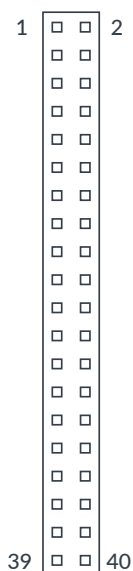


Connectors J33 and J38 select the 1V8 or 3V3 digital I/O operating voltages and the digital IOREF voltages. The IOREF voltages have maximum current limits available at the board interface pins. When using the HAT interface, you must set Shield 0/1 digital I/O voltages to the same voltage using J33 and J38.

For information on the user-links, and the maximum available IOREF currents, see [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47.

The following figure shows the Raspberry Pi HAT connector, J41.

Figure A-7: Raspberry Pi HAT connector



The following table shows the pin mapping for the HAT connector, J41.

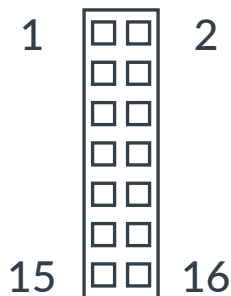
Table A-16: Connector J41 (HAT interface) signal list

Pin	HAT signal	Shared signal	FPGA pin	Pin	HAT signal	Shared signal	FPGA pin
1	3V3	-	-	2	5V	-	-
3	GPIO2	SH0_IO2	M12	4	5V	-	-
5	GPIO3	SH0_IO3	M11	6	GND	-	-
7	GPIO4	SH0_IO4	L15	8	GPIO14	SH0_IO14	J12
9	GND	-	-	10	GPIO15	SH0_IO15	J11
11	GPIO17	SH1_IO1	U11	12	GPIO18	SH1_IO2	T11
13	GPIO27	SH1_IO11	P15	14	GND	-	-
15	GPIO22	SH1_IO6	W15	16	GPIO23	SH1_IO7	P11
17	3V3	-	-	18	GPIO24	SH1_IO8	P12
19	GPIO10	SH0_IO10	K13	20	GND	-	-
21	GPIO9	SH0_IO9	V13	22	GPIO25	SH1_IO9	P14
23	GPIO11	SH0_IO11	K12	24	GPIO8	SH0_IO8	K14
25	GND	-	-	26	GPIO7	SH0_IO7	L11
27	GPIO0	SH0_IO0	M15	28	GPIO1	SH0_IO1	M13
29	GPIO5	SH0_IO5	L14	30	GND	-	-
31	GPIO6	SH0_IO6	L13	32	GPIO12	SH0_IO12	K11
33	GPIO13	SH0_IO13	J14	34	GND	-	-
35	GPIO19	SH1_IO3	R11	36	GPIO16	SH1_IO0	U12
37	GPIO26	SH1_IO10	R16	38	GPIO20	SH1_IO4	R12
39	GND	-	-	40	GPIO21	SH1_IO5	R13

Raspberry Pi board farm connector

The following figure shows the Raspberry Pi board farm connector, J5.

Figure A-8: Raspberry Pi board farm connector



The following table shows how the 16-way Raspberry Pi board farm connector (J5) maps to the 40-way Raspberry Pi HAT connector (J41).

Table A-17: 16-way board farm connector mapped to 40-way Raspberry Pi HAT connector

Pin	Farm signal	HAT signal	Description	Pin	Farm signal	HAT signal	Description
1	No connection	-	-	2	5V	-	-
3	CB_FARM0	GPIO2	Reserved 0	4	5V	-	-
5	CB_FARM1	GPIO3	Reserved 1	6	GND	-	-
7	CB_FARM2	GPIO4	Reserved 2	8	CB_FARM3	GPIO14	MCC UART, CB_URXD
9	GND	-	-	10	CB_FARM4	GPIO15	MCC UART, CB_UTXD
11	CB_FARM5	GPIO17	FPGA logic reset, CB_nPOR	12	CB_FARM6	GPIO18	FPGA CPU reset, CB_nRST
13	CB_FARM9	GPIO27	Runtime reset, nPBRST	14	GND	-	-
15	CB_FARM7	GPIO22	FPGA running, CB_RUN	16	CB_FARM8	GPIO23	Power on, nPBON

Related information

- [1.2 MPS4 board layout](#) on page 7
- [3.6.1 Arduino Shield, Pmod, and HAT interfaces](#) on page 43
- [A.2.2 Arduino Shield connectors](#) on page 83
- [A.2.3 Pmod connectors](#) on page 87
- [Shield expansion, Pmod, and HAT expansion power](#) on page 105

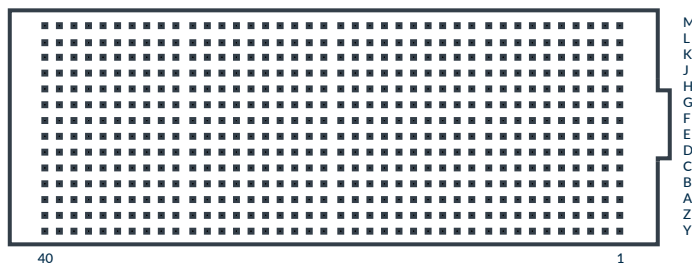
A.2.5 FMC+ HPC connector

The MPS4 board provides a 560-way Samtec SEARAY connector to support the *FPGA Mezzanine Card Plus* (FMC+) standard, *High Pin Count* (HPC) variant.

The connector supports the 400-way FMC HPC signals. The extra 160 FMC+ signals are not supported. The MPS4 board supports 1V2, 1V5, and 1V8 I/O for FMC+.

The following figure shows the FMC+ HPC connector, J11.

Figure A-9: FMC+ HPC connector



The pin mapping of the FMC+ HPC connector conforms to the *FPGA Mezzanine Card Plus (FMC+) Standard* (ANSI/VITA 57.4-2018). For the pin mapping of the FMC+ HPC connector, see <https://www.vita.com/>.

Arm supplies spreadsheets that describe the FPGA-FMC+ connectivity on the MPS4 board. For information on how to download the spreadsheets, see [FPGA-FMC+ pin connectivity](#) on page 49 .

Related information

[3.6.2 FMC+ HPC interface](#) on page 48

[FMC+ HPC power](#) on page 105

A.2.6 Samtec AcceleRate ARC6 connectors

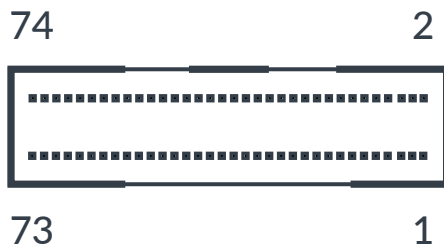
The MPS4 board has three Samtec AcceleRate ARC6 connectors. The ARC6 connectors give access to the *MultiGigabit Transceivers* (MGTS) on the FPGA, which you can use for board-to-board connectivity and the QSFP interface.

The three ARC6 connectors are:

- J49, MGT0 that uses lanes 1-11 from macros 125, 126, and 127
- J48, MGT1 that uses lanes 1-11 from macros 129, 130, and 131
- J50, MGT2 that uses lanes 1-11 from macros 227, 230, and 231

The following figure shows the ARC6 connectors, J49, J48, and J50.

Figure A-10: ARC6 connector



The following table shows the pin mapping for the ARC6 connectors, J49, J48, and J50.

Table A-18: ARC6 connectors signal list

Pin	Signal	FPGA pins J49/J48/J50	Pin	Signal	FPGA pins J49/J48/J50
2	GND	-	1	GND	-
4	MGT0/MGT1/MGT2_TXP0	BD42/AC40/AF7	3	MGT0/MGT1/MGT2_RXP0	BC45/AC45/AD2
6	MGT0/MGT1/MGT2_TXN0	BD43/AC41/AF6	5	MGT0/MGT1/MGT2_RXN0	BC46/AC46/AD1
8	GND	-	7	GND	-
10	MGT0/MGT1/MGT2_TXP1	BB42/AA40/AD7	9	MGT0/MGT1/MGT2_RXP1	BA45/AB43/AC4
12	MGT0/MGT1/MGT2_TXN1	BB43/AA41/AD6	11	MGT0/MGT1/MGT2_RXN1	BA46/AB44/AC3
14	GND	-	13	GND	-
16	MGT0/MGT1/MGT2_TXP2	AY42/W40/AB7	15	MGT0/MGT1/MGT2_RXP2	AW45/AA45/AB2

Pin	Signal	FPGA pins J49/J48/J50	Pin	Signal	FPGA pins J49/J48/J50
18	MGT0/MGT1/MGT2_TXN2	AY43/W41/AB6	17	MGT0/MGT1/MGT2_RXN2	AW46/AA46/AB1
20	GND	-	19	GND	-
22	MGT0/MGT1/MGT2_TXP3	AV42/U40/Y7	21	MGT0/MGT1/MGT2_RXP3	AU45/Y43/AA4
24	MGT0/MGT1/MGT2_TXN3	AV43/U41/Y6	23	MGT0/MGT1/MGT2_RXN3	AU46/Y44/AA3
26	GND	-	25	GND	-
28	MGT0/MGT1/MGT2_TXP4	AT42/T42/L5	27	MGT0/MGT1/MGT2_RXP4	AR45/W45/T2
30	MGT0/MGT1/MGT2_TXN4	AT43/T43/L4	29	MGT0/MGT1/MGT2_RXN4	AR46/W46/T1
32	GND	-	31	GND	-
34	MGT0/MGT1/MGT2_TXP5	AP42/P42/K7	33	MGT0/MGT1/MGT2_RXP5	AN45/U45/R4
36	MGT0/MGT1/MGT2_TXN5	AP43/P43/K6	35	MGT0/MGT1/MGT2_RXN5	AN46/U46/R3
38	GND	-	37	GND	-
40	MGT0/MGT1/MGT2_TXP6	AM42/M42/J5	39	MGT0/MGT1/MGT2_RXP6	AL45/R45/P2
42	MGT0/MGT1/MGT2_TXN6	AM43/M43/J4	41	MGT0/MGT1/MGT2_RXN6	AL46/R46/P1
44	GND	-	43	GND	-
46	MGT0/MGT1/MGT2_TXP7	AL40/K42/H7	45	MGT0/MGT1/MGT2_RXP7	AJ45/N45/M2
48	MGT0/MGT1/MGT2_TXN7	AL41/K43/H6	47	MGT0/MGT1/MGT2_RXN7	AJ46/N46/M1
50	GND	-	49	GND	-
52	MGT0/MGT1/MGT2_TXP8	AK42/H42/G5	51	MGT0/MGT1/MGT2_RXP8	AG45/L45/K2
54	MGT0/MGT1/MGT2_TXN8	AK43/H43/G4	53	MGT0/MGT1/MGT2_RXN8	AG46/L46/K1
56	GND	-	55	GND	-
58	MGT0/MGT1/MGT2_TXP9	AJ40/F42/F7	57	MGT0/MGT1/MGT2_RXP9	AF43/J45/H2
60	MGT0/MGT1/MGT2_TXN9	AJ41/F43/F6	59	MGT0/MGT1/MGT2_RXN9	AF44/J46/H1
62	GND	-	61	GND	-
64	MGT0/MGT1/MGT2_TXP10	AG40/D42/E5	63	MGT0/MGT1/MGT2_RXP10	AE45/G45/F2
66	MGT0/MGT1/MGT2_TXN10	AG41/D43/E4	65	MGT0/MGT1/MGT2_RXN10	AE46/G46/F1
68	GND	-	67	GND	-
70	MGT0/MGT1/MGT2_TXP11	AE40/B42/C5	69	MGT0/MGT1/MGT2_RXP11	AD43/E45/D2
72	MGT0/MGT1/MGT2_TXN11	AE41/B43/C4	71	MGT0/MGT1/MGT2_RXN11	AD44/E46/D1
74	GND	-	73	GND	-

The case on each ARC6 connector is connected to GND_EARTH with a further 4 pins.

Related information

[1.2 MPS4 board layout](#) on page 7

[3.6.4 Board-to-board interface and QSFP interface](#) on page 50

A.3 Display, camera, and audio connectors

The MPS4 board provides these display, camera, and audio connectors:

- 1 HDMI connector that provides digital video and digital audio to external displays (J2)

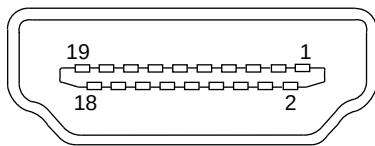
- 6 stereo jack 3.5mm connectors in 2 stacks of 3 (J3 and J4)
- 2 MIPI CSI 2-lane connectors (J46 and J47) and one 4-lane MIPI CSI connector (J44)
- 1 MIPI DSI 2-lane connector (J45)

A.3.1 HDMI connector

The HDMI 1.4a port provides digital video and digital audio to external displays.

The following figure shows the HDMI connector, J2.

Figure A-11: HDMI connector



The HDMI connector follows the standard 1.4a pin mapping available at <https://www.hdmi.org>.

Related information

[1.2 MPS4 board layout](#) on page 7

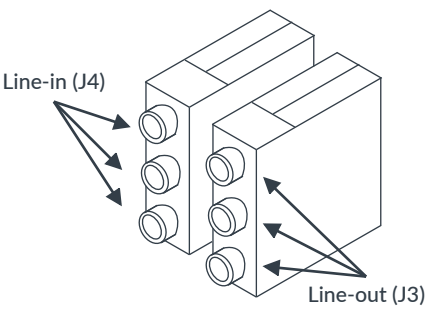
[3.5.1 HDMI display interface](#) on page 39

A.3.2 Audio connectors, stacked stereo jacks

The MPS4 board provides 6 stacked 3.5mm stereo jack connectors that connect to a 6-channel audio codec. The connectors provide line-level surround-sound input and line-level surround-sound output.

The following figure shows the stacked stereo jack connectors, J3 and J4, as viewed from the back of the board. The stacked stereo jack connectors on the left (J4) are line-in and the stacked stereo jack connectors on the right (J3) are line-out.

Figure A-12: Stacked stereo jack connectors



Related information

[3.5.2 Audio codec interface](#) on page 40

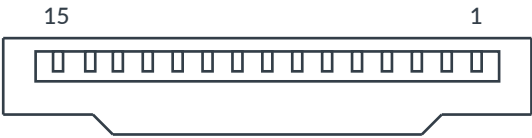
[1.2 MPS4 board layout](#) on page 7

A.3.3 MIPI DSI connector

The MIPI DSI connector (J45) is pin compatible with Raspberry Pi DSI connectors. The DSI connector supports 3V3 I/O.

The following figure shows the MIPI DSI connector, J45.

Figure A-13: MIPI DSI connector



The following table shows the pin mapping for the MIPI DSI connector, J45.

Table A-19: Connector J45 (MIPI DSI) signal list

Pin	Signal	FPGA pin	Pin	Signal
1	GND	-	16	No connection
2	DSIO_DATAN1	P21	17	
3	DSIO_DATAP1	R21	18	
4	GND	-	19	
5	DSIO_CLKN	N20	20	
6	DSIO_CLKP	P20	21	
7	GND	-	22	
8	DSIO_DATAN0	M23	23	
9	DSIO_DATAPO	N23	24	

Pin	Signal	FPGA pin	Pin	Signal
10	GND	-	25	
11	DSIO_SCL	J24	26	
12	DSIO_SDA	B23	27	
13	GND	-	28	
14	3V3	-	29	
15	3V3	-	30	

Related information

[1.2 MPS4 board layout](#) on page 7

[3.5.3 Display Serial Interface](#) on page 41

A.3.4 MIPI CSI connectors

The MPS4 board provides 2 MIPI CSI 2-lane connectors (J46 and J47), compatible with Raspberry Pi 15-pin CSI connectors, and 1 MIPI CSI 4-lane connector (J44). The CSI connectors support 3V3 I/O.

2-lane MIPI CSI connectors

The following figure shows the 2-lane MIPI CSI connectors, J46 and J47.

Figure A-14: MIPI CSI 2-lane connectors



The following table shows the pin mapping for the 2-lane MIPI CSI connector, J46.

Table A-20: Connector J46 (MIPI CSIO) signal list

Pin	Signal	FPGA pin	Pin	Signal
1	GND	-	16	No connection
2	CSIO_DATAN0	P22	17	
3	CSIO_DATAPO	R22	18	
4	GND	-	19	
5	CSIO_DATAN1	R23	20	
6	CSIO_DATAP1	T23	21	
7	GND	-	22	
8	CSIO_CLKN	M22	23	
9	CSIO_CLKP	N22	24	
10	GND	-	25	
11	CSIO_GP	A9	26	

Pin	Signal	FPGA pin	Pin	Signal
12	No connection	-	27	
13	CSIO_SCL	J22	28	
14	CSIO_SDA	M21	29	
15	3V3	-	30	

The following table shows the pin mapping for the 2-lane MIPI CSI connector, J47.

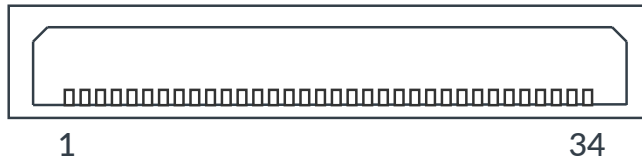
Table A-21: Connector J47 (MIPI CSI1) signal list

Pin	Signal	FPGA pin	Pin	Signal
1	GND	-	16	No connection
2	CSI1_DATAN0	A23	17	
3	CSI1_DATAPO	A24	18	
4	GND	-	19	
5	CSI1_DATAN1	C23	20	
6	CSI1_DATAP1	C24	21	
7	GND	-	22	
8	CSI1_CLKN	D24	23	
9	CSI1_CLKP	E24	24	
10	GND	-	25	
11	CSI1_GP	A8	26	
12	No connection	-	27	
13	CSI1_SCL	J21	28	
14	CSI1_SDA	M20	29	
15	3V3	-	30	

4-lane MIPI CSI connector

The following figure shows the 4-lane MIPI CSI connector, J44.

Figure A-15: MIPI CSI 4-lane connector



The following table shows the pin mapping for the 4-lane MIPI CSI connector, J44.

Table A-22: Connector J44 (MIPI connector, CSI2) signal list

Pin	Signal	FPGA pin
1	3V3	-
2	GND	-

Pin	Signal	FPGA pin
3	12V	-
4	PWR_EN	-
5	CSI2_SCL	C22
6	CSI2_SDA	L21
7	GND	-
8	CSI2_DATAP2	H24
9	CSI2_DATAN2	G23
10	GND	-
11	CSI2_DATAP0	H23
12	CSI2_DATAN0	H22
13	GND	-
14	CSI2_CLKP	G22
15	CSI2_CLKN	G21
16	GND	-
17	CSI2_DATAP1	E23
18	CSI2_DATAN1	E22
19	GND	-
20	CSI2_DATAP3	F21
21	CSI2_DATAN3	E21
22	GND	-
23	CSI2_GP3	A13
24	CSI2_GP2	A14
25	CSI2_GP1	A15
26	CSI2_GP0	A16
27	RST	-
28	PWRDN	-
29	GND	-
30	XCLK	-
31	GND	-
32	GND	-
33	GND	-
34	GND	-

Related information

[1.2 MPS4 board layout](#) on page 7

[3.5.4 Camera Serial Interfaces](#) on page 42

A.4 PCIe Switch connectors

The *PCI Express* (PCIe) card slots, USB 3.0, and Ethernet ports go through the PCIe Gen 3 Switch.

A.4.1 PCI Express connectors

The MPS4 board provides 2 PCIe x4 Gen 3 connectors. Each slot supports up to 25W output load. The MPS4 board also provides 2 4-pin PCIe power connectors that you can use for PCIe cards that require external power.

PCIe expansion slots



The maximum system power must not exceed the power supply rating.

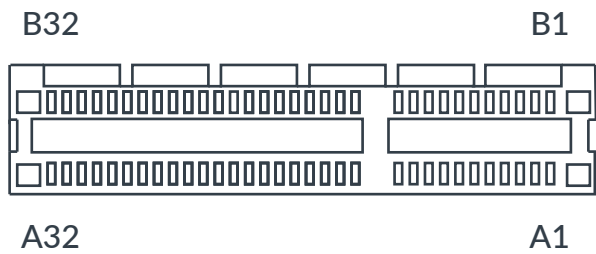
The following table shows the PCIe slots, J52 and J53.

Table A-23: PCIe expansion slots

Slot number	PCIe lane connector size	Used lanes	Unused lanes
PCIe slot 0, J52	x4	4	0
PCIe slot 1, J53	x4	4	0

The following figure shows the connectors J52 and J53.

Figure A-16: PCIe ×4 connectors



The following table shows the pin mapping for PCIe slot 0, connector J52.

Table A-24: PCIe slot 0, connector J52, signal list

Pin	Signal	Pin	Signal
B1	12V	A1	GND
B2	12V	A2	12V

Pin	Signal	Pin	Signal
B3	12V	A3	12V
B4	GND	A4	GND
B5	No connection	A5	No connection
B6	No connection	A6	PCIE_LOOP0
B7	GND	A7	PCIE_LOOP0
B8	3V3	A8	No connection
B9	No connection	A9	3V3
B10	3V3	A10	3V3
B11	PCIE_nWAKE0	A11	PCIE_nPERST0
B12	No connection	A12	GND
B13	GND	A13	SLOT0_CLKP
B14	SLOT0_PETP0	A14	SLOT0_CLKN
B15	SLOT0_PETN0	A15	GND
B16	GND	A16	SLOT0_PERP0
B17	nPRSNT0_HV	A17	SLOT0_PERN0
B18	GND	A18	GND
B19	SLOT0_PETP1	A19	No connection
B20	SLOT0_PETN1	A20	GND
B21	GND	A21	SLOT0_PERP1
B22	GND	A22	SLOT0_PERN1
B23	SLOT0_PETP2	A23	GND
B24	SLOT0_PETN2	A24	GND
B25	GND	A25	SLOT0_PERP2
B26	GND	A26	SLOT0_PERN2
B27	SLOT0_PETP3	A27	GND
B28	SLOT0_PETN3	A28	GND
B29	GND	A29	SLOT0_PERP3
B30	No connection	A30	SLOT0_PERN3
B31	nPRSNT0_HV	A31	GND
B32	GND	A32	No connection

The following table shows the pin mapping for PCIe slot 1, connector J53.

Table A-25: PCIe slot 1, connector J53, signal list

Pin	Signal	Pin	Signal
B1	12V	A1	GND
B2	12V	A2	12V
B3	12V	A3	12V
B4	GND	A4	GND
B5	No connection	A5	SLOT1_CKD0
B6	No connection	A6	SLOT1_CLKIP

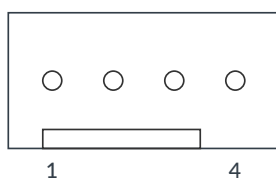
Pin	Signal	Pin	Signal
B7	GND	A7	SLOT1_CLKIN
B8	3V3	A8	SLOT1_CKD1
B9	No connection	A9	3V3
B10	3V3	A10	3V3
B11	PCIE_nWAKE1	A11	PCIE_nPERST1
B12	No connection	A12	GND
B13	GND	A13	SLOT1_CLKP
B14	SLOT1_PETP0	A14	SLOT1_CLKN
B15	SLOT1_PETN0	A15	GND
B16	GND	A16	SLOT1_PERP0
B17	nPRSNT1_HV	A17	SLOT1_PERN0
B18	GND	A18	GND
B19	SLOT1_PETP1	A19	No connection
B20	SLOT1_PETN1	A20	GND
B21	GND	A21	SLOT1_PERP1
B22	GND	A22	SLOT1_PERN1
B23	SLOT1_PETP2	A23	GND
B24	SLOT1_PETN2	A24	GND
B25	GND	A25	SLOT1_PERP2
B26	GND	A26	SLOT1_PERN2
B27	SLOT1_PETP3	A27	GND
B28	SLOT1_PETN3	A28	GND
B29	GND	A29	SLOT1_PERP3
B30	No connection	A30	SLOT1_PERN3
B31	nPRSNT1_HV	A31	GND
B32	GND	A32	No connection

PCIe power connectors

You can use the 2 4-pin PCIe power connectors, J75 and J76, for PCIe cards that require external power. Both power connectors provide 12V and 5V. The total maximum power for each PCIe slot is 25W.

The following figure shows the PCIe power connectors, J75 and J76.

Figure A-17: PCIe power connectors



The following table shows the pin mapping for the PCIe power connectors, J75 and J76.

Table A-26: PCIe power connectors pin mapping

Pin	Signal
1	5V
2	GND
3	GND
4	12V

Related information

[1.2 MPS4 board layout](#) on page 7

[3.7 PCI Express systems](#) on page 52

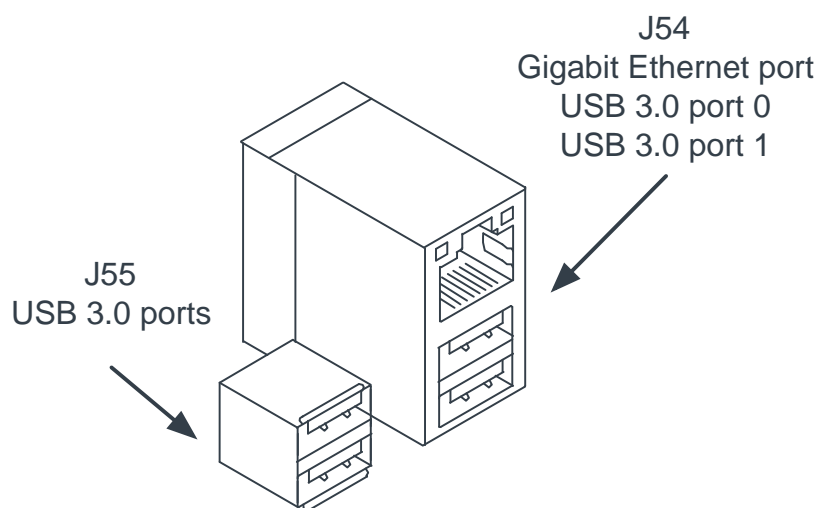
[PCIe power](#) on page 105

A.4.2 Ethernet and USB 3.0 connectors

The MPS4 board provides a combined Ethernet and dual USB 3.0 connector and a second dual USB 3.0 connector, totaling four USB 3.0 ports.

The following figure shows the combined Ethernet and dual USB 3.0 type-A connector (J54) and the second dual USB 3.0 connector (J55).

Figure A-18: Ethernet and USB 3.0 type-A connectors



Related information

[1.2 MPS4 board layout](#) on page 7

[3.7 PCI Express systems](#) on page 52

[USB power](#) on page 106

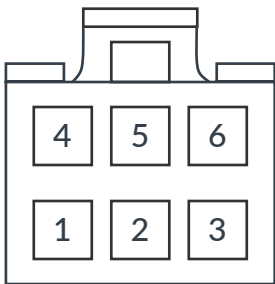
A.5 12V power connector

The MPS4 board provides a 12V DC C6P connector for connecting external power to the board. The external power supply unit provided is 252W (12V, 21A).

Connect the external mains power supply unit, which Arm supplies with the MPS4 board, to the power connector.

The following figure shows the 12V power connector, J71.

Figure A-19: 12V power connector



The following table shows the 12V power connector pin mapping, J71.

Table A-27: 12V power connector pin mapping

Pin	Signal	Pin	Signal
1	12V	4	GND
2	12V	5	GND
3	12V	6	GND

Related information

[3.3.3 Power](#) on page 32

[1.2 MPS4 board layout](#) on page 7

Appendix B Available power for expansion boards

The MPS4 board supplies power to the expansion boards or devices through the expansion connectors.

FMC+ HPC power

The MPS4 board supplies power, through the FMC+ HPC connector, to a fitted FMC+ HPC or FMC HPC board. The following table shows the maximum current that the board can supply from each power rail.

Table B-1: FMC+ HPC power

Power rail	Voltage	Max load	Comment
12P0V	12V	1A	Always ON
3P3VAUX	3V3 standby	20mA	For FMC EEPROM only
3P3V	3V3	2.5A	-
FMCVADJ	1V2/1V5/1V8	3A	Other voltages are not supported

Shield expansion, Pmod, and HAT expansion power

The MPS4 board supplies power to HAT, Shield, or Pmod expansion boards, depending on what expansion boards are fitted. The following table shows the maximum current that the board can supply from each power rail.



The I/O reference voltage is shared between the Shield, Pmod, and HAT interfaces, so ensure that the correct voltage is selected before powering up the system.

Table B-2: Shield, Pmod, and HAT power

Power rail	Max load	Comment
1V8/3V3	2A	Maximum current available for both Shields, all Pmod and HAT expansion boards. Includes digital I/O reference IOREF.
5V	2A	Maximum current available for both Shields, all Pmod and HAT expansion boards.



User-links on the board select 1V8 or 3V3 digital I/O operation. See [Selecting I/O voltage levels and reference voltages for expansion connectors](#) on page 47 for information on the digital I/O user-links.

PCIe power

The MPS4 board supplies power to devices connected through the PCIe slots.



The maximum system power must not exceed the power supply rating.

The following table shows the maximum current that the board can supply from each power rail.

Table B-3: PCIe power

Domain	Power rail	Max load	Comment
<ul style="list-style-type: none"> PCIe slot 0 PCIe slot 1 PCIe power connectors 	12V	2.1A	25W for each PCIe slot
PCIe power connectors	5V	2A	For PCIe cards that require external power only
<ul style="list-style-type: none"> PCIe slot 0 PCIe slot 1 	3V3	1.1A	For each PCIe slot

USB power

The MPS4 board supplies power to devices connected through the USB 3.0 ports. The following table shows the maximum current that the board can supply from each power rail.



Ensure the surge current of any connected devices does not exceed these limits. For example, take care when connecting mechanical hard drives that have high start-up currents.

Table B-4: USB power

Power rail	Max load
5V	1A for each port

Board farm interface power

The MPS4 board supplies power to farm boards connected to the farm interface. The following table shows the maximum current that the board can supply from each power rail.

Table B-5: Farm interface power

Power rail	Max load
5V	3A



The farm board 5V supply remains powered in Standby mode.

MIPI DSI and CSI power

The MPS4 board supplies power to DSI and CSI boards connected to the DSI and CSI. The following table shows the maximum current that the board can supply from each power rail.

Table B-6: Farm interface power

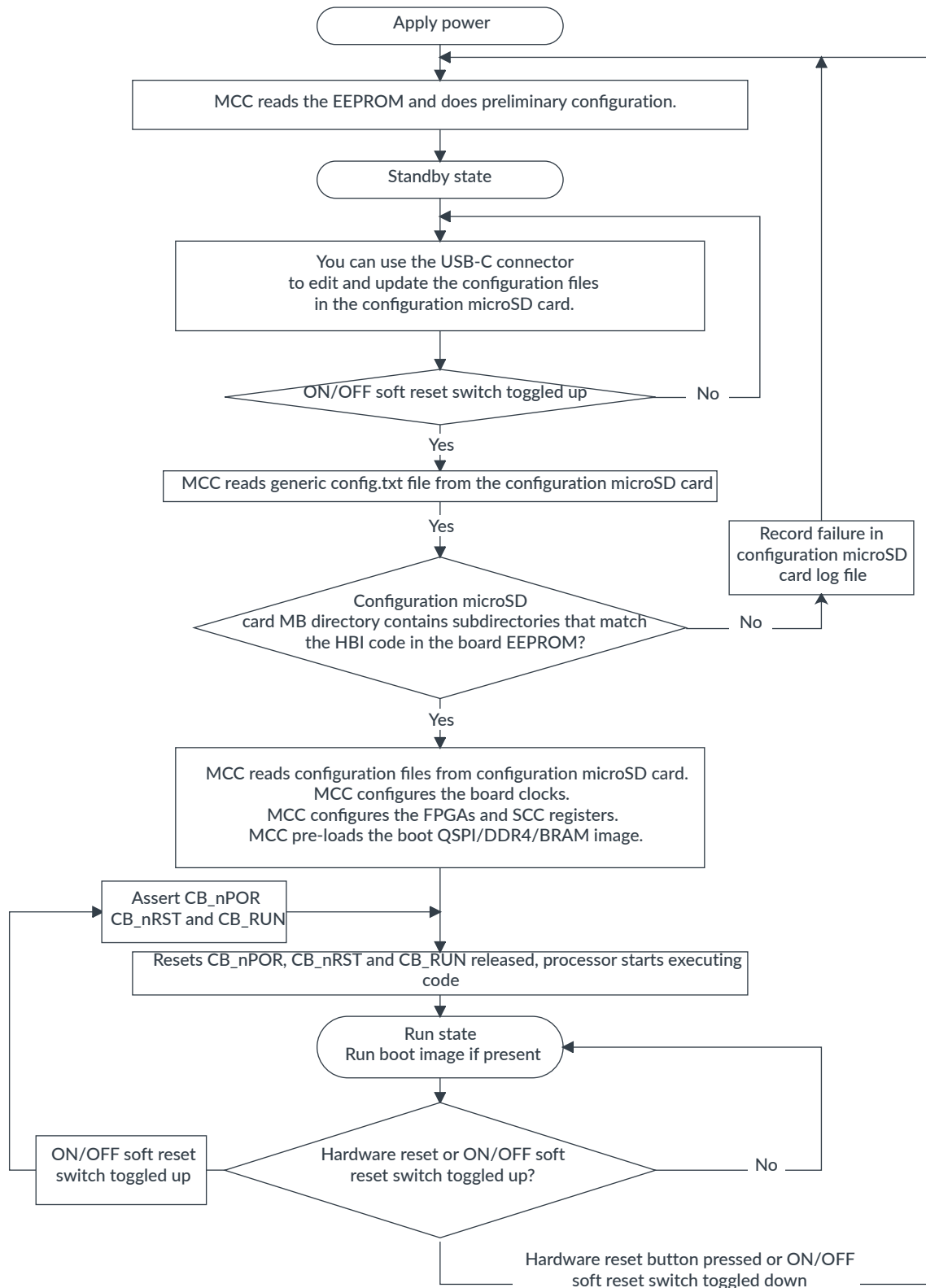
Domain	Power rail	Max load
DSI	3V3	1A
<ul style="list-style-type: none"> CSI0 CSI1 CSI2 	3V3	1A shared
CSI2	12V	1A

Appendix C Powerup and configuration sequence

The power push buttons and configuration files control the sequence of events of the board powerup and configuration process.

The following figure shows the detailed powerup and configuration sequence.

Figure C-1: MPS4 board powerup and configuration sequence





File names and directory names are in 8.3 short file name format:

- File names that you generate must be in lowercase.
- Directory names must be in uppercase.
- All configuration files must use DOS line endings, that is, 0x0D/0x0A, or \n and \r, or LF and CF.

The powerup and configuration sequence is:

1. The board applies power to the system.
2. The *Motherboard Configuration Controller* (MCC) powers the EEPROM and reads it to determine the HBI identification code for the board.
3. The system enters standby state.
4. The system enables the microSD memory card. You can connect a host computer to the USB-C connector to edit existing configuration files or drag and drop new configuration files.
5. The system stays in standby state until you toggle up the ON/OFF soft reset switch, labeled *PBON*, or type *reboot* on the MCC command-line interface.
6. The system loads the board configuration file:
 - The MCC reads the generic `config.txt` file.
 - The MCC searches the microSD card `MB` directory for the `HBI0376B` subdirectory that matches the HBI code in the board EEPROM.
7. The next steps depend on the configuration files:
 - If the MCC finds configuration subdirectories that match the HBI code of the board, configuration continues and the MCC reads the `board.txt` file.
 - If the MCC does not find the correct configuration files, it records the failure in a log file on the microSD card. Configuration stops and the system reenters the standby state.
8. The MCC measures the board power supplies.
9. The MCC configures the board clocks and FPGA SCC registers.
10. If the MCC finds new software images, it loads them into the QSPI flash, BRAM, or DDR4 memory through the FPGA.
11. The MCC releases the system resets, `CB_nPOR`, `CB_nRST`, and `CB_RUN`. The system enters the run state.
12. Normal operation continues until a new event occurs.

Appendix D Extra hardware required for selftests

FPGA images can come with a selftest program. To test the MPS4 board interfaces using the selftests, you need extra test equipment.

Mbed™ Application Shield

An Mbed™ Application Shield plug-in board is used to test the Shield interfaces. The board is available at <https://os.mbed.com/components/mbed-Application-Shield/>.

Table D-1: Mbed™ Application Shield

Peripheral	Interface	Description
RGB LEDs	GPIO	Test LED colors
Push buttons	GPIO	Test Shield push buttons
Temperature sensor	I ² C	Read temperature data
LCD	SPI	Initialize LCD and print characters using UART
Accelerometer	I ² C	Read accelerometer data
Joystick	GPIO	Read user input
Potentiometer	ADC	Data read through the MPS4 ADC
Speaker	GPIO	Test audio from speaker

Pmod QSPI module

A Digilent Pmod SF3 QSPI module is used to test the Pmod interfaces. The module is available at <https://digilent.com>.

Table D-2: Pmod QSPI module

Peripheral	Interface	Description
32 MB flash memory	SPI	SPI data interface
32 MB flash memory	GPIO	Hold and write protect

Raspberry Pi HAT CLCD and touch-screen module

An Adafruit Raspberry Pi HAT CLCD and touch-screen module is used to test a touchscreen attached to the MPS4 board. The module is available at <https://www.adafruit.com> (product code 2455).

The supplied unit requires the following modifications:

- Remove the right-angle 26-way connector, JP2.
- Fit a standard 40-way Raspberry Pi connector, available at <https://www.adafruit.com> (product code 2222).
- Fit R8 with a 10K resistor.

Table D-3: Adafruit Raspberry Pi HAT CLCD and touch-screen module

Peripheral	Interface	Description
CLCD	SPI	SPI data interface
Touchscreen	SPI	SPI data interface
Touchscreen	GPIO	Pen down interrupt

Audio codec loopback

Three 3.5mm stereo jack cables are used to loopback the six audio codec in/out channels. The part number for the cables is BC-A3MM003F, available at <https://digikey.co.uk>.

Table D-4: Audio codec loopback

Peripheral	Interface	Description
Audio codec	I ² S	Audio data interface
Audio codec	I ² C	Configuration interface

CSI ports

A Sony IMX477 camera module is used to test CSI ports 0 and 1. CSI port 2 uses a proprietary camera that is not available for external use. The part number for the CSI0/1 camera module is SC0818. The part number for the camera lens is SC0124, available at <https://thepihut.com>.

Table D-5: CSI ports

Peripheral	Interface	Description
CSI 0/1 camera	MIPI CSIs	Video data interface
CSI 0/1 camera	I ² C	Configuration interface

DSI port

The DSI display is a proprietary part and currently unavailable.

Table D-6: DSI port

Peripheral	Interface	Description
Display using the DSI port	MIPI 2-channel DSI	Video data interface

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Conformance notices

Federal Communications Commission Notice

This device is test equipment and so is exempt from part 15 of the FCC Rules under section 15.103 (c).

CE/UKCA Conformity

These marks indicate that this product meets all essential health, safety and environmental requirements. The CE mark indicates conformity within EU member states and the UKCA mark indicates conformity within the UK.

The Declarations of Conformity are available on request.



The Waste Electrical and Electronic Equipment (WEEE) marking, that is, the crossed out wheeled-bin figure, indicates that this product must not be disposed of with general waste within the European Union. To prevent possible harm to the environment from uncontrolled waste disposal, the user is required to recycle the product responsibly to promote reuse of material resources. To comply with EU law, you must dispose of the product in one of the following ways:

- Return it to the distributor where it was purchased. The distributor is required to arrange free collection when requested.
- Recycle it using local WEEE recycling facilities. These facilities are now very common and might provide free collection.
- If purchased directly from Arm, Arm provides free collection. Please email weee@arm.com for instructions.
- End-of-Life products can be disposed of safely using an Approved Authorized Treatment Facility (AATF). To support safe disposal, Arm has partnered with B2B Compliance. B2B can be contacted at the following weblink: <https://b2bcompliance.org.uk>

During the lifetime of the product, you are advised to:

- Inspect the product regularly to ensure that it is in good working order.
- Ensure that the product is free from dust and debris that might cause damage.
- Clean the product with an air duster when necessary.
- Observe the guidelines described in [1.1 Measures to ensure safety and prevent damage](#) on page 7.
- Power down the system when not in use.
- Observe ESD precautions when handling the product.

The product can radiate Radio Frequency Interference (RFI) or Electromagnetic Interference (EMI) and might cause harmful interference to radio communications. There is no guarantee that

interference cannot occur in a particular installation. If you suspect that this equipment is causing interference to other equipment, you are encouraged to try to correct the interference by one or more of the following measures:

- Ensure attached cables do not lie across any sensitive equipment.
- Increase the distance between the product and the receiver.
- Connect the equipment to an outlet on a circuit different from that to which the product is connected.
- Consult Arm for help.

The product can be sensitive to Radio Frequency Interference (RFI) or Electromagnetic Interference (EMI) which might cause incorrect operation of the product:

- Avoid using the product near sources of EMI.
- Never use the product in Safety-Critical-Systems (SCS), or Life-Critical-Systems (LCS).



Arm recommends that, wherever possible, shielded interface cables be used.

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0000-02	19 July 2024	Non-Confidential	Second release
0000-01	15 June 2022	Confidential	First release

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order.

Table 2: Differences between issue 01 and issue 02

Change	Location
First Non-Confidential release	-
Editorial updates	Whole document
Added chapter	2. Getting started on page 11
Corrected the number of on-board user components	<ul style="list-style-type: none"> 3. Hardware description on page 20 3.6.3 On-board user components on page 49

Change	Location
Corrected data rate of USB FIFO bridge	<ul style="list-style-type: none"> 3. Hardware description on page 20 3.9.2 Debug over USB on page 57
<ul style="list-style-type: none"> Removed OSC32K Updated OSCCLK destinations Updated typical uses for OSC3 and OSC4 Updated CFG_CLK frequency 	3.3.1 Clocks on page 27
Added example design	3.6.4 Board-to-board interface and QSFP interface on page 50
Updated PCIe controller and GbE controller IP	3.7 PCI Express systems on page 52
Updated file examples	<ul style="list-style-type: none"> 4.2.1 config.txt board configuration file on page 63 4.2.2.1 board.txt file on page 64 4.2.2.2 FPGA implementation .txt file on page 65 4.2.2.3 images.txt file on page 67
Added section	4.4 Configuring remote farm access on page 70
Added section	4.5 Troubleshooting on page 74
Added section	A.3.1 HDMI connector on page 95
Added PCIe power connector description	A.4.1 PCI Express connectors on page 100
Updated sequence to show that after reboot the line joins the main flow before the MCC reads EEPROM	C. Powerup and configuration sequence on page 108
Added appendix	D. Extra hardware required for selftests on page 111

Table 3: Issue 01

Change	Location
First Confidential release	-

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
bold	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <pre>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></pre>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.



This information is important and needs your attention.



This information might help you perform a task in an easier, better, or faster way.



This information reminds you of something important relating to the current content.

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
<i>Arm® CoreSight™ Components Technical Reference Manual</i>	DDI 0314	Non-Confidential
<i>Arm® CoreSight™ Trace Memory Controller Technical Reference Manual</i>	DDI 0461	Non-Confidential
<i>Arm® Development Studio Getting Started Guide</i>	101469	Non-Confidential
MPS4 board information and resources	-	-
<i>ULINKplus User's Guide</i>	101636	Non-Confidential

Non-Arm resources	Document ID	Organization
<i>AMD DMA/Bridge Subsystem for PCI Express Product Guide</i>	PG195	https://www.amd.com
Arduino Shield information and resources	-	https://www.arduino.cc
<i>Digilent Pmod Interface Specification</i> , Revised October 28, 2020, Version 1.3.1	-	https://www.digilent.com
<i>FPGA Mezzanine Card (FMC) Standard (ANSI/VITA 57.1-2019)</i>	ANSI/VITA 57.1-2019	https://www.vita.com
<i>FPGA Mezzanine Card Plus (FMC+) Standard</i>	-	https://www.vita.com
<i>IEEE Standard for Ethernet (IEEE 802.3)</i>	IEEE 802.3	https://www.ieee.org
<i>MIPI CSI-2 v3.0, MIPI Camera Serial Interface 2 (10-Sep-2019)</i>	-	https://mipi.org
<i>MIPI CSI v1.0, MIPI Camera Serial Interface (23-Mar-2004)</i>	-	https://mipi.org
<i>MIPI DSI v1.3.2, MIPI Display Serial Interface (23-Sep-2021)</i>	-	https://mipi.org
<i>PCI Express Card Electromechanical Specification Revision 3.0</i>	-	https://www.pcisig.com
Raspberry Pi HAT information and resources	-	Raspberry Pi Foundation, https://github.com/raspberrypi/hats
<i>Universal Serial Bus 3.0 Specification</i>	-	https://www.usb.org
AMD Virtex Ultrascale+ VU13P FPGA information and resources	-	https://www.amd.com