Speedster7t GDDR6 User Guide (UG091)

Speedster FPGAs

Preliminary Data

Achronix®
Data Acceleration

Copyrights, Trademarks and Disclaimers

Preliminary Data

This document contains preliminary information and is subject to change without notice. Information provided herein is based on internal engineering specifications and/or initial characterization data.

Achronix Semiconductor Corporation

2903 Bunker Hill Lane
Santa Clara, CA 95054
USA

Website: www.achronix.com
E-mail : info@achronix.com

Table of Contents

Chapter - 1: Introduction

The Speedster®7t FPGA family provides multiple GDDR6 subsystems enabling full utilization of the high-bandwidth efficiency of these interfaces for critical applications such as high-performance compute and machine learning systems. The number of GDDR6 subsystems varies with each Speedster7t FPGA. For example, the Speedster7t AC7t1500 FPGA provides eight GDDR6 interfaces (GDDR6 subsystems), four on the east side and four on the west side of the FPGA. Each subsystem comprises the GDDR6 controller and PHY hard cores and supports up to 512 Gbps. As a result, the Speedster7t AC7t1500 FPGA offers up to 4 Tbps of total bandwidth. The GDDR6 controller and PHY in the subsystem are implemented as hard IP blocks in the I/O ring of the Speedster7t FPGA. For the resource counts of other Speedster7t FPGA family members, refer to the *Speedster7t FPGA Datasheet* (DS015).

> **Note**
>
> The following sub-sections pertain to both the Speedster7t AC7t1500 and AC7t800 FPGAs.

Features

Each GDDR6 subsystem supports the following features:

- **Memory Density** – Supports GDDR6 devices from 8 Gb to 16 Gb, compliant with JEDEC GDDR6 SGRAM Standard JESD250.

  > **Note**
  >
  > There are subtle differences between Micron 8Gb and 16Gb x16 standard mode parts. The Micron 16Gb device is organized such that two WCK signals are needed for the 16 bit DQ bus. The 8Gb device, however, requires only one WCK signal. The test platform must be planned accordingly.

- **Data Rate** – Supports 12 Gbps, 14 Gbps and 16 Gbps data transfer rate per pin, delivering up to 512 Gbps per subsystem interface. As a result, the Speedster7t AC7t1500 FPGA with eight GDDR6 subsystems can deliver a total bandwidth of 4 Tbps for the entire device.
- **Memory Interface** – The GDDR6 subsystem consists of two separate channels, each providing a 16-bit interface. Hence, each subsystem provides a 32-bit interface to the external memory.
- **Controller Configuration** – Supports dual-controller configuration with an independent memory controller for each memory channel.
- **No Controller Bypass Support** – There is no option to bypass the GDDR6 controller or PHY.
- **System Configurable Modes** – The subsystem can be configured as either ×16 mode or ×8 clamshell mode for increased memory density applications. For example, in the Speedster7t AC7t1500 FPGA, in clamshell mode, the total memory capacity supported is 32GB.
- **ECC** – Only Speedster7t AC7t800 FPGAs support error check and correction of double-bit error detect and single-bit error correct functions.
- **Data Mask and Data Bus Inversion** – Supports GDDR6 data bus inversion (DBI) and command address bus inversion (CABI). Also, supports write double-byte mask and write single-byte mask operations.
- **CA and DQ format** – Double data-rate command address and data bus.

- **ZQ Calibration** – Supports multiple initiator/responder ZQ calibration.
- **AXI4 Interface** – Connects to the other IP interfaces within the Speedster7t FPGA or directly to the FPGA fabric via an AXI4 interface with support for full or half-rate clocking. The connections utilize either a 256-bit AXI4 interface to the 2D network on chip (2D NoC), which can run up to 1 GHz, or a 512-bit AXI4 direct-to-fabric interface, which can run up to 500 MHz.

Architecture Overview

The following table provides an overview of the different GDDR support items in the Speedster7t AC7t1500 and AC7t800 FPGAs:

Table 1: ***Speedster7t FPGA GDDR6 Support Overview***

GDDR6 Support Item	Speedster7t AC7t1500	Speedster7t AC7t800
Number of subsystems	8	3
Placement	GDDR6 0, 1, 2, 3 on the West and GDDR6 4, 5, 6, 7 on the East	All on the west
2D NoC Interface	All	All
DC Interface	GDDR6 1, 2, 5, 6	GDDR6 0, 1
Configuration mode	x16, x8 clamshell	x16, x8 clamshell
Data Rate	12, 14, 16Gbps	12, 14, 16Gbps
ECC	No	Yes

The following diagram details the architecture of the Achronix Speedster7t AC7t1500 FPGA. The eight GDDR6 subsystems are distributed four each on the east and west sides of the fabric. There are PLLs on the four corners of the device that supply the external reference clock to the GDDR6 SDRAM cores and other high-speed interfaces connecting with the 2D NoC over the FPGA fabric.

The GDDR6 subsystems can interface with the FPGA core in two ways:

1. 2D NoC interface – by using the network hierarchy that allows high-speed data flow between FPGA and peripheral interfaces. All the eight GDDR6 subsystems can be accessed from the FPGA fabric through the 2D NoC.
2. Direct connect interface (direct-to-fabric interface) – by using the DC interface that connects the memory controller directly to the core. There are only four GDDR6 subsystems (i.e., GDDR6 1, 2, 5 and 6 as shown in the following diagram) that connect to the FPGA fabric directly.

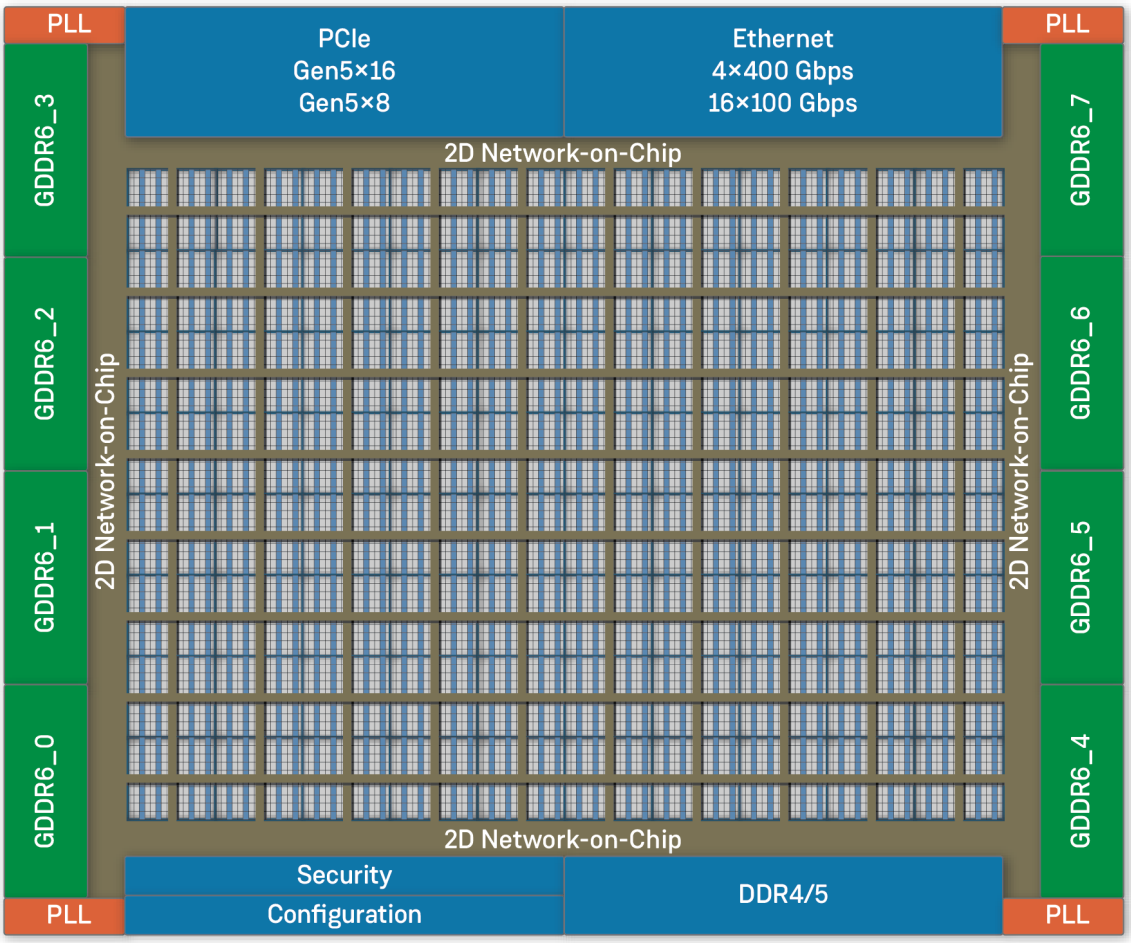


Figure 1: ***Speedster7t AC7t1500 FPGA Architecture Overview Block Diagram***

The following diagram details the architecture of the Achronix Speedster7t AC7t800 FPGA. There are three GDDR6 subsystems on the west side of the fabric. PLLs are situated on the northwest and southwest corner of the device that supply the external reference input clock to the GDDR6 SDRAM cores and other high-speed interfaces that connect with the 2D NoC over the FPGA fabric.

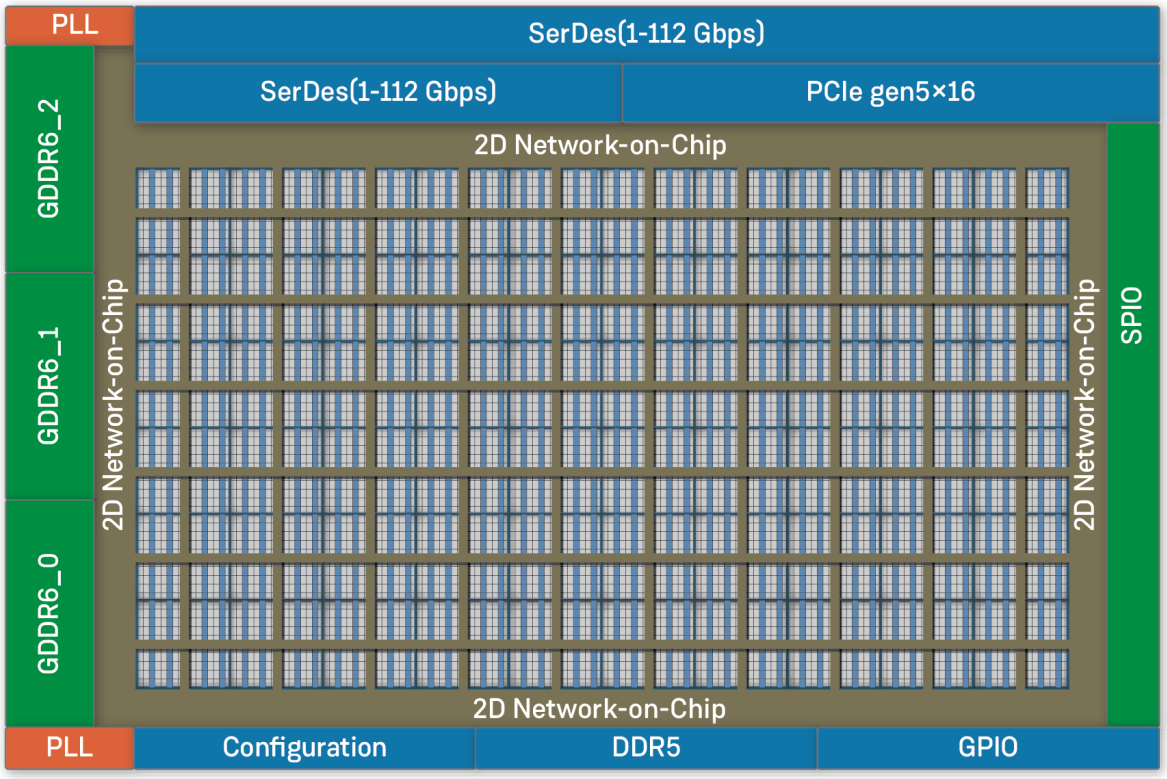


Figure 2: ***Speedster7t AC7t800 FPGA Architecture Overview Block Diagram***

GDDR6 Subsystem Overview

The GDDR6 subsystem provides a simple interface between off-chip GDDR6 memory components and the user logic mapped to the FPGA core. This memory subsystem is comprised of the following items that connect to the NoC and fabric:

- PHY IP
- Controller IP
- Clock and reset block
- APB interfaces
- AXI4 interfaces

The following is a block diagram of the GDDR6 subsystem:

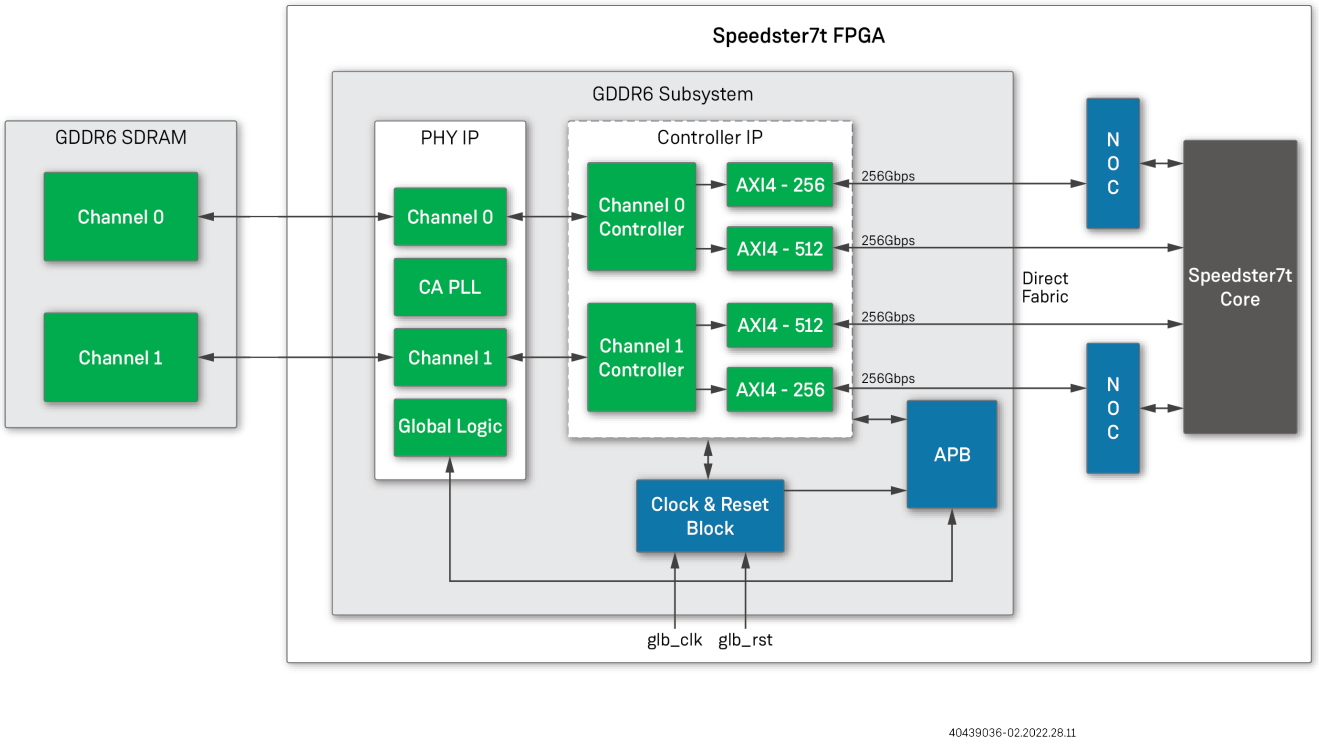


Figure 3: ***Speedster7t GDDR6 Subsystem Block Diagram***

The GDDR6 subsystem consists of the following functional blocks:

- **Clock and Reset** – The clock and reset block receives its input clocks from the on-chip PLLs and generates clocks to drive the GDDR6 memory controller and the PHY, with a maximum controller frequency of 1 GHz and a PHY clock frequency of 500 MHz. The command address clock runs at 2 GHz and the word clock (WCLK) at 8 GHz. This configuration generates data transactions at the maximum rate of 16 Gbps. The GDDR6 memory uses a double-data rate (DDR) protocol with separate data being latched at the rising and the falling edges of the clock. At reset, the controller performs the required initialization of the external memory, including calibration and programming of the internal mode registers.

- **Controller IP** – The controller IP consists of two channels, Channel 0 and Channel 1, and two controllers, one for each 16-bit channel of the GDDR6 memory. This configuration enables the two memory channels to operate completely independently. The controller IP uses the available AXI interfaces to either talk directly to the fabric or connect to it through the 2D NoC interface. On the other side, the controller is connected to the GDDR6 PHY via the DFI4.0 interface.The controller also has sub-modules such as read-modify-write, reorder, and the multi-port front-end cores. The memory controller performs writes and reads to/from the memory as follows:
  - Memory read – to perform a read, the user design signals a read request together with an address and burst size. The controller responds with an acknowledgement before the data is available. The controller translates each burst of data into multiple consecutive transactions.
  - Memory write – to perform a write, the user design signals a write request together with an address and burst size. When the GDDR6 memory is ready to receive the data, the controller generates and sends a data request to the PHY.
- **AXI4 Responder Interface** – The AXI4 responder interface is used in the memory subsystem to connect the controller to the FPGA fabric. This interface has two components: the 256-bit AXI4 interface that talks to the Speedster7t FPGA 2D NoC interface, and the 512-bit AXI4 interface that connects the signals from the controller directly to the user logic in the core through the DC interface.
- **PHY IP** – The GDDR6 memory PHY enables communication between the high-speed, high-bandwidth off-chip GDDR6 memory and the controller. The PHY supports two channels, each with a data width of 16 bits and speeds up to 16 Gbps per pin, delivering a maximum bandwidth of up to 64 GBps.
- **Memory Interface** – The GDDR6 PHY and the controller IP take care of all the details of the GDDR6 memory interface such as precharges, activates and refreshes. The controller issues commands as closely as possible, subject to the timing requirements of the GDDR6 memory, to achieve maximum efficiency.
- **APB Interface** – The APB interface operates at 250 MHz and allows configuration of the GDDR6 subsystem registers. The subsystem registers are configurable through the APB responder interface where the initiator can be from the fabric or FPGA configuration unit (FCU) through the 2D NoC. The FCU configures the subsystem registers during boot-up and these registers can be configured from the fabric during user mode.

Supported Frequency Table

The following table details the operating rates of each of the interfaces in the GDDR6 subsystem:

Table 2: ***Supported Range of GDDR6 Interface Frequencies***

Data Rate	AXI-256	AXI-512	Controller Clock	PHY Clock	Memory CA clock	Memory WCK
16 Gbps	1 GHz	500 MHz	1 GHz	500 MHz	2 GHz	DDR – 8 GHz QDR – 4 GHz
14 Gbps	875 MHz	437.5 MHz	875 MHz	437.5 MHz	1.75 GHz	DDR – 7 GHz QDR – 3.5 GHz
12 Gbps	750 Mhz	375 Mhz	750 Mhz	375 Mhz	1.5 GHz	DDR – 6 GHz QDR – 3 GHz

Chapter - 2: GDDR6 Controller Architecture

The Speedster®7t FPGA GDDR6 controller IP provides a high-performance interface to external GDDR6 SDRAM devices. The memory controller accepts read and write requests using a simple interface and translates these requests to the command sequences. The controller can automatically perform initialization and refresh functions and is also provided with programmable registers for all timing parameters and memory configurations that ensures compatibility with any valid GDDR6 subsystem integration.

The controller core interface is implemented as a queue so that new requests can be accepted on every clock cycle as long as the queue is not full. This construct allows the controller to look ahead into the queue to perform operations and precharges in advance to optimize throughput and efficiency.

The core uses bank management techniques to monitor the status of each memory bank. All banks can be managed simultaneously and can be opened or closed only when required, thus minimizing access delays. Read /write commands are issued with minimal idle time between commands, typically limited only by GDDR6 timing specifications. Proper bank management results in minimal delay between requests and enables higher memory throughput.

Controller Features

The following table provides a list of important GDDR6 memory controller features:

Table 3: *GDDR6 Controller Features*

Feature	Description
Maximum frequency	Supports GDDR6 operation at up to 16 Gbps.
Controller clock rate	Operates at half the rate of the command address clock.
Number of channels	Two independent channels (individual channels can also be disabled).
Data width	Supports GDDR6 ×16 mode or ×8 clamshell mode.
Queue depth	Reorder queue depth fixed to 64. Not configurable during run time.
Bank management	Bank management logic monitors status of each GDDR6 bank. Banks only opened or closed when necessary, minimizing access delays.
Bandwidth and latency optimization	Look-ahead logic monitors the user interface queue and examines access requests, issuing activate, precharge and auto-precharge commands as soon as possible to maximize memory bandwidth and minimize latency.
Write masks	Supports write single-mask and write double-mask operations.
Bus inversion	Supports GDDR6 data bus inversion (DBI) and CA bus inversion (CABI).

Feature	Description
Refresh	Per-bank and all-bank refresh support.
Auto-precharge	Read/write commands may be issued with or without auto-precharge.
Error detection	Supports GDDR6 error detection codes on the data bus for both read and write transfers. The memory device provides a checksum (CRC) per byte lane for any read or write data transfer allowing the controller to determine if the data transfer was completed correctly.
Error interrupt	Maskable interrupt outputs for all detected error conditions, with corresponding CSR read and clear-on-write registers.
Error retry	If the controller determines that a read or write data transfer error has occurred, the retry logic is enabled. The read or write request is retried, and the error detection and correction (EDC) results are rechecked until results are correct or the retry threshold has been exceeded.
Error status	Controller tracking of link error statistics such as retries and failures.

Controller Architecture Overview

The following figure details the memory controller and its sub-module cores:

- Multi-port front-end
- Reorder
- Read-modify-write
- Memory test
- Memory test analyzer

These blocks offer higher efficiency and throughput by re-ordering controller commands while also providing for test and debug capability.

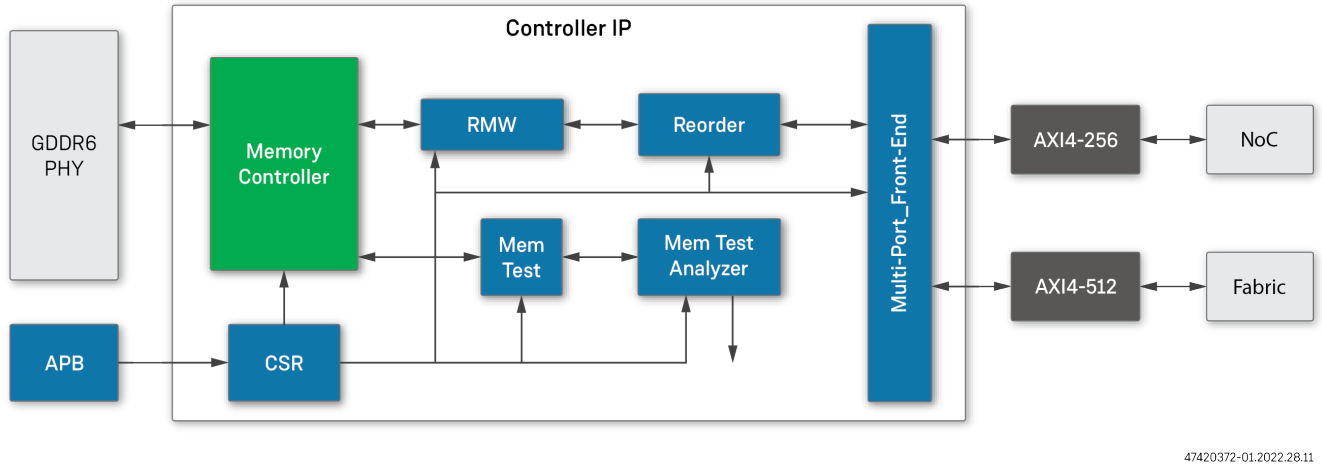


Figure 4: ***GDDR6 Controller IP Block Diagram***

The GDDR6 controller consists of the following functional blocks:

- **Multi-Port-Front-End (MPFE) Core** – The MPFE block provides a multi-port interface to connect to the controller channels.There are two MPFE ports per Channel 0 and Channel 1 controllers where one port is driven by the 2D NoC interface and the other is driven directly by the fabric.
- **Reorder Core** – This submodule is used in conjunction with the controller core to reorder user requests to the DRAM controller. Reordering can result in significant improvement of DRAM bus efficiency as it reduces bus idle times imposed by DRAM access rules. The reorder core can be parameterized to use different reorder criteria. This block can also be bypassed to maintain the original sequence of user requests. The optimal reorder criteria is chosen based on the nature of the requests coming from the user logic. The controller offers a queue depth of 64 for optimized performance.
- **Read Modify Write (RMW) Core** – The RMW submodule supports the address masking feature.
- **Memory Test Core** – The memory test core can be connected to the controller core to perform write and read operations to verify the integrity of the memory interface and memory devices. The core consists of different pattern generators to support standalone testing during board bring-up.
- **Memory Test Analyzer Core** – The memory test analyzer can compare the expected data with the read data and provide status to the user. The core also can be used to capture memory test signals of interest.
- **Memory Controller Core** – This queue-based, high-performance interface helps the controller perform a queue look-ahead in advance of upcoming commands to better optimize throughput and efficiency. The core also uses management techniques to monitor the status of each memory bank, including programmable registers for all timing parameters as well as memory configuration settings.

The controller also interfaces with the following functional blocks:

- **AXI4 Interface** – The AXI4 interface controller can access either the 2D NoC interface via the 256-bit AXI4 interface or connect directly to core fabric using the 512-bit AXI4 interface.
- **APB Interface** - There are four APB responders in the GDDR6 subsystem, one per controller and one for the PHY. The interface also includes a few register maps to enable clock and reset functions. The clock and reset of the APB responders are connected by CSR signals. The last APB responder is connected to IPCNTL components.

Modes of Operation

The Speedster7t FPGA GDDR6 controller supports two read/write channels, each with an independent memory controller. The GDDR6 subsystem supports the following two modes:

By 16 Mode

In this mode, each controller provides an interface to a single 16-bit memory channel. The following block diagram details the dual-controller system using a single memory device in ×16 mode:

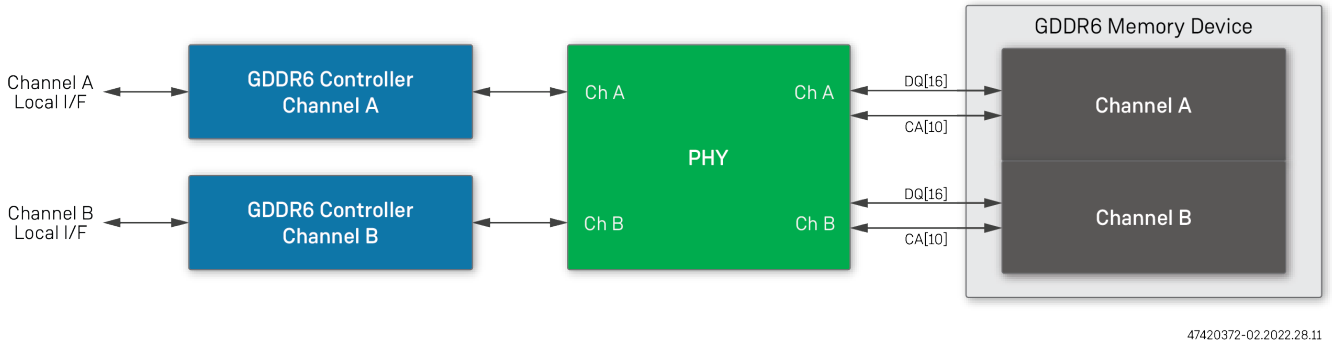


Figure 5: ***Dual-Controller ×16 Mode Block Diagram***

By 8 Clamshell Mode

The controller can also be configured in ×8 mode clamshell mode to address two memory devices. Clamshell mode provides a method to double the density of the system by sharing the same command/address bus between two devices in the system. The following block diagram details the configuration in a clamshell arrangement:

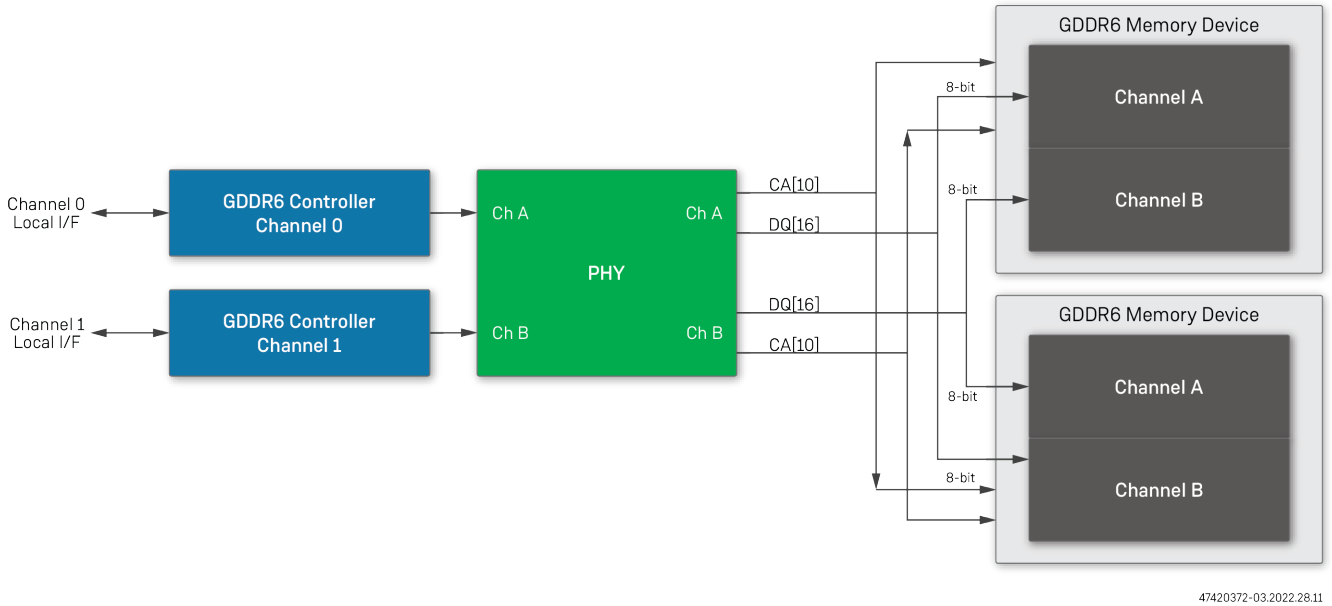


Figure 6: ***Dual-Controller ×8 Clamshell Mode Block Diagram***

Chapter - 3: GDDR6 PHY Architecture

PHY Overview

The embedded Speedster®7t FPGA GDDR6 PHY supports the GDDR6 memory standard at the channel interface and DFI-4.0 interface on the FPGA side with the memory controller. The PHY supports a maximum data rate of 16 Gbps and is targeted for systems that require low-latency and high-bandwidth memory solutions.

The PHY consists of two independent 16-bit channels, each composed of a modular command/address block (CA) and two data byte (DQ0 and DQ1) blocks. The following figure details the PHY interfacing with the off-chip GDDR memory on one side and with the memory controller on the FPGA side.

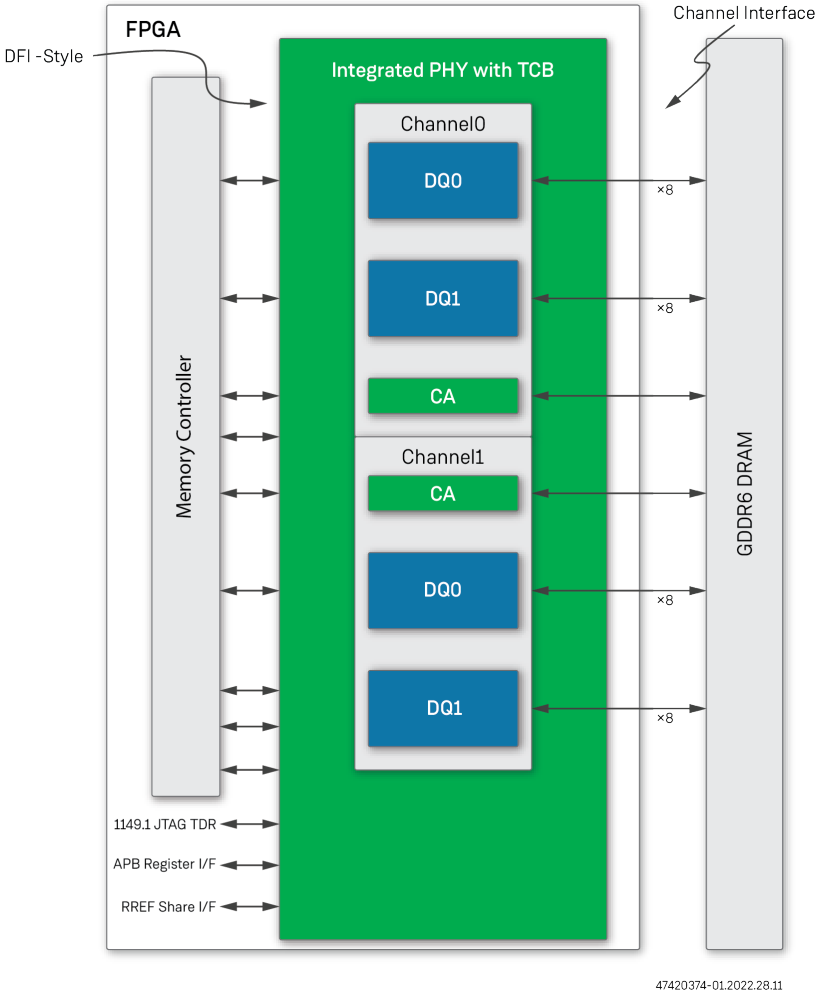


Figure 7: ***Speedster7t FPGA GDDR6 PHY Block Diagram***

PHY Features

Table 4: *Speedster7t FPGA GDDR6 PHY Features*

Feature	Description
DRAM density	SDRAM Density up to 16 Gb per component supported.
DRAM speeds	The Speedster7t AC7t1500 FPGA supports 12 Gbps, 14 Gbps and 16 Gbps data rates.
Number of channels	Two independent 16-bit channels. Individual channels can also be disabled.
GDDR PHY interface	DFI-style interface provided for PHY. The PCLK is a clock input to the PHY and PCLK:CK frequency ratio is fixed at 1:2
Command address bus inversion (CABI)	Supports CABI where each controller has a bit which enables CABI.
CA format	Double data rate (DDR) where data is latched on both edges of the clock.
CA serialization ratio	4:1 (corresponds to command address clock CK to PHY clock PCLK frequency ratio of 2:1).
CA driver impedance (RON)	40/48/60Ω.
CA termination	60/120/240Ω.
Data bus inversion (DBI)	Supports DBI where each DQ byte has a DBI bit.
DQ format	DDR and QDR based on WCLK.
DQ serialization ratio	16:1 (corresponds to PCLK to CK frequency ratio of 1:2 and QDR /DDR WCK mode).
DQ burst length	Supports a burst length of 16.
Receiver configuration	POD style receiver. Internal V_{REF} and DFE.
DQ driver impedance (RON)	40/48/60Ω.
Error detection code (EDC)	One EDC bit per DQ byte.

PHY Architecture

The PHY consists of the following blocks:

- Command/address (CA)
- DQ byte
- PLLs
- Global logic

There are three PLLs present in the entire PHY: one for the CA block and one for each of the two DQ words.

Command/Address Block

The command/address block executes the following operations:

- PHY configuration using DFI status interface.
- Serialization of commands and controls in the transmit data path. The controller provides the parallel data in the PCLK domain and that data is transmitted to the DRAM in CK domain.
- Per CA bus timing adjustment capability through CA training.
- Memory controller initiated update to interface for periodic driver impedance calibration
- PHY-initiated update interface for periodic training in PHY independent mode.

DQ Block

Each DQ block handles:

- Serialization of write data in the transmit data path. The controller provides the parallel data and write control signals in the PCLK domain. Data is then transmitted to DRAM based on the WCK frequency.
- De-serialization of read data in the receive data path. The memory controller provides the read control signals in the PCLK domain. Data is received from the DRAM with reference to WCK and passed on to the controller in the PCLK domain.
- Per-DQ eye timing adjustment for both transmit and receive paths.
- Read/write eye training and calibration such as WCK to CK.
- Per-pin internal V_{REF} generation and calibration.
- Registers for debug and control.

CA PLL

The CA PLL block handles high-speed clock (CK/CKN) generation using the PLL. CK/CKN is common for both channels of the DRAM. Different DRAM data rates are supported with appropriate PLL multiplier and post-divider ratios. Reference and DFI clocks are provided by the FPGA. The internal or local PCLK is aligned to the FPGA PCLK using a DLL.

DQ PLL

The DQ PLL (one per ×16 interface) handles high-speed clock (WCK/WCKN) generation using a PLL. WCK /WCKN is present per byte or per word as a per DRAM configuration. Different DRAM data rates are supported with appropriate PLL multiplier ratios and post-divider ratios. The PCLK is supplied by the FPGA to the PHY per DQ channel. The internal or local PCLK (LPCLK) is aligned to the FPGA PCLK using an embedded PHY DLL that is present per DQ byte channel. The global logic block handles the APB register interface from the memory controller and CA PLL configuration. Initialization and frequency changes are handled using the DFI status interface.

The following block diagram details the high-level PHY I/O:

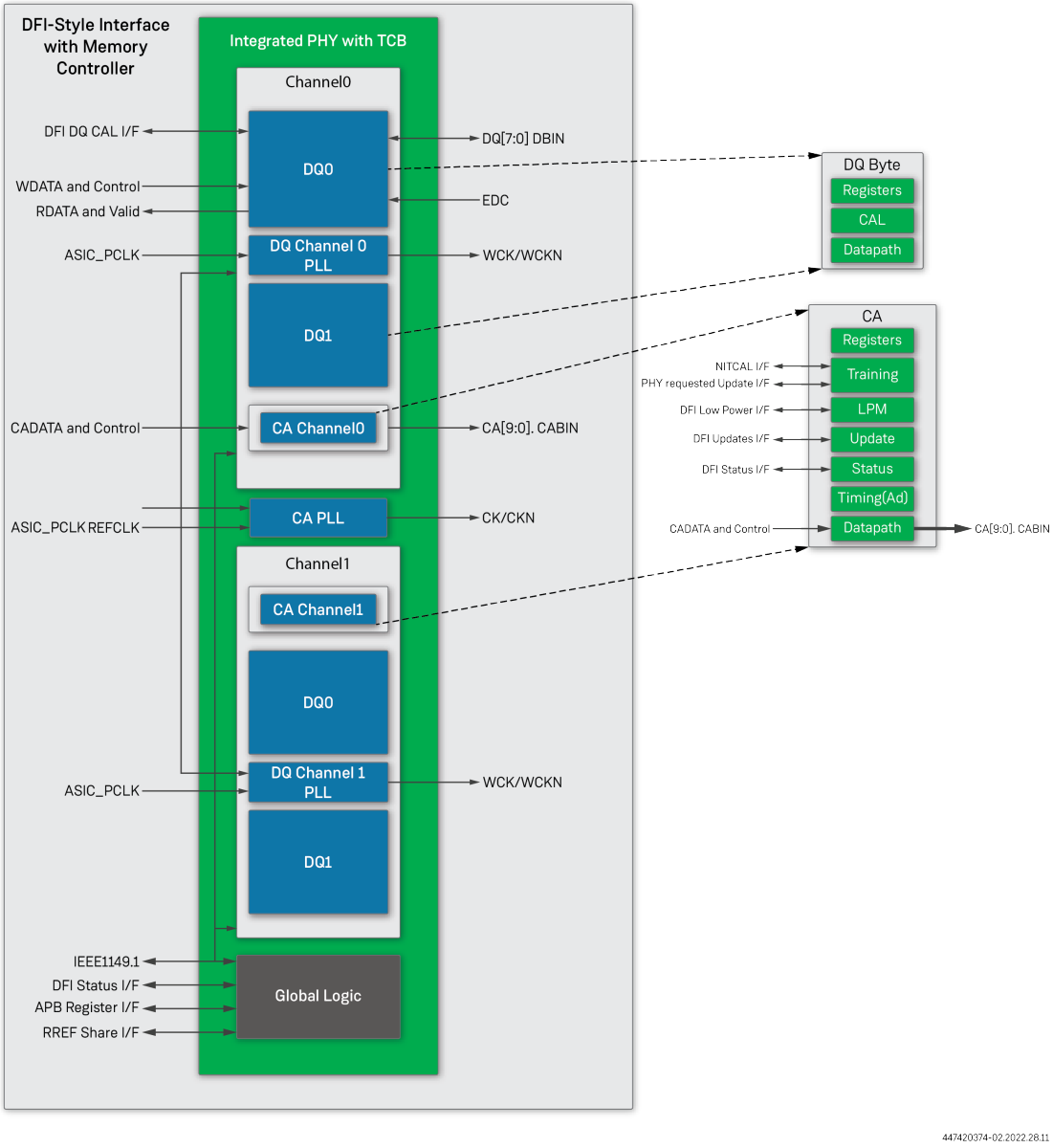


Figure 8: ***High-Level PHY I/O Block Diagram***

Chapter - 4: GDDR6 Clock and Reset Architecture

Overview

The Speedster®7t FPGA GDDR6 subsystem requires an external input reference clock and reset signals to drive the subsystem. The FPGA clock and reset generator module, consisting of PLLs, DLLs and reset circuitry, helps generate the required subsystem clock and reset signals at valid rates.

There is a clock and reset generator in every corner of a Speedster7t FPGA. Each clock and reset generator has four PLLs, with each PLL capable of generating up to four clock outputs. Each clock and reset generator can produce up to 16 clocks which can be routed to the global clock network. The GDDR6 subsystem has access to 32 global clocks from two adjacent clock generators. The subsystem clocks can be chosen from any of these 32 global clocks. Refer to the *Speedster7t Clock and Reset Architecture User Guide* (UG083) for further details.

The external input reference clock is applied to the FPGA PLL through the clock input/output pins. The PLL generates three input clocks required for the GDDR6 subsystem. These clocks can be sourced from either a single or multiple PLLs. The three clocks necessary for a GDDR6 design are:

1. GDDR6 memory controller and PHY reference clock.
2. Reference input clock for the 256-bit AXI4 2D NoC interface.
3. Reference input clock for the 512-bit direct-to-fabric-connect AXI4 interface.

Similarly, there are 32 global resets produced by the clock and reset generators. Also, there are an additional 48 active-low FPGA configuration unit (FCU) startup resets. Any of these 80 resets can be selected to provide the reset input to the GDDR6 subsystem.

The diagram below shows the clocks and resets required to drive a GDDR6 subsystem:

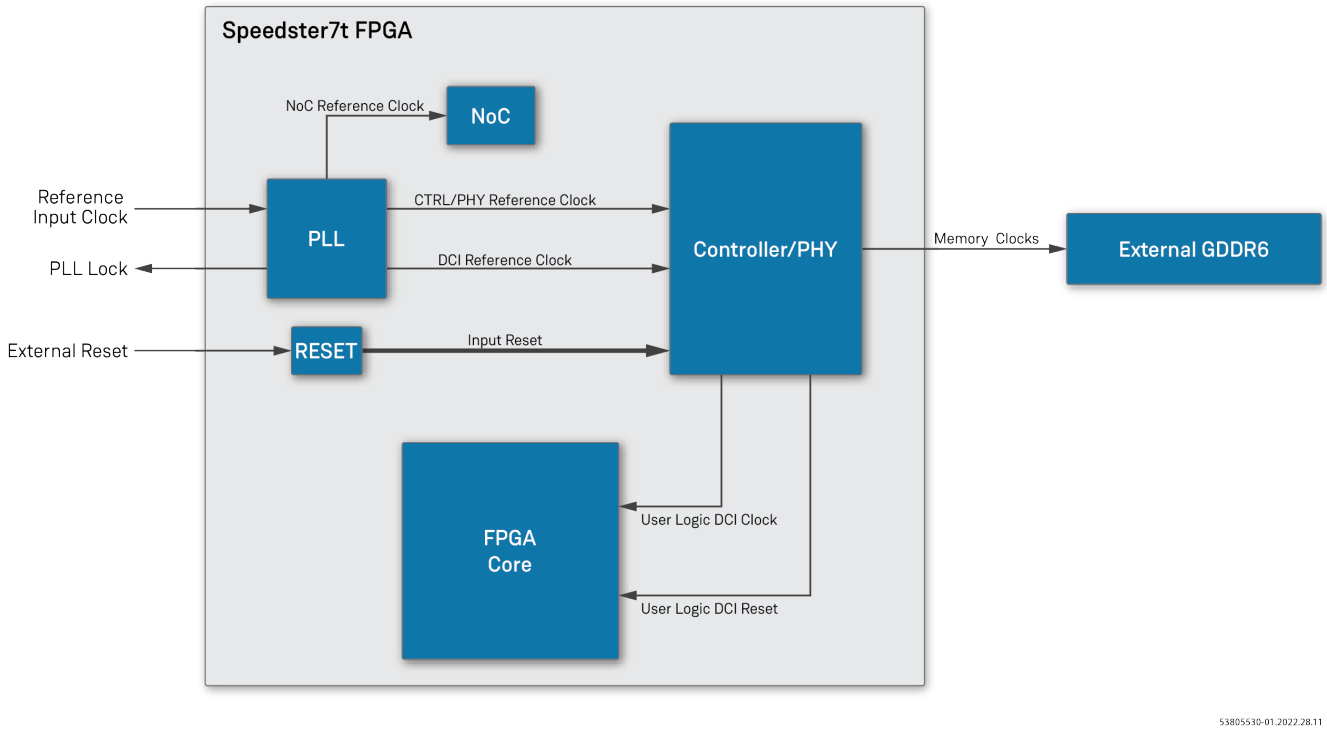


Figure 9: ***GDDR6 Subsystem Clock and Reset Architecture***

The GDDR6 memory controller clock, selected from one of the global clocks generated by the PLL, runs at a maximum of 1 GHz to support a maximum data rate of 16 Gbps (achieved when SGRAM WCK runs at 8 GHz). The PHY clock is also from the same clock source as the controller clock. The PHY internal PLL generates the CA and WCK memory clocks. The PHY maximum rate of operation is 500 MHz.

The AXI4 interface requires two asynchronous clocks selected separately in ACE I/O designer. These clocks drive the 256-bit AXI4 interface connected to the 2D NoC and the 512-bit AXI4 interface connected to the fabric. The clock driving the user logic for the GDDR6 2D NoC interface is handled internally within the 2D NoC and can operate at a maximum rate of 1Ghz. The direct connect (DC) AXI clock, also chosen from a global clock, can run at a maximum frequency of 500 MHz and drives the user logic for the GDDR6 DC interface. The 2D NoC and DC interface reference input clock rates are independent of the controller/PHY clock rate. These clocks can be scaled based on their throughput requirements.

The AXI interface clocks are routed through the GDDR subsystem and then made available to the 2D NoC or the DC interface for driving user logic to minimize clock divergence issues.

The GDDR subsystem has access to 80 available resets as stated before. All resets can also be configured through IPCNTL reset selection registers. When the 2D NoC interface is exercised, the NAPs require a reset input that can be driven from any of the available resets or generated by user logic. When the direct connect interface is utilized, the reset is supplied by the subsystem to the FPGA fabric (DCI Reset as shown in the above block diagram), making the reset synchronous to the DC interface clock.

Speedster7t AC7t1500 FPGA Clocking for GDDR6

For the Speedster7t AC7t1500 FPGA, the GDDR6 interfaces on the east side receive their clocks from the two PLLs on the east side and similarly, the two PLLs on the west side generate clocks for the west GDDR6 controllers. There is no clock domain crossing between the east and west side GDDR6 controllers, helping to achieve the maximum data rates across all eight GDDR6 interfaces. The resets for all controllers can be tied to an external reset.

The following figures detail how the reference input clocks can connect to drive all eight GDDR6 controllers. It is also recommended to use an input clock frequency of 100MHz. The preferred configuration is based on the jitter and skew assessments of the user design.

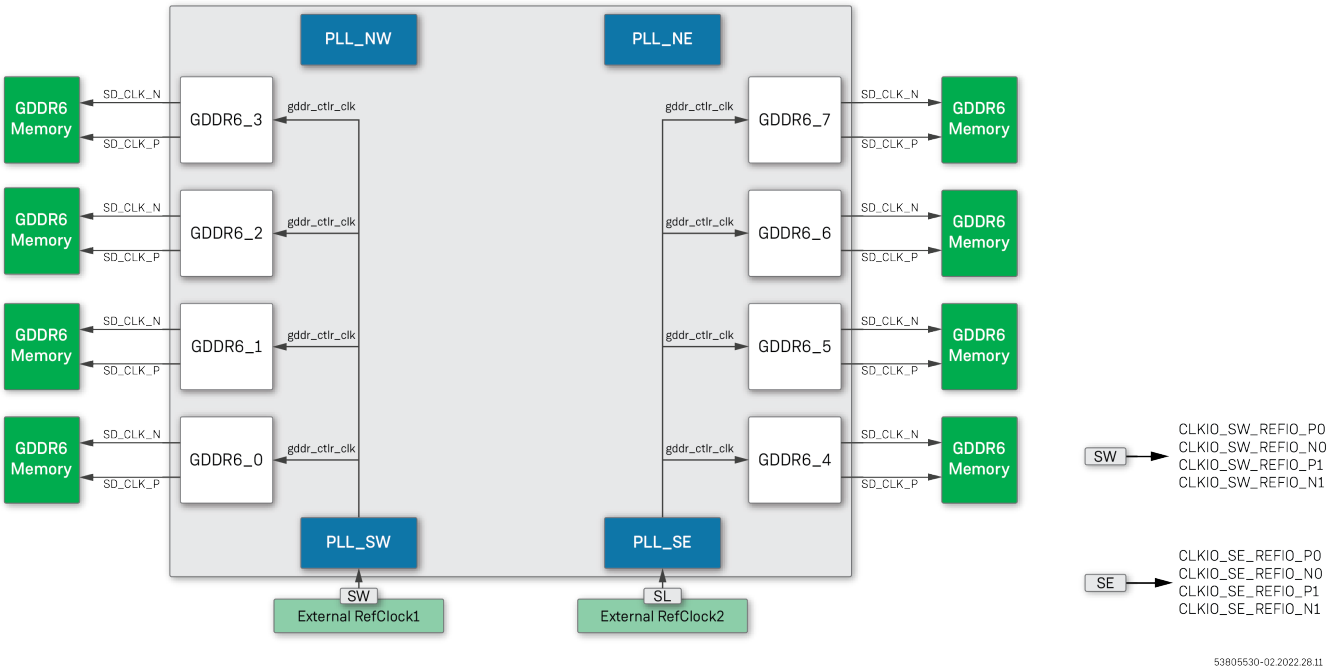


Figure 10: ***Speedster7t AC7t1500 FPGA East/West GDDR Subsystem Driven By External Clock to South PLLs***

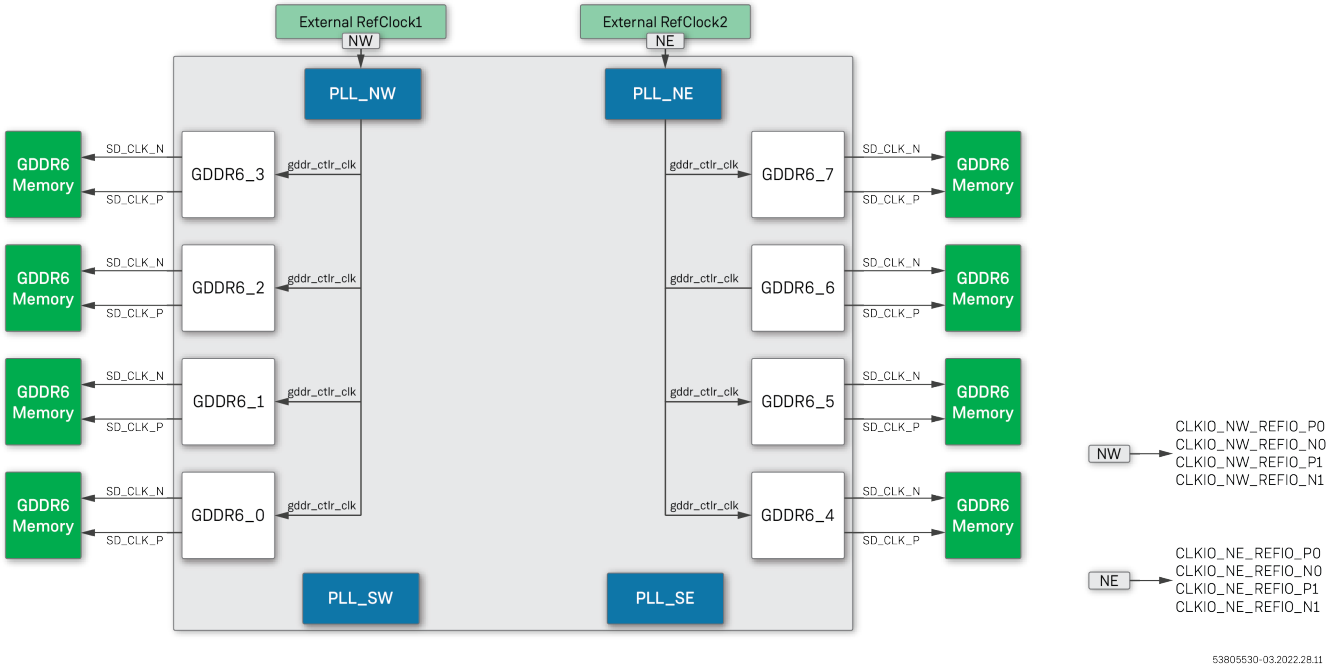


Figure 11: ***Speedster7t AC7t1500 FPGA East/West GDDR Subsystem Driven By External Clock to North PLLs***

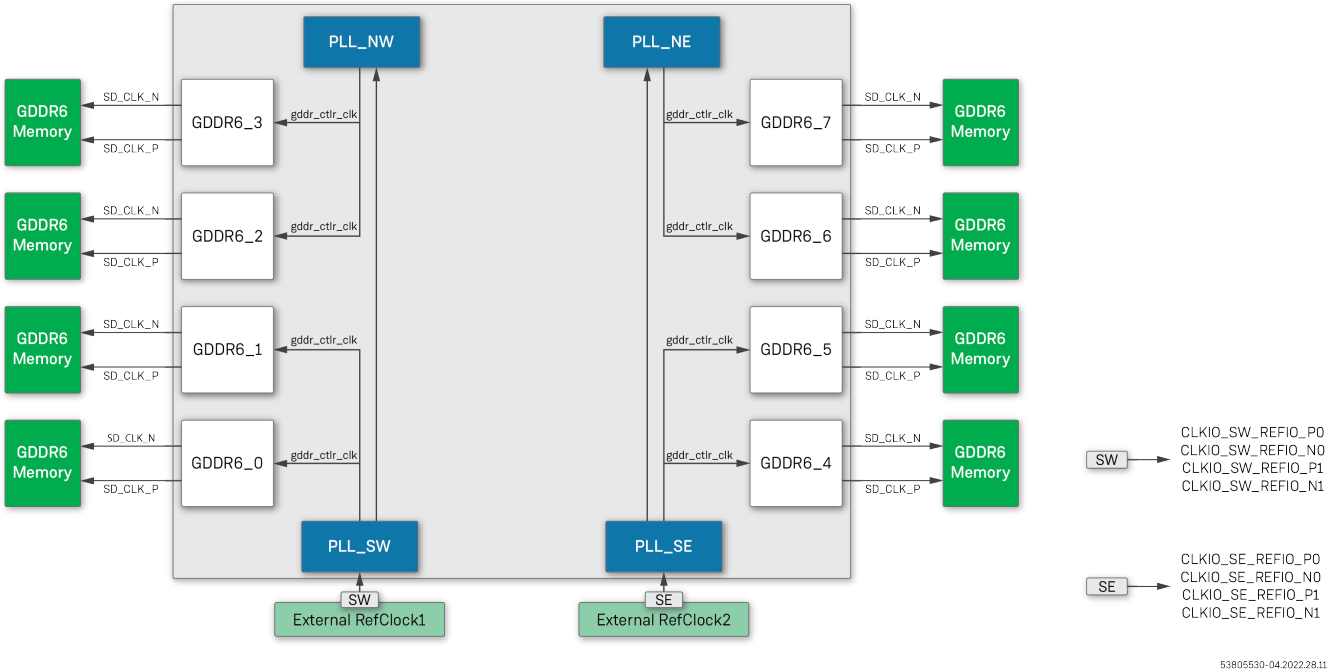


Figure 12: ***Speedster7t AC7t1500 FPGA East/West GDDR Subsystem and North Plls Driven By South PLLs***

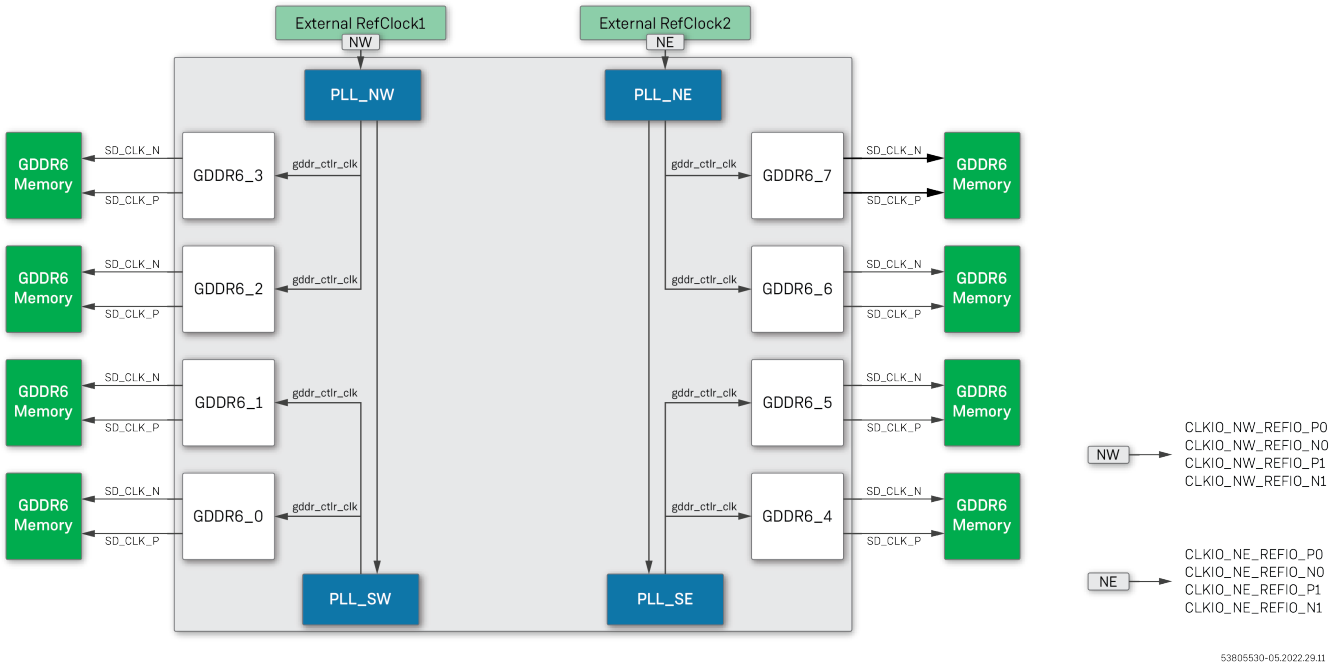


Figure 13: ***Speedster7t AC7t1500 FPGA East/West GDDR Subsystem and South Plls Driven By North PLLs***

The above scenarios all use two external clocks, one for each of the west and east side GDDR6 instances. A single input reference clock can also be used to drive all eight GDDR subsystems. This can be implemented in two ways:

1. Cascaded PLLs with generated clock.
2. Advanced PLL inner bypass path.

The following diagrams explain these configurations.

> **Note**
>
> The example frequencies below reflect a data rate of 12Gbps.

Using Cascaded PLL

The input reference clock is applied to the SW PLL at 100Mhz. A generated clock from the SW PLL is driving the SE PLL. The west GDDRs are being driven by clocks generated by the local (SW) PLL while the east GDDRs are being driven by the clocks generated from the SE PLL.

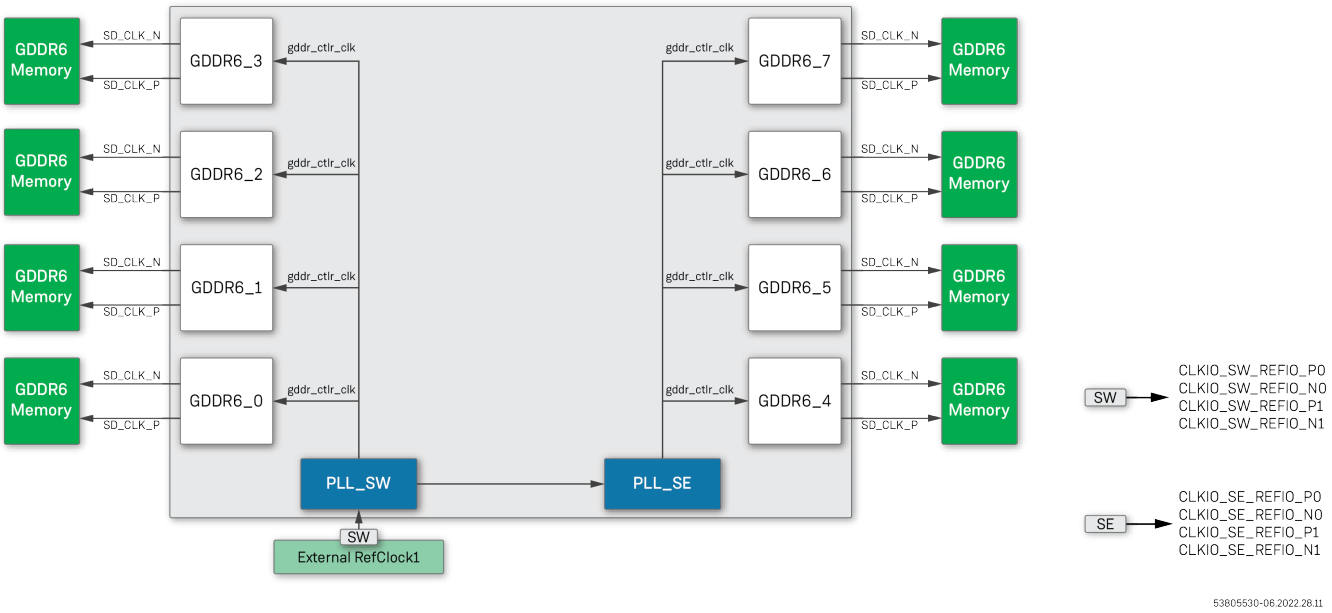


Figure 14: ***Speedster7t AC7t1500 FPGA East/West GDDR Subsystem Driven By Single CLock Via Cascaded PLL***

The following diagram details the ACE PLL configuration flow:

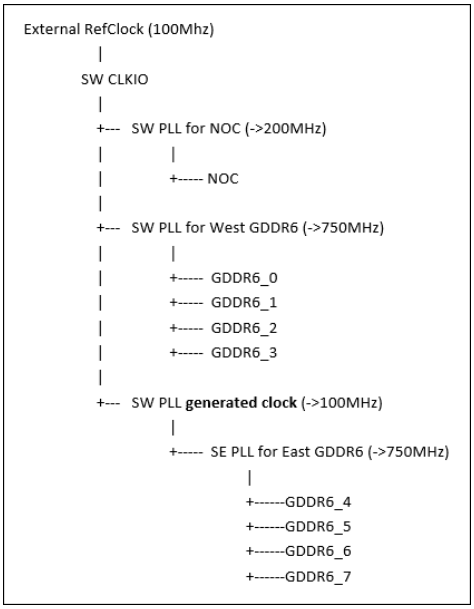


Figure 15: ***Cascased PLL Configuration Flow***

Using Advanced PLL Inner Bypass

The single reference input clock driving all GDDR subsystems on the Speedster7t AC7t1500 FPGA can also be achieved by utilizing the Advanced PLL IP inner bypass mode. In this mode, the local SW PLL can be bypassed via the inner bypass path. The following diagram details the ACE PLL configuration flow with bypass path enabled:

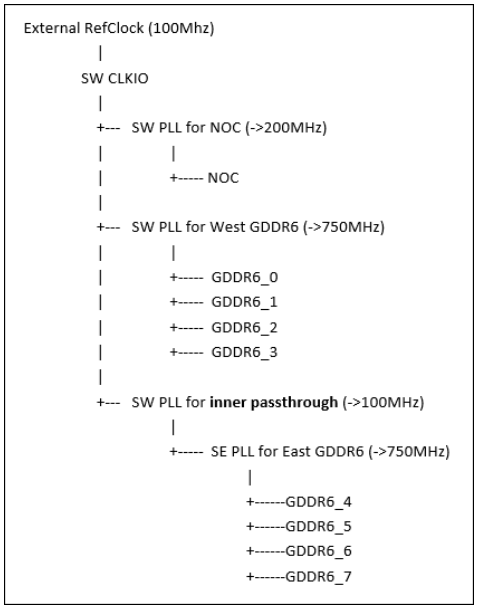


Figure 16: ***Advanced PLL Inner Bypass Configuration Flow***

Warning!

There are dedicated PLL placements allowed for specific reference clock source inputs. Ensure that your design abides by the placements. For further details please refer to the "Enabling the Inner PLL Bypass" section of the *Speedster7t Clock and Reset Architecture User Guide* (UG083).

Chapter - 5: GDDR6 Interface Connectivity

The following sections describe the 2D NoC and the DC interfaces which are supported by the GDDR6 subsystem and connect to the user logic in the FPGA fabric.

Connectivity to the 2D NoC

The Speedster7t family of FPGAs has a network hierarchy that enables extremely high-speed data flow between the FPGA core and the interfaces around the periphery as well as between logic within the FPGA itself. This on-chip network hierarchy supports a cross-sectional bidirectional bandwidth exceeding 20 Tbps. It supports a multitude of interface protocols including GDDR6, DDR4/5, 400G Ethernet, and PCI Express Gen5 data streams while greatly simplifying access to memory and high-speed protocols. The Achronix two-dimensional network on chip (2D NoC) provides for read/write transactions throughout the device as well as specialized support for 400G Ethernet streams in selected columns.

For more details, see the *Speedster7t Network on Chip User Guide* (UG089).

The GDDR6 subsystems can be connected to enable transactions with the PCIe or FPGA fabric through the 2D NoC interface. PCIe can initiate transactions to any GDDR6 channel using the 2D NoC, in which case, the PCIe endpoint is the initiator with the GDDR6 acting as the responder. Similarly, the FPGA initiator logic can issue a transaction to its local network access point (NAP), which carries the transaction to the east or west side of the FPGA core, where it is presented to the 2D NoC. From there, the 2D NoC carries data to the appropriate GDDR6 interface. The responses follow the same path in reverse.

In addition, the 2D NoC provides a connection from the FPGA fabric and IP interfaces to the FPGA configuration unit (FCU). The FCU receives bitstreams and is used to configure the FPGA fabric as well as the various IP interfaces on the device. The 2D NoC also provides read and write access to the control and status register (CSR) space. The CSR space includes control registers and status registers for the IP interfaces.

2D NoC connectivity is the default path in ACE I/O Designer and is the primary interface expected to be used. The input reference clock for the 2D NoC is selected from the global clock outputs and always operates at 200 MHz. The 2D NoC connection is a 256-bit AXI4 interface with one interface available per controller per subsystem, running up to 1 GHz and generating data rates of 16 Gbps.

- AXI4 256b is a 256-bit responder interface connected to the 2D NoC initiator AXI.
- AXI4 256b operates in single-clock mode. A synchronous clock must be provided to both the read and write ports. Write and read data width is 256 bits.
- AXI4 converts AXI transactions to the local bus transactions which connect directly to the MPFE and has independent command, write and read data FIFOs.
- AXI4 supports all burst sizes, types, and lengths including incremental and wrapping bursts.

The following figure is the I/O diagram of an AXI4-256b Interface. For more details on AXI transactions, refer to the AMBA AXI Protocol Specification.

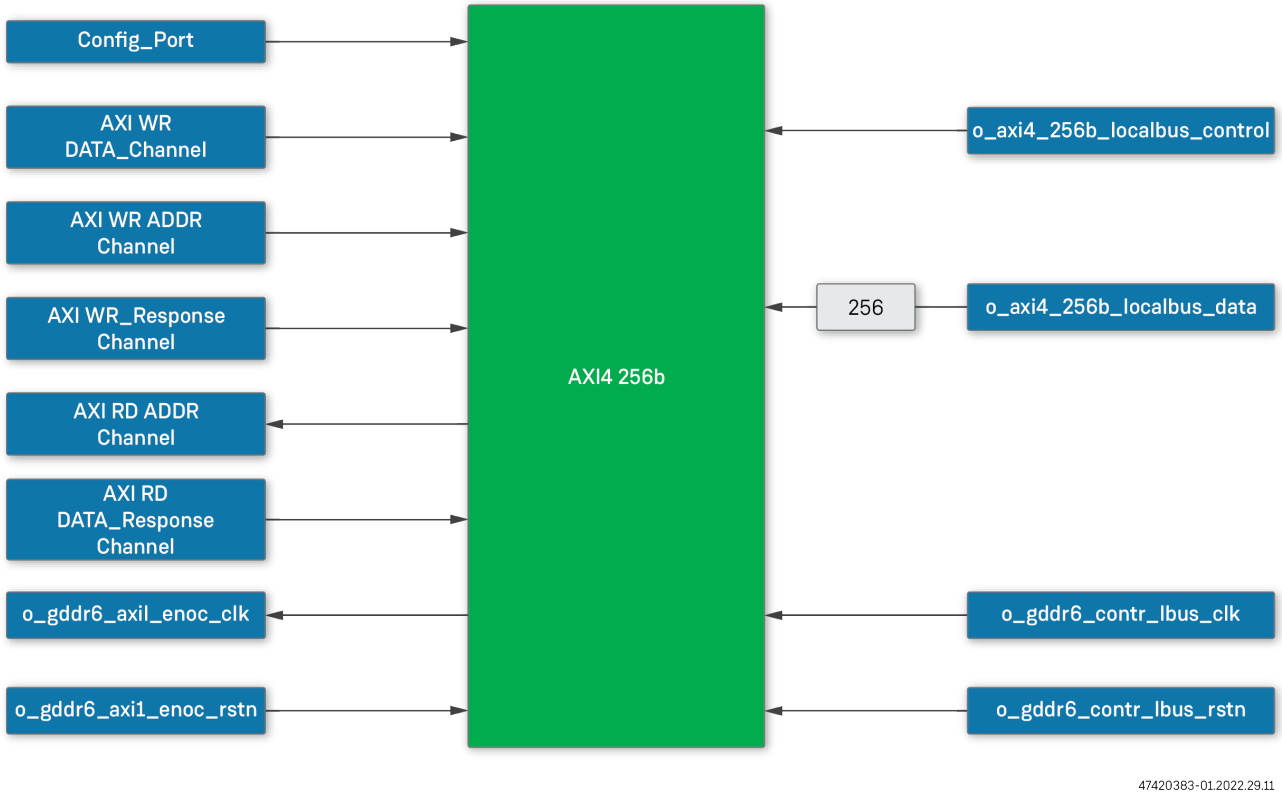


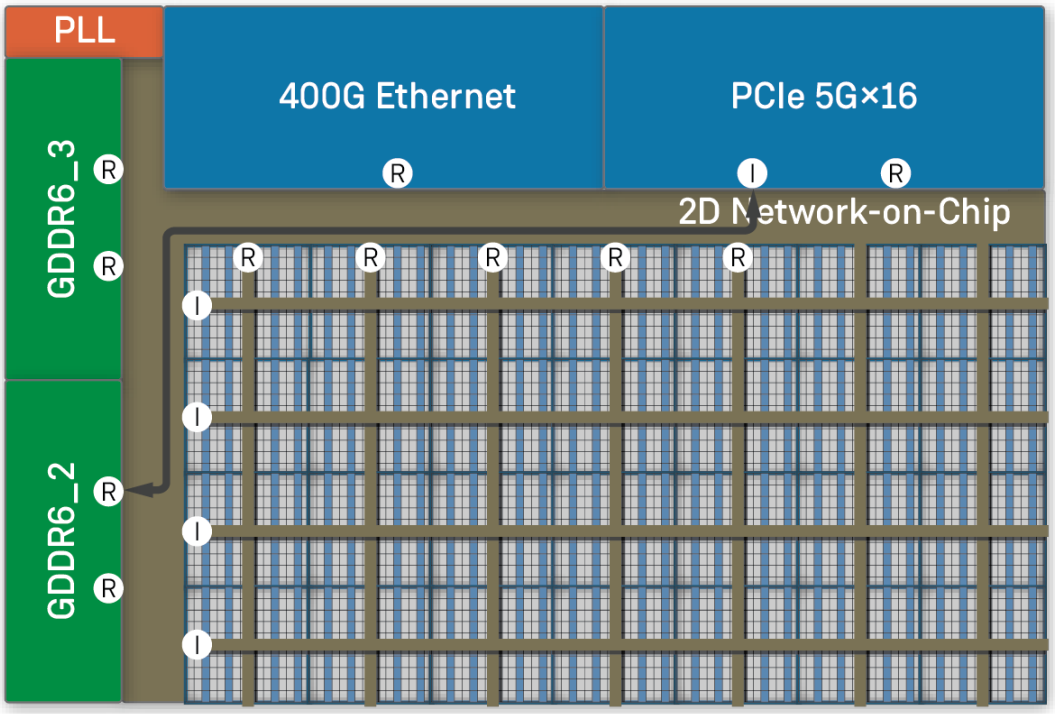
Figure 17: ***AXI4-256b I/O Diagram***

The following table provides the parameters given to the AXI4 Interface:

Table 5: ***AXI4-256b Interface Parameters***

Parameter	Value	Description
AXI_DATA_WIDTH	256	Data width of the AXI4 interface.
AXI_SLAVE_ID_WIDTH	7	Width of awid, wid, bid, arid and rid ports.
AXI_CMD_FIFO_AWIDTH	3	Sets the depth of the read and write command FIFOs. The depth is equal to 2AXI_CMD_FIFO_AWIDTH .
AXI_ADDR_WIDTH	33	Width of the awaddr and araddr ports.
AXI_LEN_WIDTH	8	Width of the length port.
SDRAM_DSIZE	256	Size of the SDRAM.
MAX_SN_WIDTH	8	Bit width of the number of slots to be allocated in the read and write data buffers in the AXI4 Interface.
ENABLE_INTRAPORT_REORDER	1	If enabled, requests within a port as well as requests across different ports may be reordered. When intra-port reordering is enabled, data may be written to or read from the memory devices in an order different from that in which they are issued by the AXI4 interface. In this case, the AXI core must reorder the read data coming from the memory so that it is presented to the AXI4 interface in the proper order. The AXI core must present write data to the memory controller in the order requested by the controller.
ADDR_MAP_SIG_BITS	10	Sets the number of significant bits used in the address value comparison (typically 10).
BURST_SIZE_WIDTH	13	Width of burst size.
AXI_READ_ONLY	0	When set to "1", disables write accesses through the port and removes the write logic.
AXI_WRITE_ONLY	0	When set to "1", disables read accesses through the port and removes the read logic.
REQ_PRIORITY_WIDTH	3	Width of the request priority attribute.
SINGLE_CLOCK_MODE	1	Set when aclk and l_clk are in the same clock domain, to optimize latency.

The following figure details how the PCI Express initiator issues a transaction to the 2D NoC, which transmits it directly to the GDDR6 interface without involving any resources in the FPGA fabric.



47420383-03.2023.17.02

Figure 18: ***Data Flow From PCIe Interface to GDDR6 Subsystem Through 2D NoC***

The following figure shows how the initiator logic in the FPGA fabric interacts with the GDDR6 interface utilizing the 2D NoC.

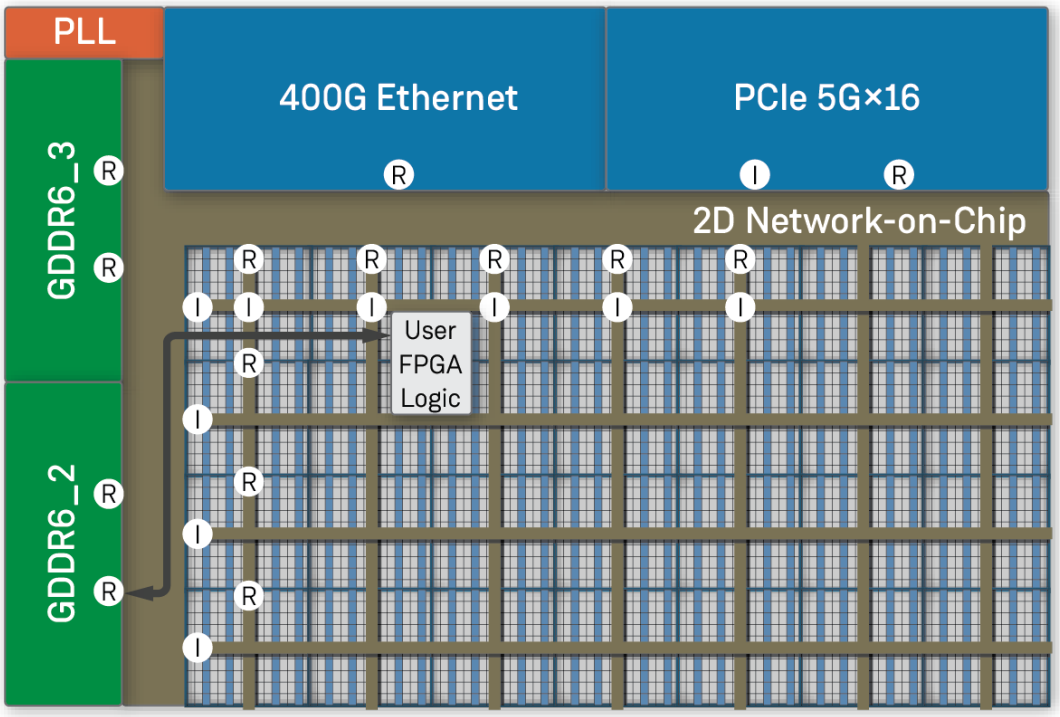


Figure 19: ***Data Flow From FPGA Fabric to GDDR6 Subsystem Through 2D NoC***

2D NoC Addressing for GDDR6

To access a GDDR6 subsystem, the address must be set in the top-level RTL to access a GDDR6 subsystem. The address for a GDDR6 subsystem is represented by the target ID and the memory address. The target ID comprises the nine most significant bits (`Addr[41:33]`). The remaining bits (`Addr[32:0]`) represent the external memory address.

The following table details the GDDR6 2D NoC address.

Table 6: *GDDR6 2D NoC Addressing Scheme*

Address Bit	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	…	0
GDDR6	0	0	0	0	0	Ctrl ID				Memory Address												

The 2D NoC GDDR6 address components are detailed in the following table:

Table 7: *GDDR6 2D NoC Address Components*

Bits	Field	Description
`Addr[41:37]`	Constant	`5'b00000`
`Addr[36:33]`	Ctrl ID	Selects one of the eight destination GDDR6 controllers and its channel. `Addr[36:34]` selects the controller. `Addr[33]` selects one of the two channels on each controller.
`Addr[32:0]`	Memory Address	The memory address of the controller and channel specified by Ctrl ID.

GDDR6 Controller ID Mappings

The following table details the addressing scheme for transactions targeting the different GDDR6 subsystems, their respective control IDs, and their Speedster7t AC7t1500 FPGA interface names to be specified in the user RTL:

Table 8: *GDDR6 Subsystem Control ID and ACE Interface Map for 2D NoC*

ACE Instance Name	Control ID	GDDR6 Channel No.	Speedster7t AC7t1500 Interface Name
GDDR6_0	4'b1100	Channel 0	`gddr6_w0_noc0`
	4'b1101	Channel 1	`gddr6_w0_noc1`
GDDR6_1	4'b0100	Channel 0	`gddr6_w1_noc0`
	4'b0101	Channel 1	`gddr6_w1_noc1`
GDDR6_2	4'b0000	Channel 0	`gddr6_w2_noc0`
	4'b0001	Channel 1	`gddr6_w2_noc1`
GDDR6_3	4'b1000	Channel 0	`gddr6_w3_noc0`
	4'b1001	Channel 1	`gddr6_w3_noc1`
GDDR6_4	4'b1111	Channel 0	`gddr6_e0_noc1`
	4'b1110	Channel 1	`gddr6_e0_noc0`

ACE Instance Name	Control ID	GDDR6 Channel No.	Speedster7t AC7t1500 Interface Name
GDDR6_5	4'b0111	Channel 0	gddr6_e1_noc1
	4'b0110	Channel 1	gddr6_e1_noc0
GDDR6_6	4'b0011	Channel 0	gddr6_e2_noc1
	4'b0010	Channel 1	gddr6_e2_noc0
GDDR6_7	4'b1011	Channel 0	gddr6_e3_noc1
	4'b1010	Channel 1	gddr6_e3_noc0

Warning!

The address mappings apply only to the GDDR6 memory address space. For configuration space (CSR) address mappings, please see the appropriate CSR mapping table. Also, GDDR6 subsystems on the west side use odd addresses for channel 1, whereas the east side uses even addresses.

The following is an Example target ID setting to access GDDR6_5 channel 1 that must be set in the top-level RTL:

```
parameter GDDR6_ADDR_ID = 9'b000000110;
```

The mapping of the addresses is explained in the section GDDR6 Memory Address Mapping to AXI Addresses (see page 37).

Connectivity Through the DC Interface

The middle two GDDR6 subsystems on the east and west sides of the chip are the only four that enable the DC interface connection directly to the fabric. This connection is a 512-bit AXI4 interface (one per controller per subsystem), capable of running up to 500 MHz and supporting data rates of up to 16 Gbps. The interface offers:

- An AXI4 512b, 512-bit responder interface, connected to a fabric initiator AXI interface.
- Asynchronous read and write clock. The write clock is half of the read clock. An asynchronous FIFO handles clock domain crossing and 512-bit AXI to 256-bit local data conversion.
- Write clock provided by one of the global clocks. The read clock is the controller clock which is synchronous to controller, AXI1 and PHY interfaces. The selection of clocks are controlled via the IPCNTRL register.
- Conversion of AXI transactions to the local bus transactions.

The following is an I/O diagram of an AXI4-512b Interface:

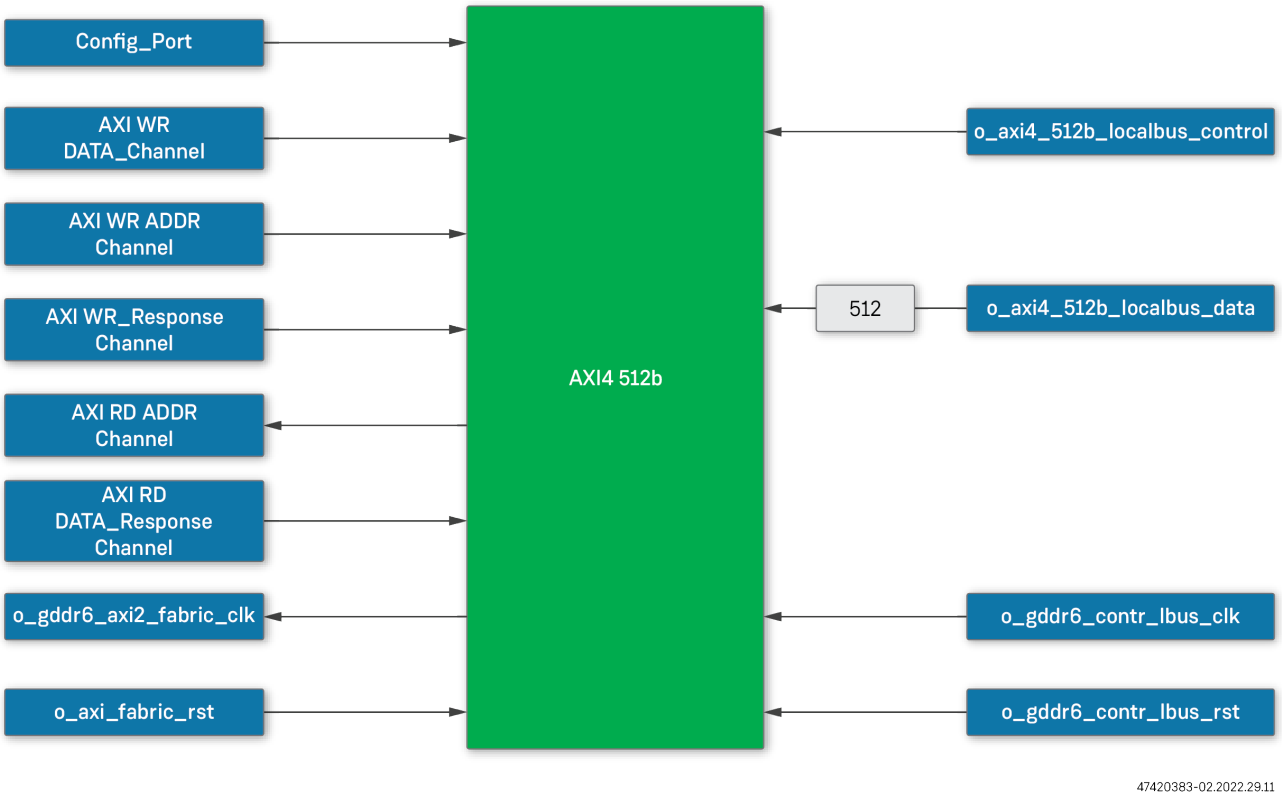


Figure 20: ***AXI4-512b I/O Diagram***

The following table details the parameters given to the AXI4 Interface.

Table 9: ***AXI4-512b Interface Parameters***

Parameter	Value	Description
AXI_DATA_WIDTH	512	Data width of the AXI4 interface.
AXI_SLAVE_ID_WIDTH	7	Width of awid, wid, bid, arid and rid ports.
AXI_CMD_FIFO_AWIDTH	3	Sets the depth of the read and write command FIFOs. The depth is equal to 2AXI_CMD_FIFO_AWIDTH .
AXI_ADDR_WIDTH	33	Width of the awaddr and araddr ports.
AXI_LEN_WIDTH	8	Width of the length port.
SDRAM_DSIZE	256	Size of SDRAM.
MAX_SN_WIDTH	8	Bit width of the number of slots to be allocated in the read and write data buffers of the AXI4 interface.
ENABLE_INTRAPORT_REORDER	1	If specified, requests within a port as well as requests across different ports may be reordered. When intra-port reordering is enabled, data may be written to or read from the memory devices in an order different from that in which they are issued by the AXI4 interface. In this case, the AXI core must reorder the read data coming from the memory so that it is presented to the AXI4 interface in the proper order. The AXI core must present write data to the memory controller in the order requested by the controller.
ADDR_MAP_SIG_BITS	10	Sets the number of significant bits used in the address value comparison (typically 10).
BURST_SIZE_WIDTH	13	Width of burst size.
AXI_READ_ONLY	0	When set to "1", disables write accesses through the port and removes the write logic.
AXI_WRITE_ONLY	0	When set to "1", disables read accesses through the port and removes the read logic.
REQ_PRIORITY_WIDTH	3	Width of request priority attribute.
SINGLE_CLOCK_MODE	0	Set when aclk and l_clk are in the same clock domain.

DC Addressing for GDDR6

Similar to the 2D NoC interface, for GDDR6 subsystems accessing the FPGA fabric via the DC interface, the AXI transactions use the address mapping shown in the following table along with the DC-specific GDDR6 subsystems and their corresponding channel mappings:

Table 10: *GDDR6 Subsystem ACE Interface Map for DC Interface*

ACE Instance Name	GDDR6 Channel No.	Speedster7t AC7t1500 Interface Name
GDDR6_1	Channel 0	`gddr6_w1_dc0`
	Channel 1	`gddr6_w1_dc1`
GDDR6_2	Channel 0	`gddr6_w2_dc0`
	Channel 1	`gddr6_w2_dc1`
GDDR6_5	Channel 0	`gddr6_e1_dc0`
	Channel 1	`gddr6_e1_dc1`
GDDR6_6	Channel 0	`gddr6_e2_dc0`
	Channel 1	`gddr6_e2_dc1`

GDDR6 Memory Address Mapping to AXI Addresses

The Speedster7t FPGA supports a maximum memory density of 16 Gb, which is 8 Gb per channel. The AXI memory address, `Addr[32:0]`, is 33 bits wide. The following is an example of the address mapping for a GDDR6 device density of 16 Gb in ×16 mode:

- `Addr[29:16]` – Row address
- `Addr[15:12]` – Bank address
- `Addr[11:5]` – Column address
- `Addr[4:0]` – AXI Byte address

The bits `Addr[32:30]` for the above example are set to zero.

For the same 8 gb per channel configuration in ×8 mode, the address mapping is:

- `Addr[30:16]` – Row address
- `Addr[15:12]` – Bank address
- `Addr[11:5]` – Column address
- `Addr[4:0]` – AXI Byte address

The bits `Addr[32:31]` for the above example are set to zero. The GDDR6 SGRAM addressing scheme for other supported device densities can be found in the JEDEC specification standard JESD250.

Chapter - 6: GDDR6 Core and Interface Signals

This section provides a detailed list of all signals that interface with each GDDR6 subsystem.

Clock and Reset

The following table summarizes the different direct-connect interface clock and reset signals that originate as outputs from the GDDR6 subsystems and are driven into the fabric core so that the associated logic can use these signals for clock and reset purposes. There is one clock and reset signal per channel for each GDDR6 subsystem.

Table 11: *Clock and Reset Signals*

Pin Name	Direction	Width	Description
`<prefix>_chan[0/1]_clk`	Output	1	Output clock from the GDDR6 subsystem to the fabric. Generated from within the subsystem using the GDDR6 controller global clock provided by the user.
`<prefix>_chan[0/1]_rstn`	Output	1	Output reset from the GDDR6 subsystem to the fabric. Generated from within the subsystem using the GDDR6 controller global reset provided by the user.

Errors and Interrupts

The following table summarizes the error and interrupt output signals from the GDDR6 subsystem.

Table 12: *Error and Interrupt Signals*

Pin Name	Direction	Width	Description
`<prefix>_chan[0/1]_crc_error`	Output	1	Read or write CRC error signal for corresponding channels indicating error on CRC mismatch.
`<prefix>_chan[0/1]_interrupt`	Output	3	Controller interrupt output signal indicating further action needed based on error condition.
`<prefix>_chan[0/1]_interrupt_or`	Output	1	ORed output of all interrupt signals connected to fabric.

AXI Interface Signals

The table below shows the Controller to AXI4 Interface connects to the Fabric in the GDDR6 subsystem.

Table 13: *Controller to AXI Interface Signals*

Pin Name	Direction	Width	Description
<prefix>_chan0/1_awid	Input	7	Sets write address channel ID (AWID, identification tag for write address group of signals).
<prefix>_chan0/1_awaddr	Input	33	Sets write address (address of first transfer in write burst transaction).
<prefix>_chan0/1_awlen	Input	4	Sets burst length (exact number of transfers in burst). Determines number of data transfers associated with address.
<prefix>_chan0/1_awsize	Input	3	Sets burst size (size of each transfer in burst).
<prefix>_chan0/1_awburst	Input	2	Sets burst type. Burst type and size information determines how address for each transfer within burst is calculated.
<prefix>_chan0/1_awlock	Input	2	Sets lock type (provides additional information about atomic characteristics of transfer).
<prefix>_chan0/1_awcache	Input	4	Sets memory type (indicates how transactions must progress through system).
<prefix>_chan0/1_awprot	Input	3	Sets protection type (indicates transaction privilege, security level, and whether transaction is data or instruction access.
<prefix>_chan0/1_awvalid	Input	1	Sets write address valid (indicates channel is signaling valid write address and control information).
<prefix>_chan0/1_awready	Output	1	Indicates write address ready (responder is ready to accept address and associated control signals).
<prefix>_chan0/1_awqos	Input	3	Sets QoS identifier (sent on write address channel for each write transaction).
<prefix>_chan0/1_wid	Input	7	Sets write ID tag (ID tag of write data transfer).
<prefix>_chan0/1_wdata	Input	512	256-bit wide write data input signal.
<prefix>_chan0/1_wstrb	Input	32	Sets write strobes (indicates which byte lanes hold valid data).
<prefix>_chan0/1_wlast	Input	1	Sets write last (indicates the last transfer in write burst).

Pin Name	Direction	Width	Description
<prefix>_chan0/1_wvalid	Input	1	Indicates write data channel signals are valid.
<prefix>_chan0/1_wready	Output	1	Indicates transfer on write data channel can be accepted.
<prefix>_chan0/1_bid	Output	7	Identification tag for write response.
<prefix>_chan0/1_bresp	Output	2	Write response signal indicating status of write transaction.
<prefix>_chan0/1_bvalid	Output	1	Indicates write response channel signals are valid.
<prefix>_chan0/1_bready	Input	1	Indicates transfer on write response channel can be accepted.
<prefix>_chan0/1_arid	Input	7	Sets identification tag for read transaction.
<prefix>_chan0/1_araddr	Input	33	Address of first transfer in read transaction.
<prefix>_chan0/1_arlen	Input	4	Sets exact number of data transfers in read transaction.
<prefix>_chan0/1_arsize	Input	3	Sets number of bytes in each data transfer in read transaction.
<prefix>_chan0/1_arburst	Input	2	Sets burst type (indicates type of address change between each transfer in read transaction).
<prefix>_chan0/1_arlock	Input	2	Provides information about atomic characteristics of read transaction.
<prefix>_chan0/2_arcache	Input	4	Indicates how read transaction is required to progress through system.
<prefix>_chan0/1_arprot	Input	3	Sets protection attributes of read transaction (privilege, security level, access type).
<prefix>_chan0/1_arvalid	Input	1	Indicates read address channel signals are valid.
<prefix>_chan0/1_arready	Output	1	Indicates transfer on read address channel can be accepted.
<prefix>_chan0/1_arqos	Input	3	QoS identifier for read transaction.
<prefix>_chan0/1_rid	Output	7	Identification tag for read data and response (responder must ensure that RID value of any returned data matches ARID value of corresponding address).
<prefix>_chan0/1_rdata	Output	512	256-bit wide read data from memory.
<prefix>_chan0_rresp	Output	2	Read response signal indicating status of read transfer.

Pin Name	Direction	Width	Description
<prefix>_chan0_rlast	Output	1	Indicates last data transfer in read transaction.
<prefix>_chan0/1_rvalid	Output	1	Indicates read data channel signals are valid.
<prefix>_chan0/1_rready	Input	1	Indicates transfer on read data channel can be accepted.

PHY Memory Signals

The table below summarizes the external memory to PHY interface signals.

Table 14: *Memory-to-PHY Interface Signals*

Pin Name	Direction	Width	Description
<prefix>_sd_clk_[n/p]	Output	1	Differential clock inputs for command address bus.
<prefix>_sd_reset_n	Output	1	Active-low reset to GDDR6 memory.
<prefix>_[c0/c1]_sd_cke_n	Output	1	Active-low clock enable input to GDDR6 memory.
<prefix>_[c0/c1]_sd_ca	Output	10	Command address inputs to GDDR6 memory.
<prefix>_[c0/c1]_sd_cabi_n	Output	1	Active-low command address bus inversion input to GDDR6 memory.
<prefix>_[c0/c1]_sd_wck_[n/p]	Output	2	Differential clock inputs for data bus.
<prefix>_[c0/c1]_sd_dq	inout	16	Bidirectional data input/output to and from memory.
<prefix>_[c0/c1]_sd_dbi_n	inout	2	Active-low data bus inversion inputs to GDDR6 memory.
<prefix>_[c0/c1]_sd_edc	Input	2	Error detection code from GDDR6 memory.

More information on the GDDR6 device-level pins can be found in the *Speedster7t Pin Connectivity User Guide* (UG084), and the power-level requirements for the GDDR6 signals can be found in the *Speedster7t Power User Guide"* (UG087).

Chapter - 7: GDDR6 IP Software Support in ACE

Overview

The GDDR6 IP generation in ACE provides a GUI-based interface to generate and integrate the GDDR6 subsystem instances based on the user-specified inputs. The I/O designer toolkit in ACE supports the configuration and integration of all of the chosen IP for the user design. The toolkit also allows selection of the placement for each individual IP and to visualize the location of the configured subsystem. When the desired IP is configured via the I/O designer GUI interface, ACE generates a bitstream for the entire IP interface which is independent of the bitstream generated for the core fabric. The tool then integrates both of these bitstreams into a single configurable bitstream targeting a Speedster7t FPGA.

The following steps briefly describe creating a GDDR6 IP interface design for the Speedster7t AC7t1500 FPGA:

1. Create a Project (see page 43)
2. IP Configuration and Placement (see page 44)
3. Configure the PLL (see page 45)
4. Configure the 2D NoC (see page 46)
5. Configure the GDDR6 Subsystem (see page 47)
6. Clone a GDDR6 Instance (Optional) (see page 49)
7. Configure the Achronix Device Manager (see page 50)
8. Check for Errors and Generate Files (see page 51)

Create a Project

1. Create a new project in ACE.
2. In the Options tab of the Project perspective, set **Target Device** to **AC7t1500ES0** to ensure that the appropriate IP options are available in the IP perspective window:

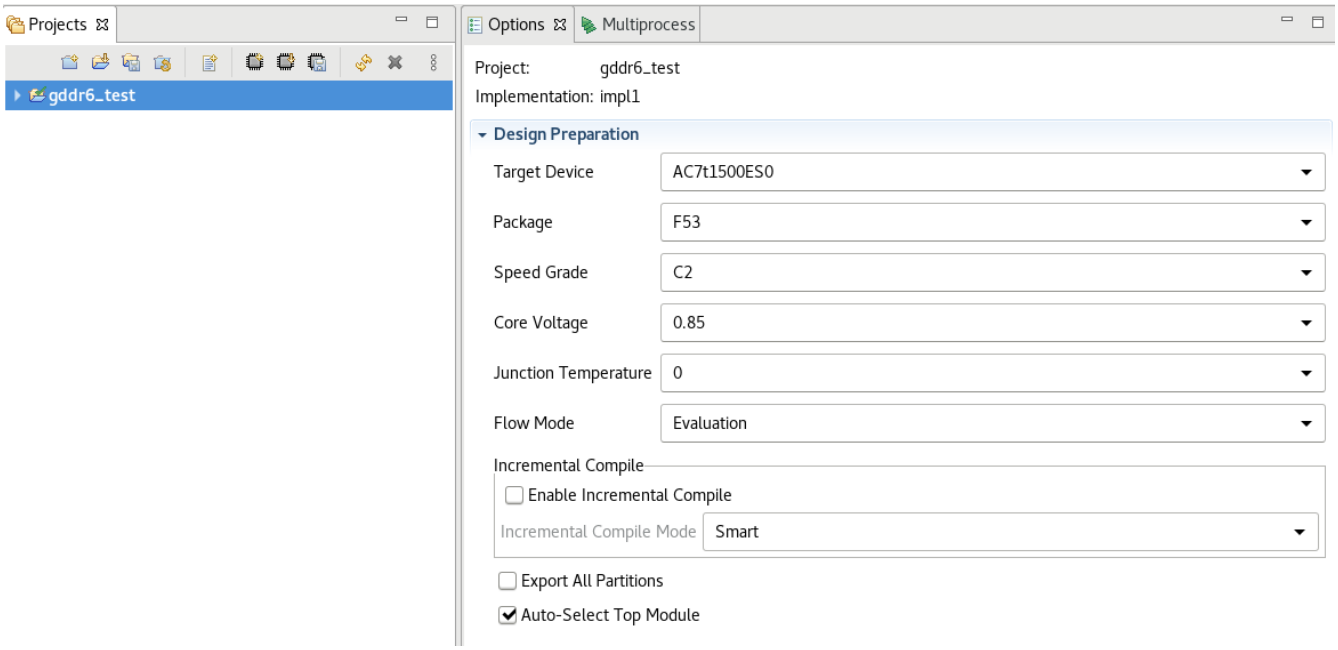


Figure 21: ***Design Preparation Options in the ACE Project***

IP Configuration and Placement

1. Switch to the IP Configuration perspective.
2. In the IP Libraries window, select **Speedster7t → IO Ring → Clock I/O Bank**. This selection creates an `.acxip` file that can be used to create the external input clock source to the hard IPs.
3. Select the ball placement for each of these pins and the desired frequency for the clock input.
   When the selection is made, the layout diagram highlights the chosen clock input and the top-level pins.
   Any errors or warnings that occur while configuring the clock I/O are highlighted in the IP Problems window.

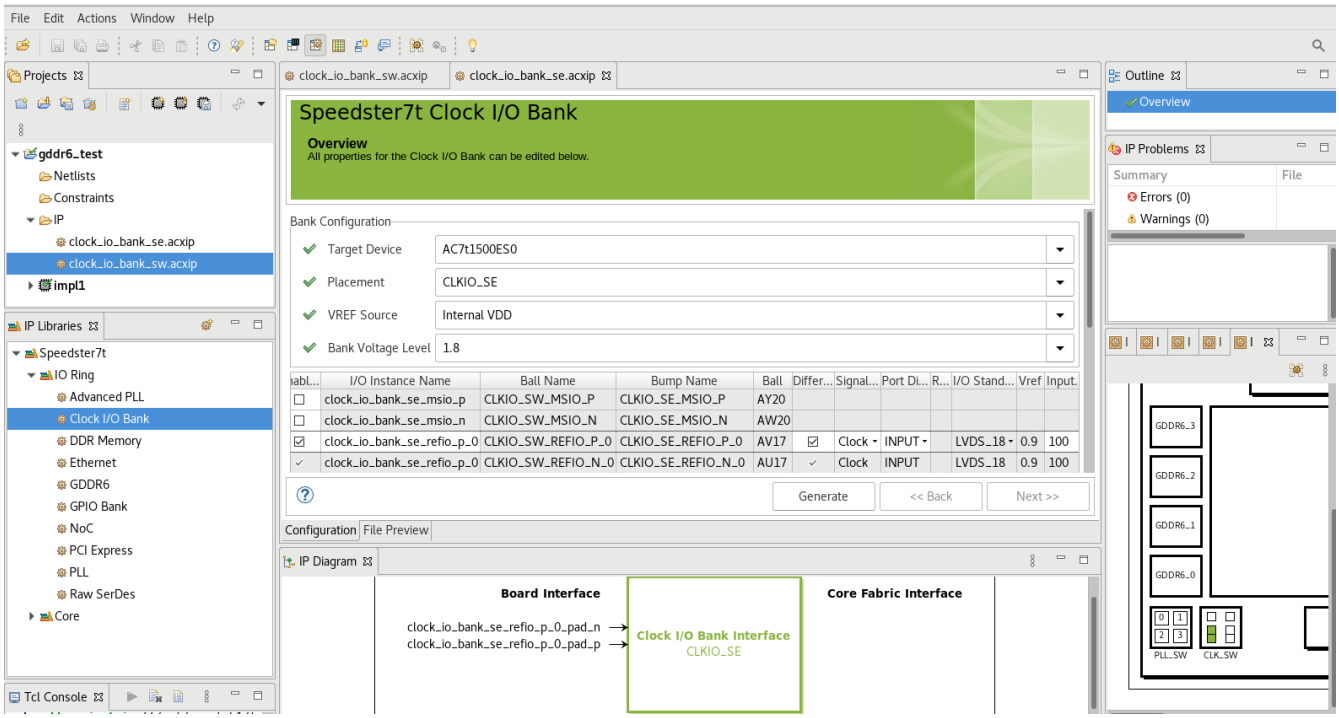


Figure 22: *Clock I/O Bank IP Configuration in ACE I/O Designer*

Configure the PLL

1. In the IP Libraries window, select **Speedster7t → IO Ring → PLL**.
2. Configure the PLL IP with the desired placement and appropriate clock output frequencies based on the data rate and the interfaces required for the GDDR6 subsystem.
   The GDDR6 IP requires a GDDR6 reference clock for the controller and PHY operations, a 200 Mhz reference input clock for the 2D NoC interface and, if the GDDR6 subsystem uses the direct-connect (DC) interface, the PLL also needs to supply a DC interface clock. As result, the number of PLL clock outputs must match the number of required clock inputs for the GDDR6 subsystem with the appropriate clock frequencies.
   The GDDR6 controller clock, 2D NoC clock or DC interface clock need not be exposed to the fabric core unless these clocks are used by other blocks in the core logic. However, if these clocks are required, the **Expose Clock Output to Core Fabric** option must be enabled to allow ACE to connect this clock input to the fabric core.

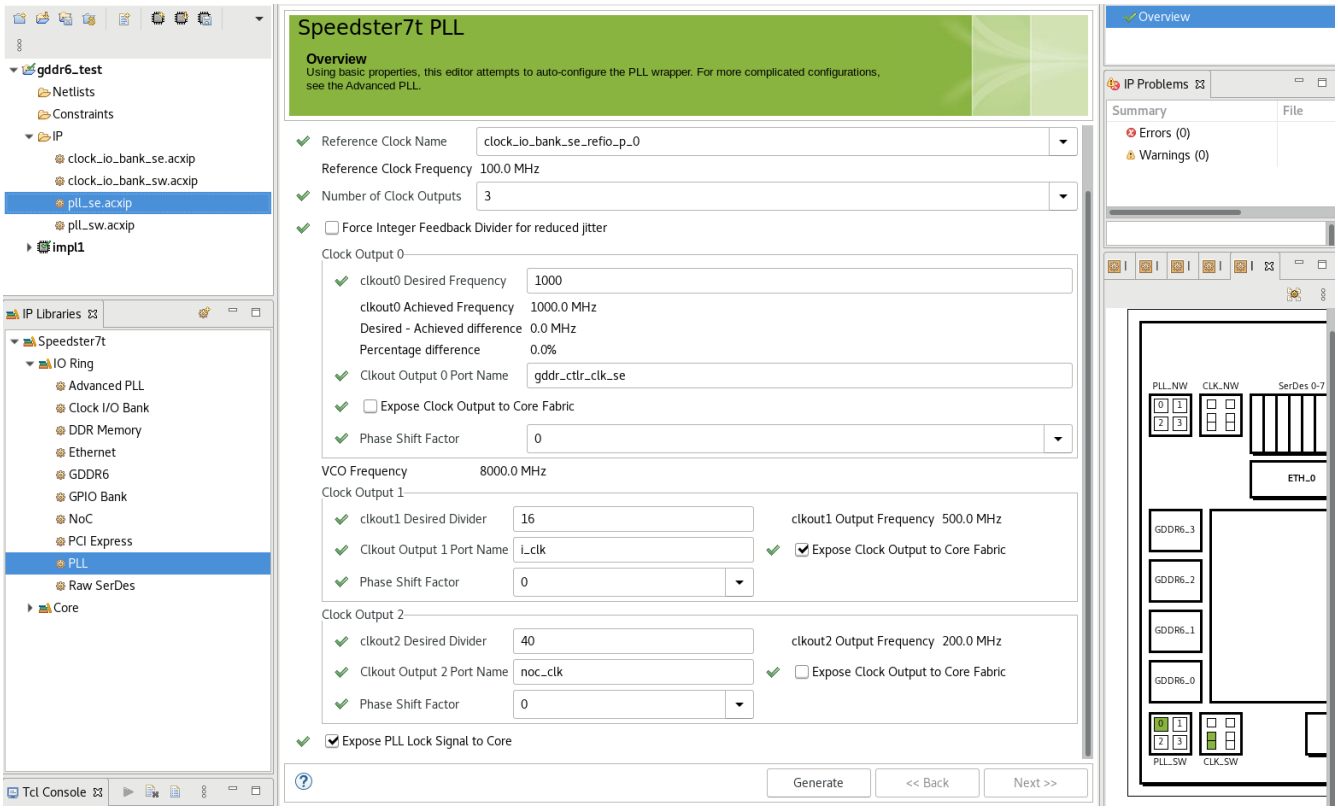


Figure 23: ***PLL IP Configuration in ACE I/O Designer***

Configure the 2D NoC

1. In the IP Libraries window, select **Speedster7t → IO Ring → NoC**.
2. Configure the reference clock name and frequency.

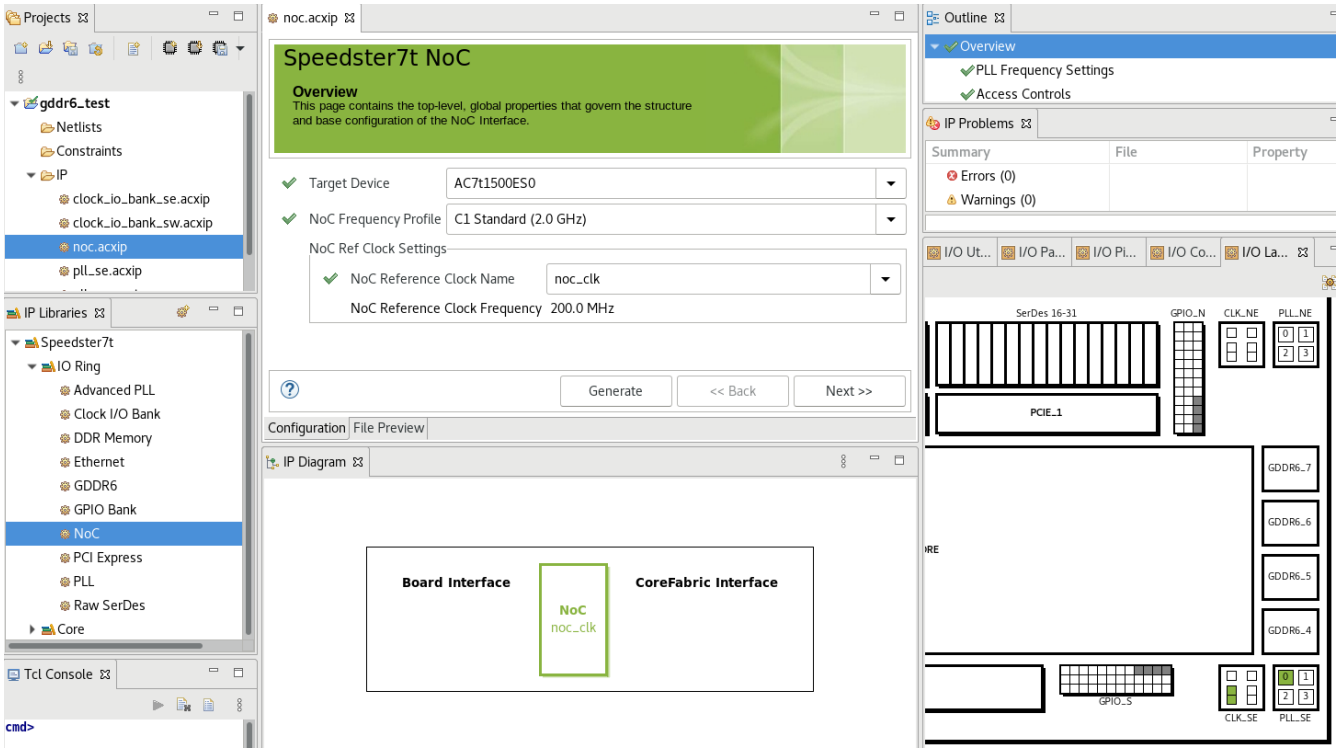


Figure 24: ***NoC IP Configuration in ACE I/O Designer***

Every GDDR user design must interface with the 2D NoC for two reasons:

1. The Achronix device manager responsible for training GDDR subsystems utilizes the 2D NoC.
2. The 2D NoC is needed if the GDDR subsystems use the 2D NoC interface for data transactions.

Thus, the 2D NoC IP must be instantiated and the appropriate 2D NoC reference clock input needed by the PLL must be selected as shown.

Configure the GDDR6 Subsystem

Each GDDR6 subsystem to be used in the design must be configured.

1. In the IP Libraries window, select **Speedster7t → IO Ring → GDDR6**.
2. Select the following items:
   - The desired placement for the particular GDDR6 interface
   - The memory part number
   - The data rate
   - The mode of operation

   The GDDR6 clock settings show the available valid clock input selections for the GDDR6 reference clock and the DC interface AXI clock based on the clock outputs available from the PLL. As the Speedster7t ACt1500 FPGA has eight GDDR6 subsystems, with a subset being connected directly to the fabric interface, there is the option of enabling the fabric interfaces based on the selected placement of the GDDR6 IP by enabling **Expose Channel 0/1 AXI Interface to Fabric Pins**. If any of these options are enabled, the IP Diagram window shows the corresponding pins from the GDDR6 subsystem that are exposed to the fabric including the clock and reset signals for each channel as well as the error/interrupt signals. The user design must use these clock and reset outputs from the GDDR6 subsystem to drive any GDDR6 DC interface logic in the fabric core.

   Note

   Although all eight GDDR6 subsystems support 2D NoC interfaces, only the middle two GDDR6 subsystems (GDDR6_[1/2/5/6]) on the east and west sides support direct connectivity to the fabric.

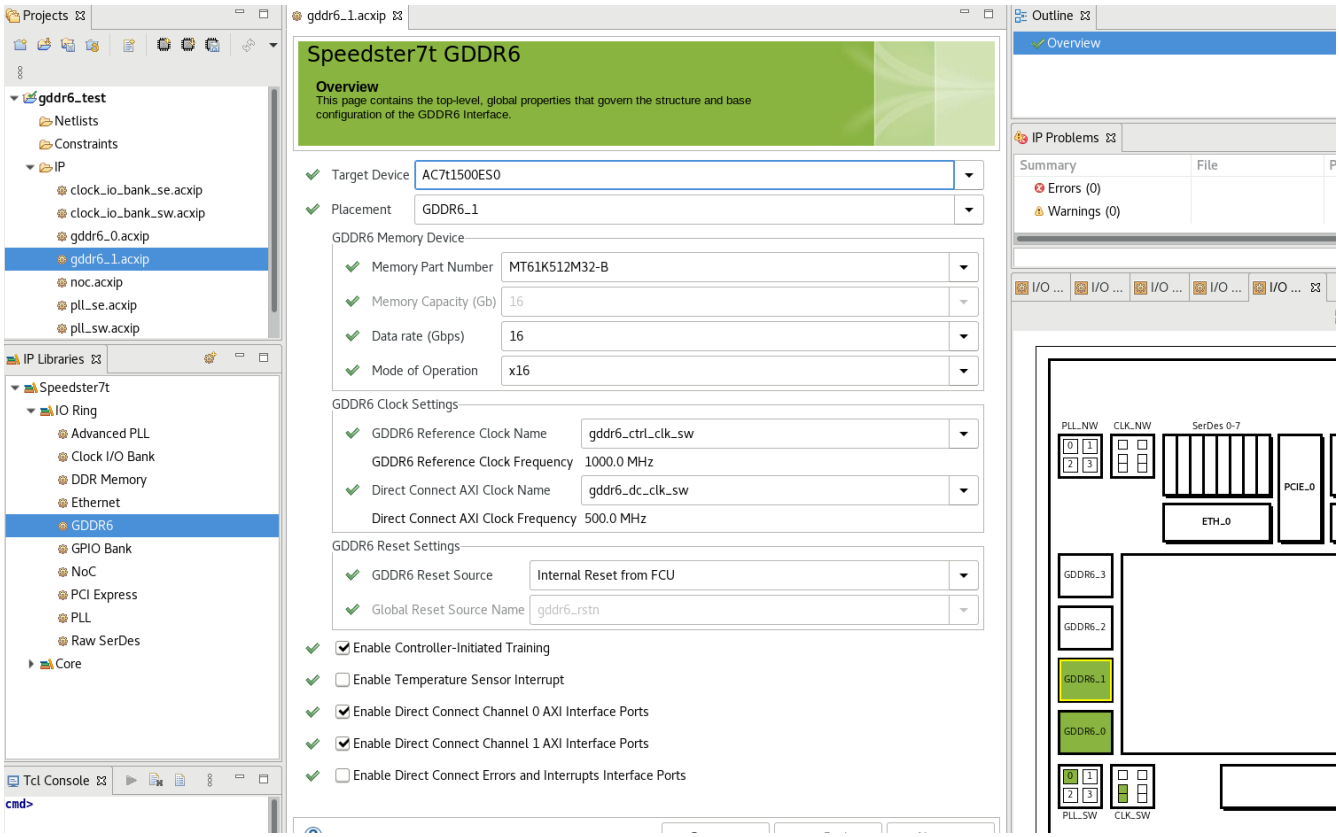


Figure 25: *GDDR6 Subsystem Configuration in ACE I/O Designer*

Clone a GDDR6 Instance (Optional)

If the user design employs multiple GDDR6 subsystems, an existing GDDR6 subsystem can be cloned by right-clicking the appropriate GDDR6 IP configuration and selecting **Clone IP**. Since the cloned subsystem is placed in the same location as the original, ensure the cloned instance is configured individually later with the appropriate change in placement of the instance. The interface pins and signals for the new IP instances can be seen in the IP Diagram window.

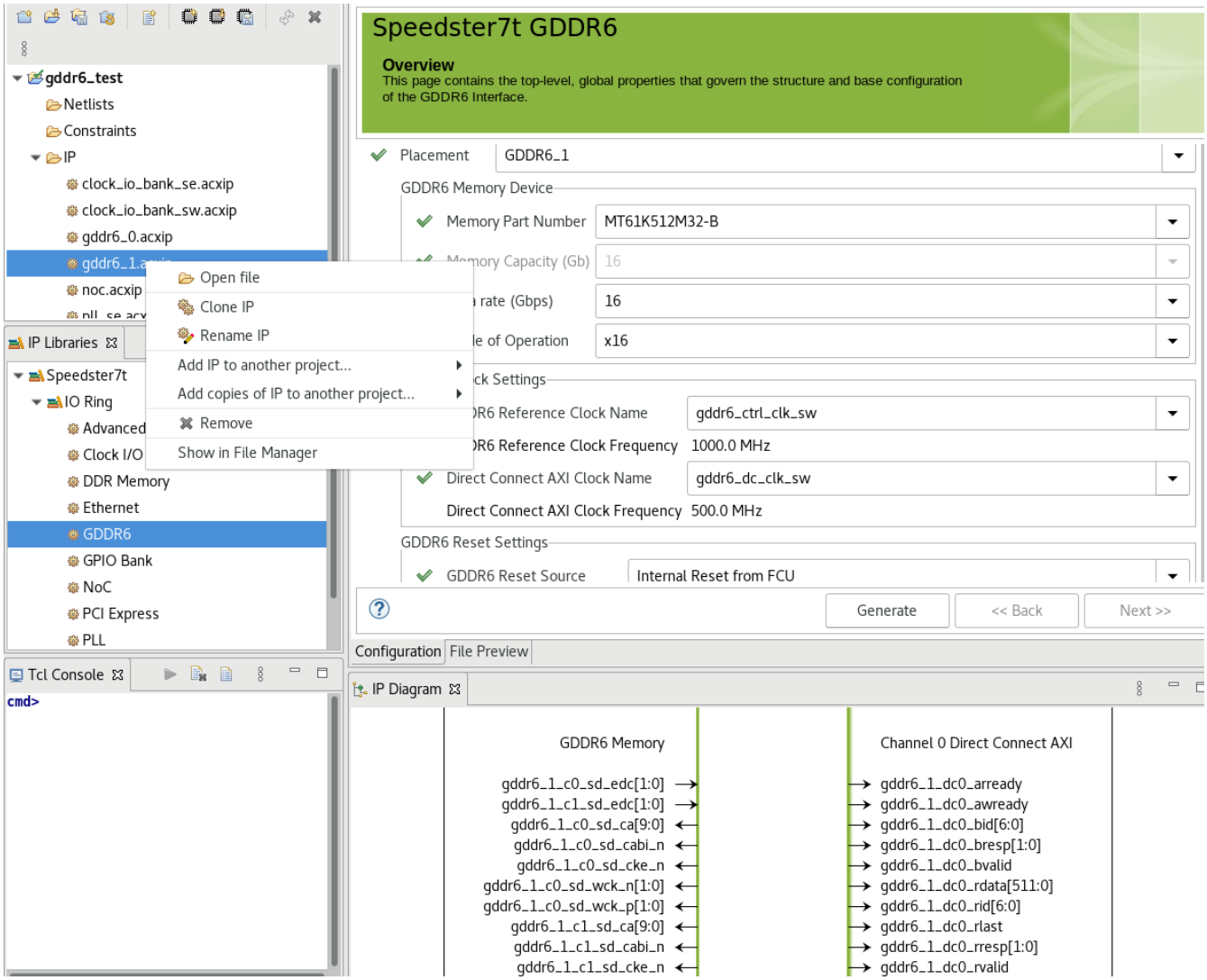


Figure 26: ***Cloning an Existing GDDR6 Subsystem***

Configure the Achronix Device Manager

It is necessary to configure the Achronix Device Manager IP which enables GDDR6 memory training and initialization.

1. In the IP Libraries window, select **Speedster7t → Core → Device Management → Device Manager**.
2. Connect this soft IP to the desired NAP row and column location.
3. Select the **Generate** action to generate the design file in the specified location as shown:

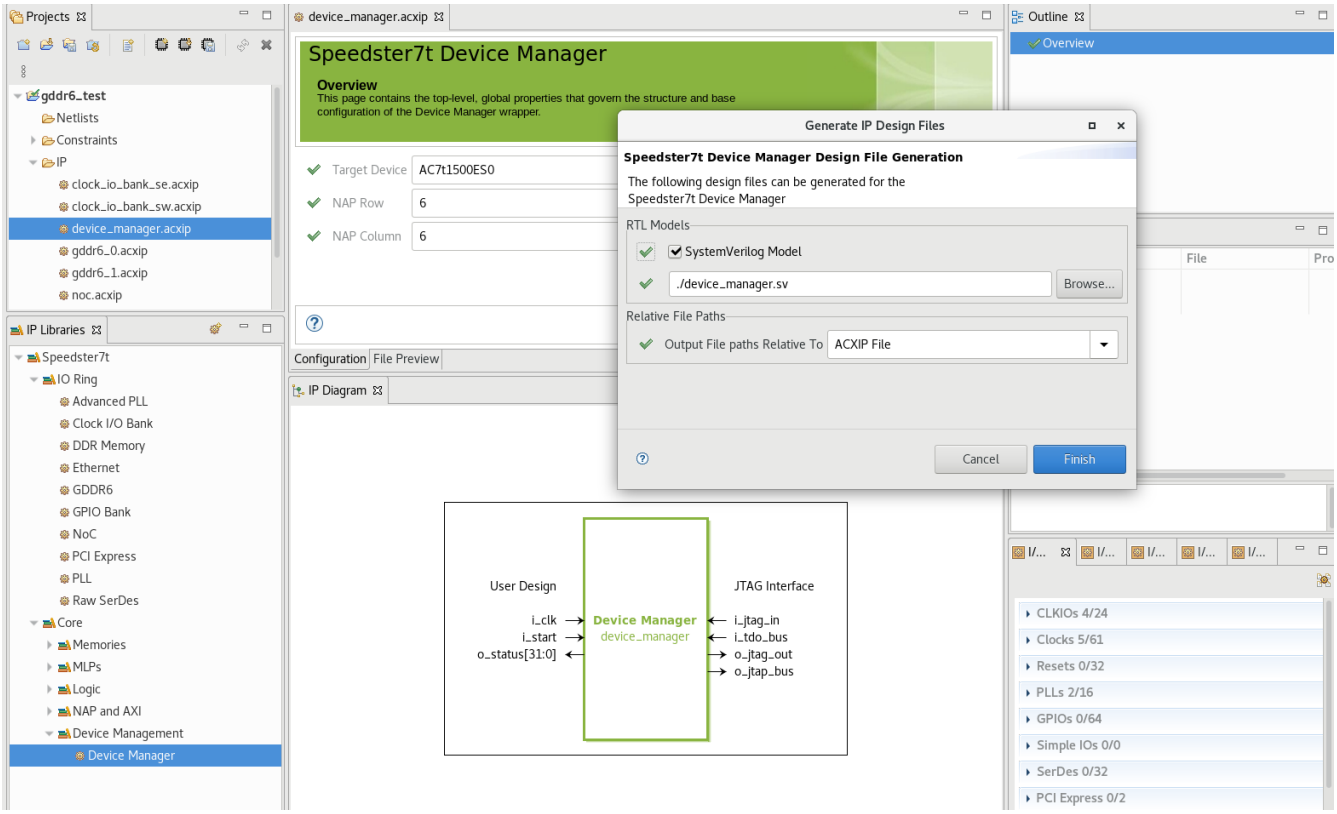


Figure 27: *Speedster7t FPGA Device Manager ACE Configuration*

The generated device manager template file must be instantiated in the top level RTL of the user design.

> **Note**
>
> The updated top level design file must be synthesized to generate a compatible netlist which is fed to ACE for final device bitstream generation.

More details and in-depth information on the Device Manager can be found in the "Speedster7t Device Manager" section of the *Speedster7t Soft IP User Guide* (UG103).

Check for Errors and Generate Files

After all configuration options are selected, the IP Problems window reports any errors or warnings that occur with the configuration. If there are no errors reported, the entire I/O interface with all of the required IP should be integrated properly and should close timing at the required clock frequency.

When these checks are completed, click **Generate IO Ring Design Files** in the I/O Designer window or the **Generate** option for any of the `.acxip` files. This action starts the process to produce the collateral needed by ACE bitstream generation for hardware implementation.

Files are generated in a folder chosen by the user. The default location is `<implementation_directory>/ioring_design`.

The files generated are:

- SDC files with timing constraints for all clocks exposed to the fabric and the GDDR6 channel clock outputs (if the DC interface is enabled)
- PDC files with pin placements for all GPIO pins and GDDR6 DC interface pins if enabled.
- ACE configuration files and port list files used for design simulation.

This step completes the I/O ring configuration. The constraint files and bitstream files needed by ACE are added to the project automatically if **Add to active project** is selected.

Conclusion

This core design can now be integrated by adding the netlist file in ACE. Along with the bitstream files generated for the I/O interface, the final full-chip integrated GDDR6 bitstream is obtained by running the complete ACE bitstream generation flow. For further details on the usage of ACE, refer to the *ACE User Guide* (UG070).

Achronix also provides reference designs that contain design source code, runtime scripts, ACE bitstreams and full documentation. These designs can be leveraged to build and simulate custom applications. The Speedster7t FPGA GDDR6 reference design can be found in Reference Designs for Speedster7t.

Revision History

Version	Date	Description
1.0	11 Oct 2019	• Initial release.
1.1	17 Apr 2020	• Updated the chapters, GDDR6 IP Software Support in ACE (see page 42) and GDDR6 Core and Interface Signals (see page 38), to align with latest tool capabilities. • Added details on the NoC Addressing Scheme for the GDDR6 interfaces to chapter, GDDR6 Interface Connectivity. (see page 28) • Other minor updates and edits.
2.0	21 Feb 2023	• Added support for AC7t800. • Remove controller features that are not controllable by user. • Updated GDDR6 Clock and Reset Architecture (see page 20) with details on how to drive all GDDR interfaces with single reference input clock. • Added GDDR subsystem control ID and ACE interface mapping for 2D NoC. • Added GDDR subsystem DC channel mapping in ACE. • Updated GDDR6 IP Software Support in ACE (see page 42) to align with latest ACE tool capabilities. • Other minor edits and updates.