Using High-Speed Debug Port with SmartLynq+ Module on ADM-VB630

Introduction

This application note provides guidance on using the High-Speed Debug Port (HSDP) on the ADM-VB630. It includes instructions for configuring the HSDP in Vivado and using it to obtain high-speed debug access to the device. An example based on the ADM-VB630 Processing System (PS) Base SDK is used for demonstration. This example can be downloaded from the Alpha Data website.

HSDP Overview

The HSDP is a high-speed debug feature available in high-end AMD devices. It enables high-speed bidirectional debug and trace operations over a Gigabit Transceiver (GT) channel. In general, Versal devices provide three high-speed pathways to and from the Debug Packet Controller (DPC) through the GT channel. Access to this GT channel can be provisioned via an integrated Aurora 64B/66B block in the PS (hard Aurora interface), via CPM/PCIe, or through a soft Aurora IP instantiated in the Programmable Logic (PL).

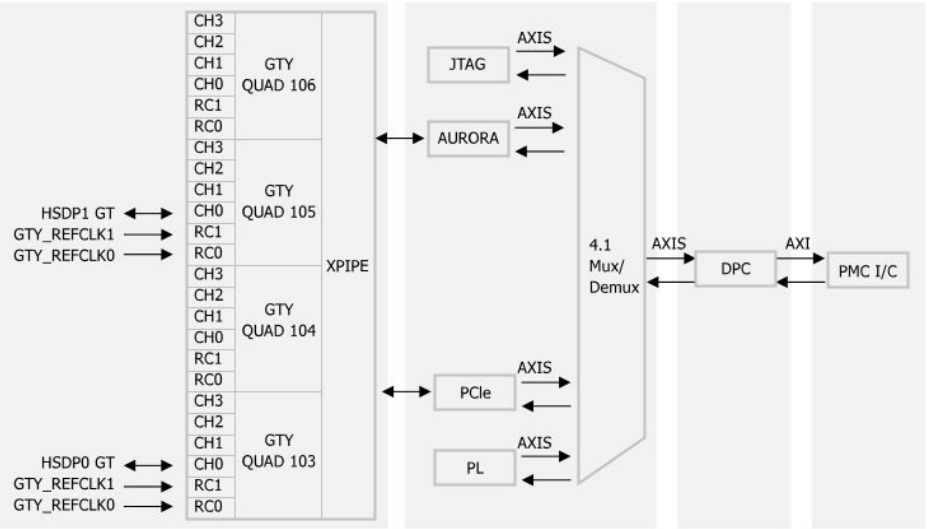

Figure 1 : HSDP pathways to the DPC

HSDP Over USB-C Connection

The USB-C connector provides 10-Gb/s high-speed differential lanes and supports custom data-transfer protocols. HSDP support can therefore be added to a board by wiring the Versal ASoC dedicated HSDP GT ports to a USB-C connector. Consequently, the GT channel used for HSDP must be configured to operate at a maximum line rate of 10 Gb/s.

Further details on HSDP support over a USB-C connector can be found in HSDP Target Interface | SmartLynq+ Module User Guide (UG1514).

SmartLynq+ Module

The HSDP uses the USB-C connector solely as a physical interface; data transfer does not use the USB protocol. A custom bidirectional data-transfer protocol is provided by the AMD SmartLynq+ programming, debug, and trace module. The SmartLynq+/HSDP link enables bidirectional communication between the host and the DPC. The host connects via either a USB 3.0 or an Ethernet interface.

> **Note:**
> The SmartLynq+ module only supports a single USB-C cable orientation for HSDP capability. As such, a lit amber LED on the module is an indication that the cable is not connected or needs to be flipped.

Further details on the SmartLynq+ module can be found in the SmartLynq+ Module User Guide.

Figure 2 : SmartLynq+ module

HSDP Support on the ADM-VB630

Channel 0 of GTY Quad 104 on the device is multiplexed between the HSDP interface and channel 0 of the QSFP1 port. The position of switch SW1-1 on the ADM-VB630 Rear Transition Module (RTM) determines which port uses this GT channel. When SW1-1 is in the OFF position, the GT channel is routed to the USB-C port (HSDP); otherwise, it is routed to the QSFP1 port.

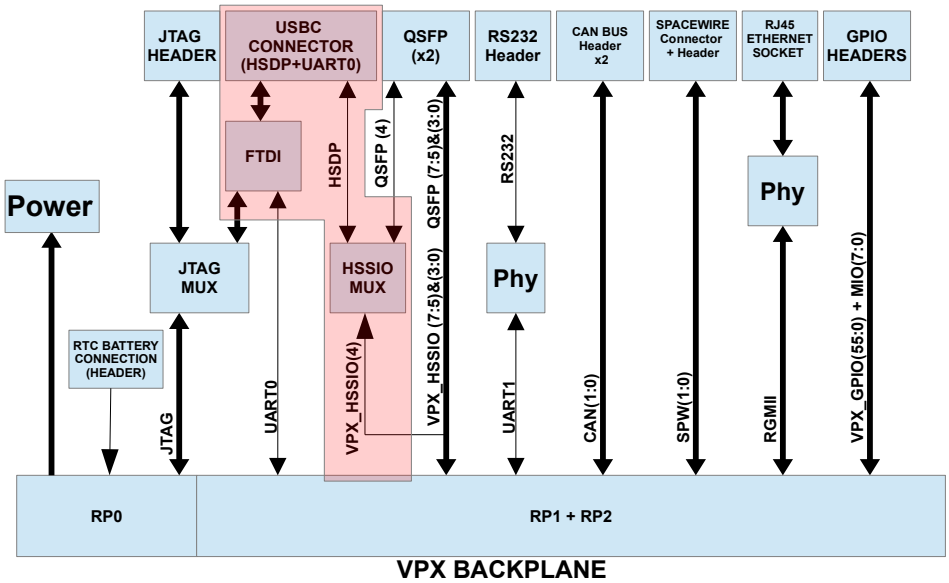

Figure 3 : HSDP interface on the ADM-VB630-RTM

Enabling the HSDP in Vivado

The HSDP is configured within the CIPS PSPMC configuration window. The ADM-VB630 PS Base Vivado project includes a block design containing the CIPS and will be used as the starting point for integrating HSDP support. To generate the base project, open Vivado and select **Tools** **Run Tcl Script...**. In the file selection dialog box, choose <project_root>/fpga/proj/base/mkxpr-base.tcl.

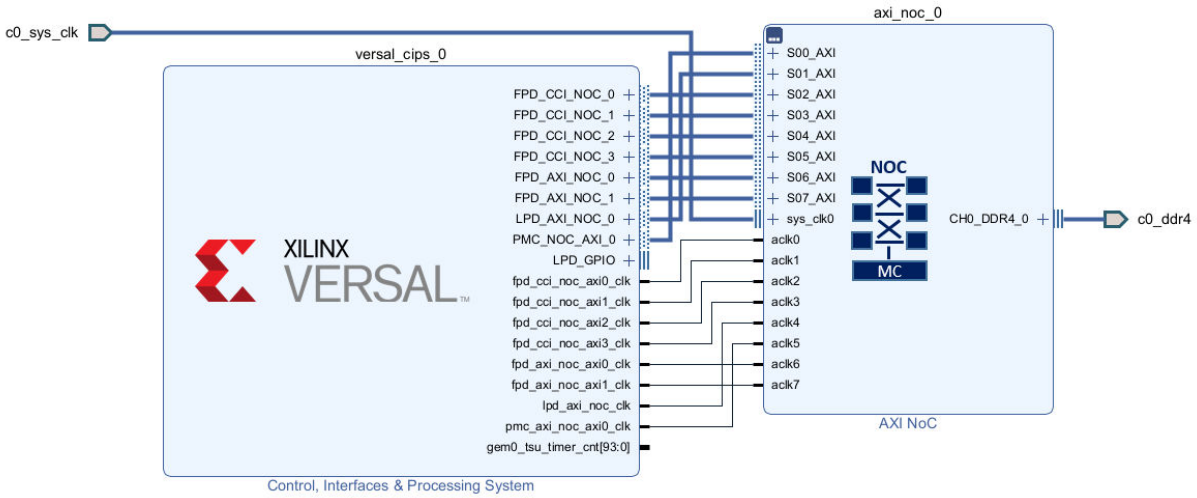

Figure 4 : ADM-VB630 PS Base design block diagram

Take the following steps to enable the HSDP:

1 Double-click the Versal CIPS IP core to recustomize it. Select **Next**, then select the blue box labeled **PS PMC**. In the resulting window and under the **PSPMC** tab, click **Debug** in the left pane.

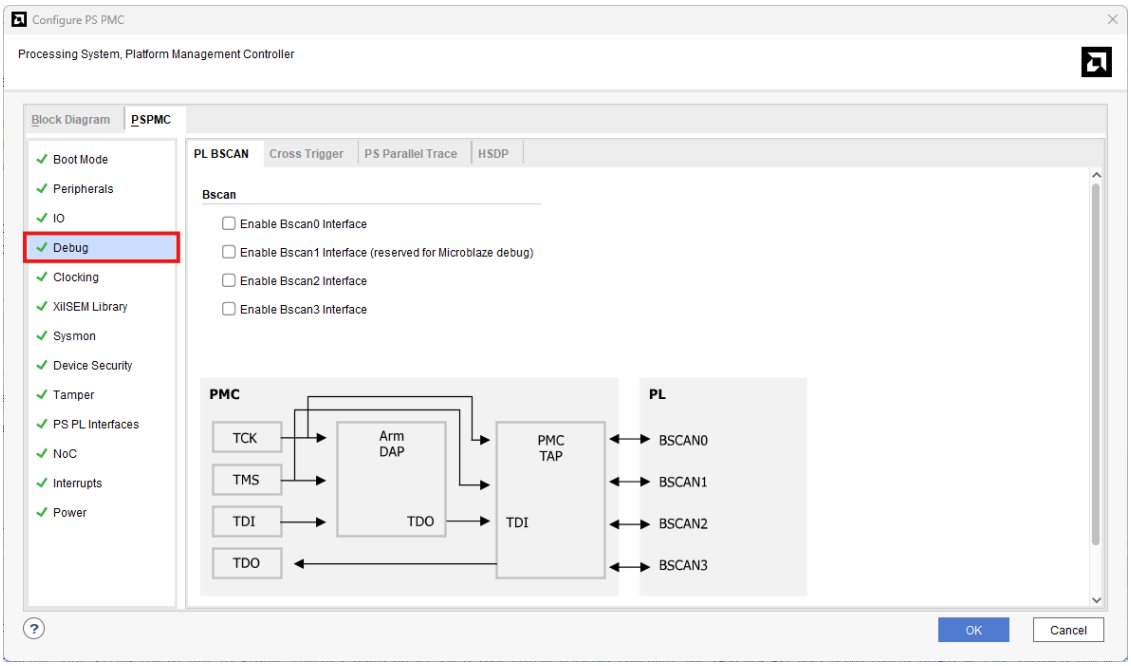

Figure 5 : Select the Debug Tab

2 Click on the **HSDP** tab and under **High-Speed Debug Port (HSDP)**, select **PL** as the **Pathway to/from Debug Packet Controller (DPC)**. The options available for the XCVE2302 device used in the ADM-VB630 are **JTAG** and **PL** as it does not have a hard Aurora core in the PS. The high-speed pathway will go through the PL in this case; thus, the choice of PL. Next, click **OK** and then **Finish** to save the changes.

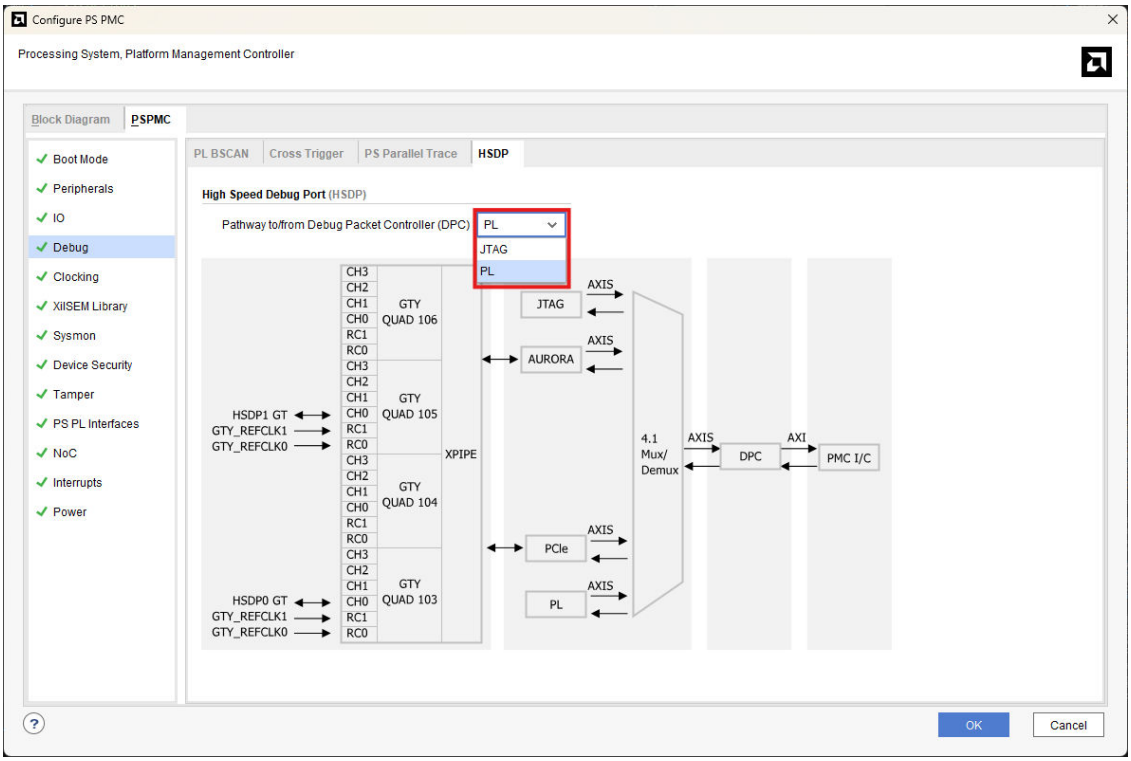

Figure 6 : Select the PL as pathway to/from the DPC

3 Once the HSDP is enabled, with "PL" set as the traffic pathway, three additional ports appear on the CIPS instance, namely: *S_AXIS_HSDP_INGRESS*, *S_AXIS_HSDP_EGRESS*, and the associated clock, *hsdp_ref_clk*. These need to be hooked up to a soft Aurora core connected to a GT interface via a GT Bridge IP, all in the PL.

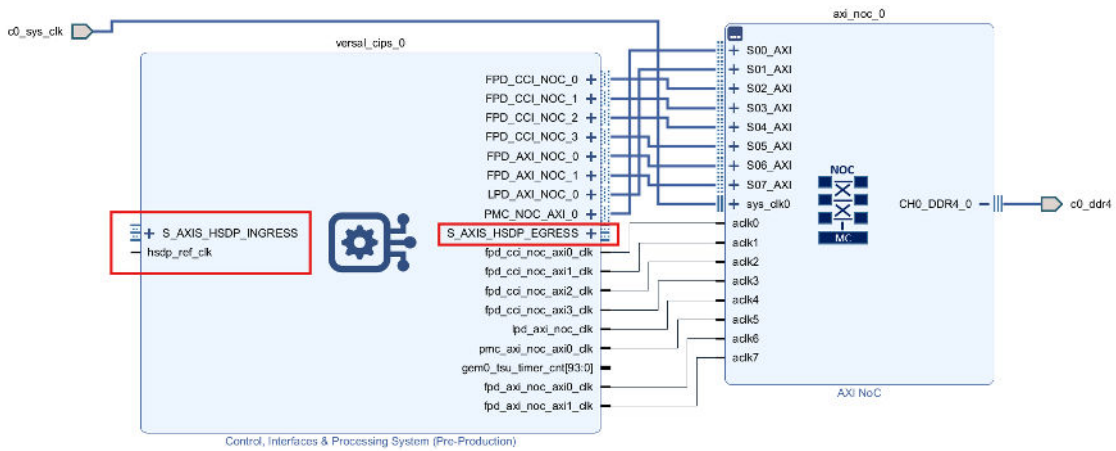

Figure 7 : HSDP ports on the CIPS

Example HSDP Project

The HSDP Vivado project contains a block design with the Versal ASoC CIPS IP configured to enable HSDP. This project is based on the ADM-VB630 PS Base project, so it also contains a NoC IP to give the PS access to the SDRAM. This enables PetaLinux to run in its own memory space.

The HSDP ingress, egress, and clock ports are then hooked up to a soft Aurora core (Aurora 64B66B) configured to 10 GB/s line rate and its TX and RX lanes connected to a GT Bridge IP (Versal ACAPs Transceivers Wizard). A 156.25MHz GT reference clock (PROG_CLK_1), routed through a clock buffer, is provided to the GT Quad from one of the programmable clocks on the board. In the example that accompanies this document, the soft Aurora design is adapted from the AMD "Versal HSDP Soft Aurora" example. This AMD example includes a VIO block that can be used to control and monitor the Aurora and GT Bridge IPs. This VIO is included in the ADM-VB630 HSDP example for verification of the GT link.

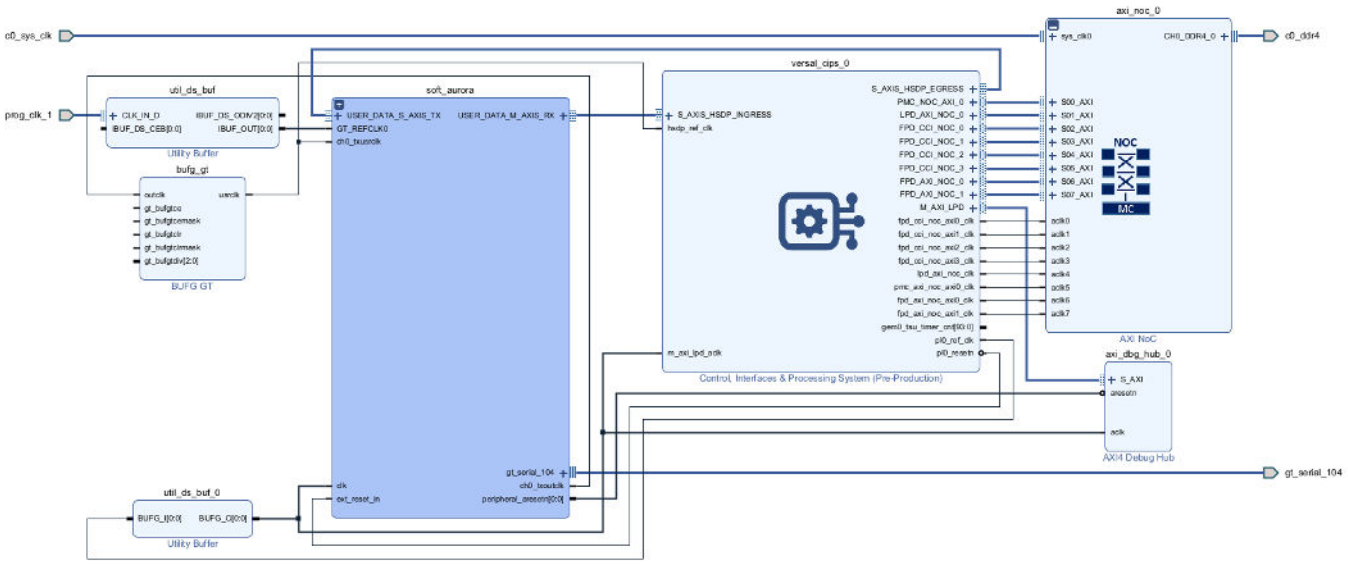

Figure 8 : HSDP design block diagram

The entire HSDP example project can be created and built from scratch using the Tcl scripts included in <project_root>/fpga/proj/hsdp/.

To generate the project, open Vivado and select **Tools Run Tcl Script...** from the main menu. In the file selection dialog box, choose <project_root>/fpga/proj/hsdp/mkxpr-hsdp.tcl. After the script completes, the project remains open and can be implemented or used to generate a device image in the usual way.

The project can also be generated from the command line by using Vivado in batch mode:

```
vivado -mode batch -source <project_root>/fpga/proj/hsdp/mkxpr-hsdp.tcl
```

> **Note:**
> The HSDP project sets the target device and configures the CIPS and NoC IPs through the ADM-VB630 board file. This is all scripted in the project creation Tcl.

To create the hardware description file, click on **Export-> Export Hardware-> Include device image** and name the file as **hsdp_admvb630_wrapper.xsa**. This will generate the .xsa file in the specified location. This .xsa file can then be used to build a PetaLinux image for the board. A PetaLinux flow for the ADM-VB630 can be found in the PS Base SDK documentation included in the example (<project_root>/doc/ad-ug-0221_v1_0.pdf).

Debugging with HSDP and SmartLynq+

The example here involves downloading a large binary file that contains a regular pattern of numbers into a section of the SDRAM using the HSDP. The downloaded file is then verified from the command line. The time to do this is compared with the time taken to download the same file over JTAG.

Preparing the Test Binary File

A python script is provided for the generation of a dummy biinary file. See <project_root>/test/gen_pattern.py. The file size is 16 MB by default. You can change this by editing the following line in gen_pattern.py

```
num_of_mb = 16
```

To generate the binary file (pattern.bin) and save it to the "<project_root>/test" folder, run the following coommands:

```
cd <project_root>/test
```

```
python gen_pattern.py
```

The pattern.bin file should be created inside "<project_root>/test/". You can open it with a Hex Editor to inspect.

test > pattern.bin

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded Text
00000000	31	31	31	31	31	31	31	31	32	32	32	32	32	32	32	32	1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
00000010	33	33	33	33	33	33	33	33	34	34	34	34	34	34	34	34	3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4
00000020	35	35	35	35	35	35	35	35	36	36	36	36	36	36	36	36	5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6
00000030	37	37	37	37	37	37	37	37	38	38	38	38	38	38	38	38	7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8
00000040	38	38	38	38	38	38	38	38	39	39	39	39	39	39	39	39	8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9
00000050	41	41	41	41	41	41	41	41	42	42	42	42	42	42	42	42	A A A A A A A A B B B B B B B B
00000060	43	43	43	43	43	43	43	43	44	44	44	44	44	44	44	44	C C C C C C C C D D D D D D D D
00000070	45	45	45	45	45	45	45	45	46	46	46	46	46	46	46	46	E E E E E E E E F F F F F F F F
00000080	31	31	31	31	31	31	31	31	32	32	32	32	32	32	32	32	1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2

Figure 9 : A section of the test file

Hardware Requirements

The following is a list of the required hardware:

- ADM-VB630 kit (ADM-VB630 board and RTM)
- VITA46 rack
- Micro SD card reader and micro SD card
- SmartLynq+ kit

Hardware Setup

1. Install the ADM-VB630 into the VITA46 rack.
2. Connect the USB-C cable from the SmartLynq+ kit between the ADM-VB630-RTM USB-C connector and the SmartLynq+ module.
3. Connect the SmartLynq+ to either Ethernet or USB. If using USB, see the SmartLynq+ Module User Guide for details on Windows/Linux USB 3.0 setup.
4. Copy the contents of the prebuilt/uSD directory to a blank SD card, formatted to FAT32.
5. Insert the SD card into the ADM-VB630. Take care with the delicate mechanism of the card holder.
6. Set SW2[1:8]=01010001 on ADM-VB630 to boot from the uSD card and with VPX IO enabled.
7. Set SW1-5=1 on the ADM-VB630 to enable VPX JTAG.

8 Set SW1-1=0 on the ADM-VB630-RTM to connect the channel 0 of the GTY Quaq 104 to the HSDP interface.

Figure 10 : Hardware setup

Preparing the SmartLynq+ Module

The SmartLynq+ can be used to remotely view the UART output from the ADM-VB630. Detailed instructions on setting this up can be found under the "Using the SmartLynq+ as a Serial Terminal" section of the System Design Example for High-Speed Debug Port with SmartLynq+ Module. The SmartLynq+ module has the Minicom application pre-installed, which can be used to connect directly to the UART on the ADM-VB630.

The following are the essential steps:

1 Power on the SmartLynq+ module and wait for it to finish booting up — an IP address will appear on its screen.

2 Using SSH, connect to it with the username: *xilinx*, password: *xilinx*, and its IP address.

```
ssh xilinx@<smartlynq+ ip>
```

3 Disable hardware flow control in the Minicom as it is not used on the ADM-VB630 UART.

```
echo "pu rtscts No" | sudo tee -a /etc/minicom/minirc.dfl
```

4 Connect to the ADM-VB630 serial terminal and leave it open

```
sudo minicom --device /dev/ttyUSB1
```

```
C:\>ssh xilinx@10.0.9.88
xilinx@10.0.9.88's password:
Last login: Thu Jan 26 13:21:50 2023 from 10.0.1.81
xilinx@slp-000A35080AC0 ~ $ echo "pu rtscts No" | sudo tee -a /etc/minicom/minirc.dfl
pu rtscts No
xilinx@slp-000A35080AC0 ~ $ sudo minicom --device /dev/ttyUSB1

Welcome to minicom 2.8

OPTIONS:
Compiled on Apr  1 2025, 23:03:08.
Port /dev/ttyUSB1, 08:13:47

Press CTRL-A Z for help on special keys
```

Figure 11 : SmartLynq+ Minicom

Booting the Board

1 Power on the ADM-VB630-RTM. You should see the board booting from the uSD card and the boot messages displayed on the Minicom terminal opened earlier.

```
xilinx@slp-000A35080AC0 ~ $ sudo minicom --device /dev/ttyUSB1

Welcome to minicom 2.8

OPTIONS:
Compiled on Apr  1 2025, 23:03:08.
Port /dev/ttyUSB1, 08:13:47

Press CTRL-A Z for help on special keys

[0.016]**************************************
[0.049]Xilinx Versal Platform Loader and Manager
[0.084]Release 2023.2   Oct 12 2023  -  15:51:06
[0.120]Platform Version: v2.0 PMC: v2.0, PS: v2.0
[0.162]BOOTMODE: 0x5, MULTIBOOT: 0xF0000000
[0.196]**************************************
[0.412]Non Secure Boot
[3.468]PLM Initialization Time
[3.497]***********Boot PDI Load: Started***********
[3.536]Loading PDI from SD1
[3.563]Monolithic/Master Device
[276.634]273.089 ms: PDI initialization time
[276.675]+++Loading Image#: 0x1, Name: lpd, Id: 0x04210002
```

Figure 12 : Boot messages in the Minicom terminal

2 Log in with user "petalinux". You will be asked to change the password. The PetaLinux command prompt will then be displayed.

```
[  OK  ] Mounted /run/media/mmcblk0p1.
[  OK  ] Finished OpenSSH Key Generation.

PetaLinux 2023.2+release-S10121051 hsdp-admvb630 ttyAMA0

hsdp-admvb630 login: petalinux
You are required to change your password immediately (administrator enforced).
New password:
Retype new password:
```

Figure 13 : First-time login into PetaLinux with the default user

Downloading the Test File Over HSDP

The Tcl script <project_root>/test/file_download.tcl can be used to download the test pattern.bin file into the SDRAM at address **0x40000000**. This address is chosen to ensure boot files and any system data on the primary memory are not inadvertently overwritten.

On the host system (the machine with access to the SmartLynq+ module), open the Vivado Tcl shell and issue the following command to download the test file using HSDP:

```
cd <project_root>/test

xsdb file_download.tcl <smartlynq+ ip> pattern.bin HSDP
```

Note:
It is assumed that xsdb is in your path/environment and available through the Vivado Tcl shell. If not, you can run XSDB using the full installation path, e.g.,"C:/Xilinx/Vivado/2023.2/bin/xsdb"

If the download is successful, you should have a screen similar to the following for a 16-MB test file:

```
Connected to ADM-VB630 over HSDP
Downloading test file
Time to download file via HSDP is 296767 microseconds (0.30s)
```

Figure 14 : HSDP file download performance

A simple verification of the file download can be done by dumping the memory location to which the test pattern file was just downloaded. Run the following in the Minicom terminal:

```
sudo hexdump -C /dev/mem -s 0x40000000 -n 512
```

```
hsdp-admvb630:~$ sudo hexdump -C /dev/mem -s 0x40000000 -n 512
40000000  31 31 31 31 31 31 31 31  32 32 32 32 32 32 32 32  |1111111122222222|
40000010  33 33 33 33 33 33 33 33  34 34 34 34 34 34 34 34  |3333333344444444|
40000020  35 35 35 35 35 35 35 35  36 36 36 36 36 36 36 36  |5555555566666666|
40000030  37 37 37 37 37 37 37 37  38 38 38 38 38 38 38 38  |7777777788888888|
40000040  38 38 38 38 38 38 38 38  39 39 39 39 39 39 39 39  |8888888899999999|
40000050  41 41 41 41 41 41 41 41  42 42 42 42 42 42 42 42  |AAAAAAAABBBBBBBB|
40000060  43 43 43 43 43 43 43 43  44 44 44 44 44 44 44 44  |CCCCCCCCDDDDDDDD|
40000070  45 45 45 45 45 45 45 45  46 46 46 46 46 46 46 46  |EEEEEEEEFFFFFFFF|
40000080  31 31 31 31 31 31 31 31  32 32 32 32 32 32 32 32  |1111111122222222|
```

Figure 15 : Verification of file download

Downloading the Test File Over JTAG

The same **file_download.tcl** script can be used to download the test file to SDRAM over JTAG through the APU. Remove the USB-C cable that connects the ADM-VB630-RTM to the SmartLynq+ module, and use a USB-C/USB-A adapter cable to connect the RTM directly to the host system. Then rerun the download command with **HSDP** changed to **JTAG**:

```
cd <project_root>/test
```

```
xsdb file_download.tcl <smartlynq+ ip> pattern.bin JTAG
```

If the download is successful, you should have a screen similar to the following for the same 16-MB test file:

```
Connected to ADM-VB630 over JTAG
JTAG frequency = 30000000
Downloading test file
Time to download file via JTAG is 14107358 microseconds (14.11s)
```

Figure 16 : JTAG file download performance

Performance Comparison

The file download over HSDP shows a considerable increase in speed, over 47x in this case, compared to JTAG at 30 MHz.

Useful Links

1. System Debug Planning | Versal Adaptive SoC System and Solution Planning Methodology Guide (UG1504)
2. ADM-VB630 User Manual (AD-UG-1540)
3. ADM-VB630-RTM User Manual (AD-UG-1541)
4. SmartLynq+ Module User Guide (UG1514)
5. System Design Example for High-Speed Debug Port with SmartLynq+ Module

Revision History

Date	Revision	Nature of Change
Mar 10, 2025	v1.0	Initial release

Address: 160 Dundee Street,
Edinburgh, EH11 1DQ, UK
Telephone: +44 131 558 2600
Email: sales@alpha-data.com
Website: http://www.alpha-data.com

Address: 10822 West Toller Drive, Suite 120,
Deer Creek Technology Center, Littleton, CO 80127
Telephone: (303) 954 8768
Email: sales@alpha-data.com
Website: http://www.alpha-data.com