



# Versal<sup>®</sup> Architecture and Subsystems for PCIe<sup>®</sup>

## CPM

- ▶ 將GT,NOC,DDR,PS,PL之間專門硬核化
- ▶ 另外多整合了兩個PCIE Controller

- ▶ 將GT,NOC,DDR,PS,PL之間專門硬核化
- ▶ 另外多整合了兩個PCIE Controller

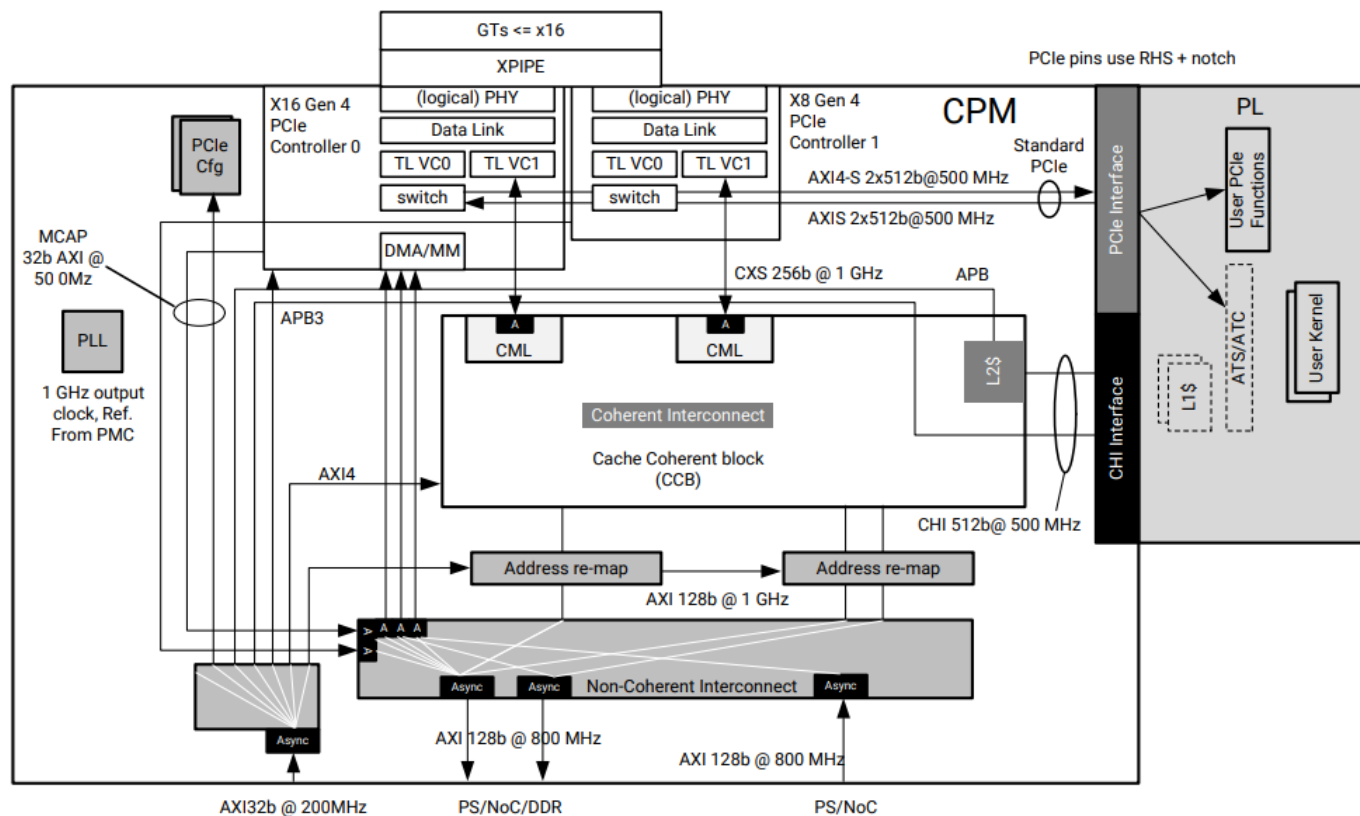


# Versal CPM Architecture

## CPM PCIe

- ▶ 具有較高的頻寬且整合了DMA
- ▶ 開機啟動速度比PL端的Pcie更快且 Address可以自動配置

Figure 3: CPM PCIe Block Diagram



X22777-101419

## Vivado Example Design

### Select Project Template

Select one of the below predefined templates on which to base your new project

---


**Templates**  

- ✓ Versal IBERT
- ✓ Versal CPM Tandem PCIe
- ▼ Zynq
  - ✓ Zynq UltraScale+ MPSoC Design Presets
  - ✓ Zynq-7000 Design Presets
- ▼ nonPI
  - ✓ BFT
  - ✓ CPU (HDL)
  - ✓ CPU (Synthesized)
  - ✓ Wavegen (HDL)
- ▼ PCIe
  - ✓ Versal CPM Bridge RP Design
  - ✓ Versal CPM QDMA EP Design
  - ✓ Versal CPM PCIe PIO EP Design
  - ✓ Versal CPM PCIe BMD EP Design
  - ✓ Versal CPM PCIe BMD EP Simulation Design
  - ✓ Versal CPM QDMA EP Simulation Design
  - ✓ **Versal CPM QDMA EP Design (Part based)**

Refresh
Catalog was last updated on 05/08/2024 5:20:22 PM

### Description

**Versal CPM QDMA EP Design (Part based)**  
This is CPM QDMA EP Design



< Back

Next >

Finish

Cancel

# Example Design

## CPM IP Setting

Configure CPM

Configure

Switch to Defaults

Configuration Options

CPM5 Basic Configuration

CPM5 PCIe Controller 1

Preset selection

Preset Custom

In Custom preset mode user can select all the available options.

PCIe Controller 0

Mode None

PCIe Controller 1

Mode DMALink Speed 32.0 GT/sLane Width X8

CPM5 Interfaces

☒ CPM to NoC 0☒ CPM to NoC 1☐ NoC to CPM☐ CPM to PL 0☐ CPM to PL 1

☒ Enable PCIe PIPE interfaces for simulation

Tandem Configuration

Tandem Mode None

Note:- Please refer to the relevant IP product guides ( [PG346](#) [PG347](#) ) based on the selected CPM5 mode for guidelines covering selection of GTYP quads, board layout and lane reversal requirements.

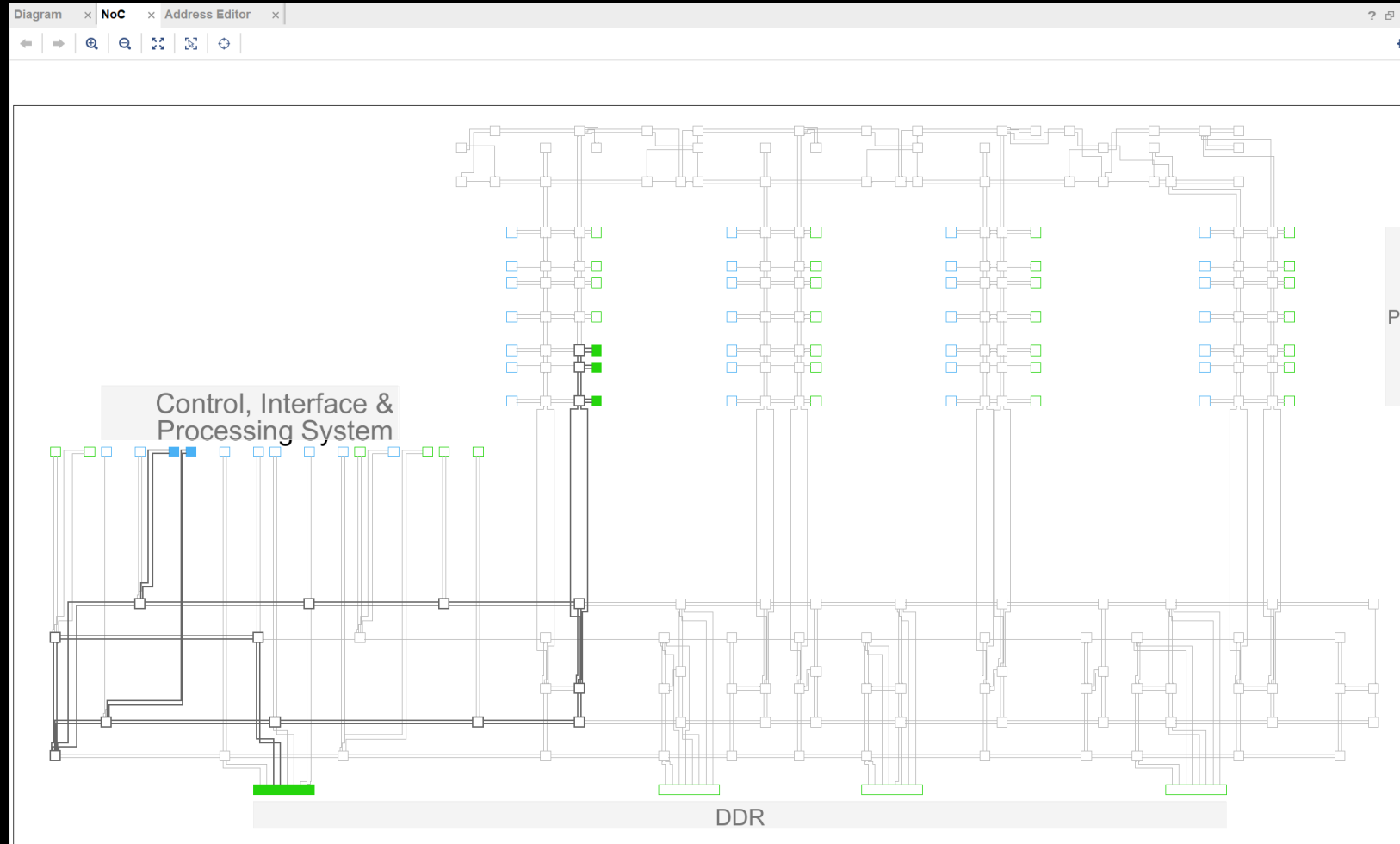
?

OK

Cancel

# Example Design

NOC



# XDMA IP Setting

Re-customize IP

DMA/Bridge Subsystem for PCI Express (4.1)

Documentation IP Location

☐ Show disabled ports

Component Name: xdma\_0

Basic PCIe ID PCIe : BARs PCIe : MISC PCIe : DMA

Functional Mode: DMA

Mode: Basic

Device / Port Type: PCI Express Endpoint device

PCIe Block Location: X0Y1

PCIe Interface

Lane Width: X1

Maximum Link Speed: 2.5 GT/s 5.0 GT/s 8.0 GT/s

Reference Clock Frequency (MHz): 100 MHz

AXI Interface

AXI Address Width: 64 [32 - 64]

AXI Data Width: 64 bit

AXI Clock Frequency: 62.5 125 250

DMA Interface option

AXI Memory Mapped AXI Stream

☐ AXI-Lite Slave Interface

☐ Enable PIPE Simulation

☐ Enable GT Channel DRP Ports

☐ Enable PCIe DRP Ports

☐ Additional Transceiver Control and Status Ports

sys\_clk

sys\_rst\_n

usr\_irq\_req[15:0]

M\_AXI

M\_AXI\_BYPASS

pcie\_mgt

user\_lnk\_up

axi\_aclk

axi\_aresetn

usr\_irq\_ack[15:0]

msix\_enable

Pcie bus

傳輸速度

系統時鐘

AXI bus

AXI clk

OK Cancel



# Install Host PC Driver

上電燒錄後，須確保FPGA不斷電的情況下重啟Ubuntu主機  
重啟後先安裝必須的依賴庫及套件

```
$ sudo -s  
$ apt-get update  
$ apt-get install build-essential  
$ apt-get install libaio-dev
```

下載Xilinx提供的dma driver

[https://github.com/Xilinx/dma\\_ip\\_drivers](https://github.com/Xilinx/dma_ip_drivers)

cd至以下目錄中，並編譯xdma driver

```
$ unzip dma_ip_drivers-master.zip # 解壓縮driver包  
$ cd dma_ip_drivers-master/QDMA/linux-kernel # cd進目錄中  
$ make clean  
$ make # build driver  
$ make install # install xdma driver  
$ sudo reboot
```



# Load Driver

Ubuntu重開後確認是否有成功辨識到Pcie Device

```
$ sudo -s # need to be root
$ lspci -vvd:b03f # note BDF e.g. "86:00.0" on 1st line
```

利用script生成對應的設定檔

```
$ dma_ip_drivers-master/QDMA/linux-kernel
$ cd scripts ; chmod 0755 *.sh
$ ./qdma_generate_conf_file.sh 0x86 1 1 0 # 利用Xilinx提供的script生成裝置的設定檔
$ cd ..
```

加載驅動

```
$ modprobe qdma-pf # 加載驅動
$ lspci -vvd:b03f # 再搜尋一次確認是否有成功加載
```

# Load Driver

使用官網提供的測試檔案

```
$ echo 8 > /sys/bus/pci/devices/0000\:86\:00.0/qdma/qmax # 使用之前生成的設定檔
```

添加序列

```
$ dma-ctl qdma86000 q add idx 0 dir bi # adds  
$ dma-ctl qdma86000 q start idx 0 dir bi # starts
```

讀寫測試序列

```
$ dma-to-device -d /dev/qdma86000-MM-0 -s 32 # PC寫入DMA  
$ dma-from-device -d /dev/qdma86000-MM-0 -s 32 # PC讀取DMA
```

停止序列

```
$ dma-ctl qdma86000 q stop idx 0 dir bi # stop  
$ dma-ctl qdma86000 q del idx 0 dir bi # deletes
```

