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 Speedster7t FPGA device family provides multiple GDDR6 subsystems that enables the user to fully utilize 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 Speedster7t device. For example, the Speedster7t1500 device 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 7t1500 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 a Speedster7t FPGA. For resource counts for other Speedster7t family members, refer to the *Speedster7t FPGA Datasheet* (DS015).

> **Note**
>
> The following sub-sections in this user guide pertain to the 7t1500 device with eight GDDR6 subsystems.

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.
- **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 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.
- **System Configurable Modes** – The subsystem can be configured as either ×16 mode or ×8 clamshell mode for increased memory density applications.
- **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 master/slave ZQ calibration.
- **AXI4 Interface** – Connects to the other IP interfaces within the Speedster7t device 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 network on chip (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 diagram below shows the architecture of Achronix's 7t1500 FPGA. The eight GDDR6 subsystem are distributed four on the east and west sides each of the fabric. There are PLLs on four corners of the device that supply the external reference clock to the GDDR6 SDRAM cores and other high-speed interfaces that connect with the peripheral NoC over the FPGA fabric.

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

- NoC interface – By using the network hierarchy that allows high-speed data flow between FPGA and peripheral interfaces. All the eight GDDR6 subsystems can accessed from the FPGA fabric through the NoC.
- 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 (namely GDDR6 1, 2, 5 and 6 from the diagram below) that connect to the FPGA fabric directly.

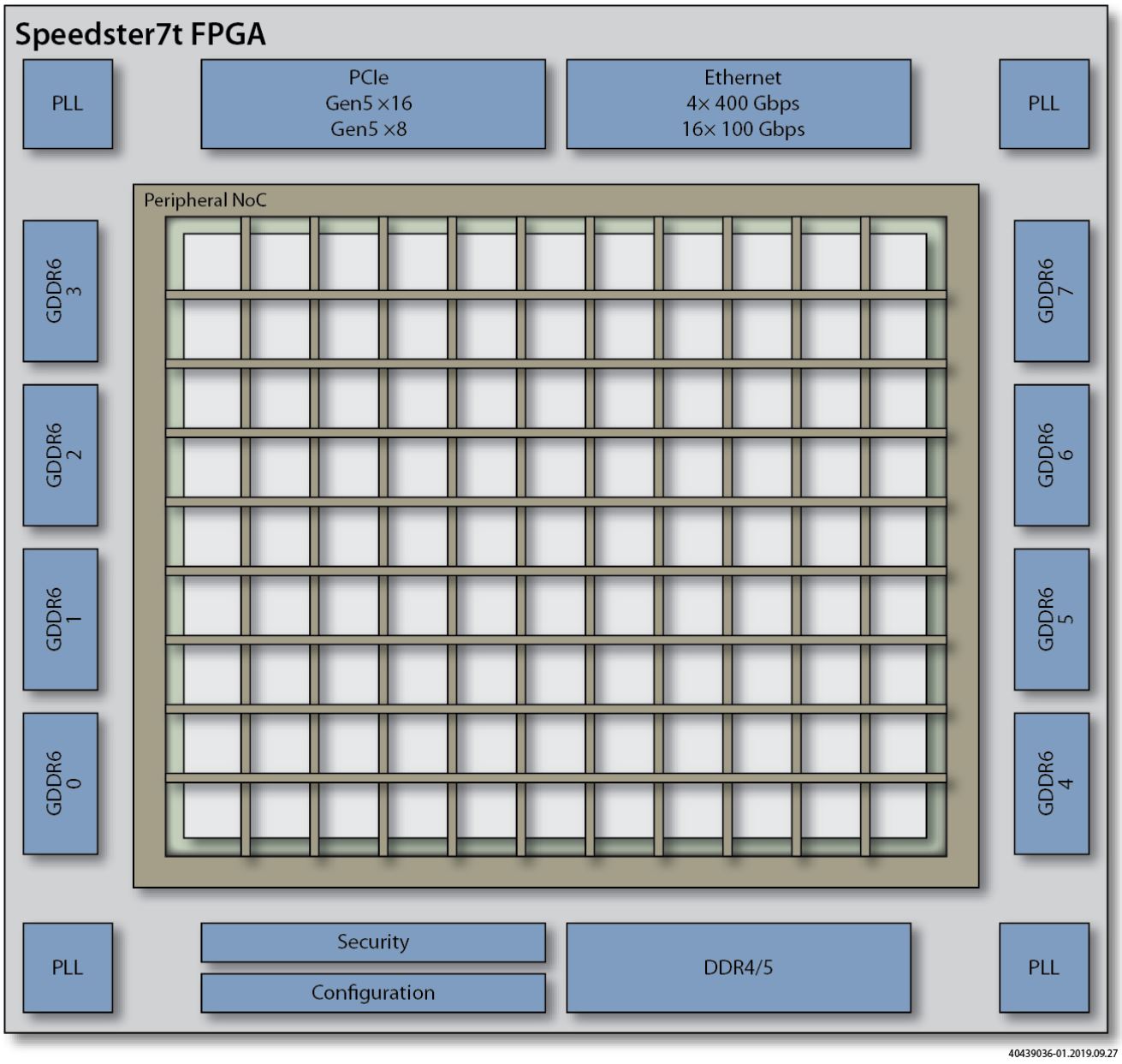

Figure 1: ***Speedster7t1500 Architecture Overview Block Diagram***

GDDR6 Subsystem Overview

The GDDR6 subsystem provides a simple interface between off-chip GDDR6 memory component and the user logic mapped to the FPGA core. This memory subsystem comprises the PHY IP, the controller IP, clock and reset block, APB interfaces and AXI4 interfaces to connect to the NoC and fabric. Below is a block diagram of the GDDR6 subsystem.

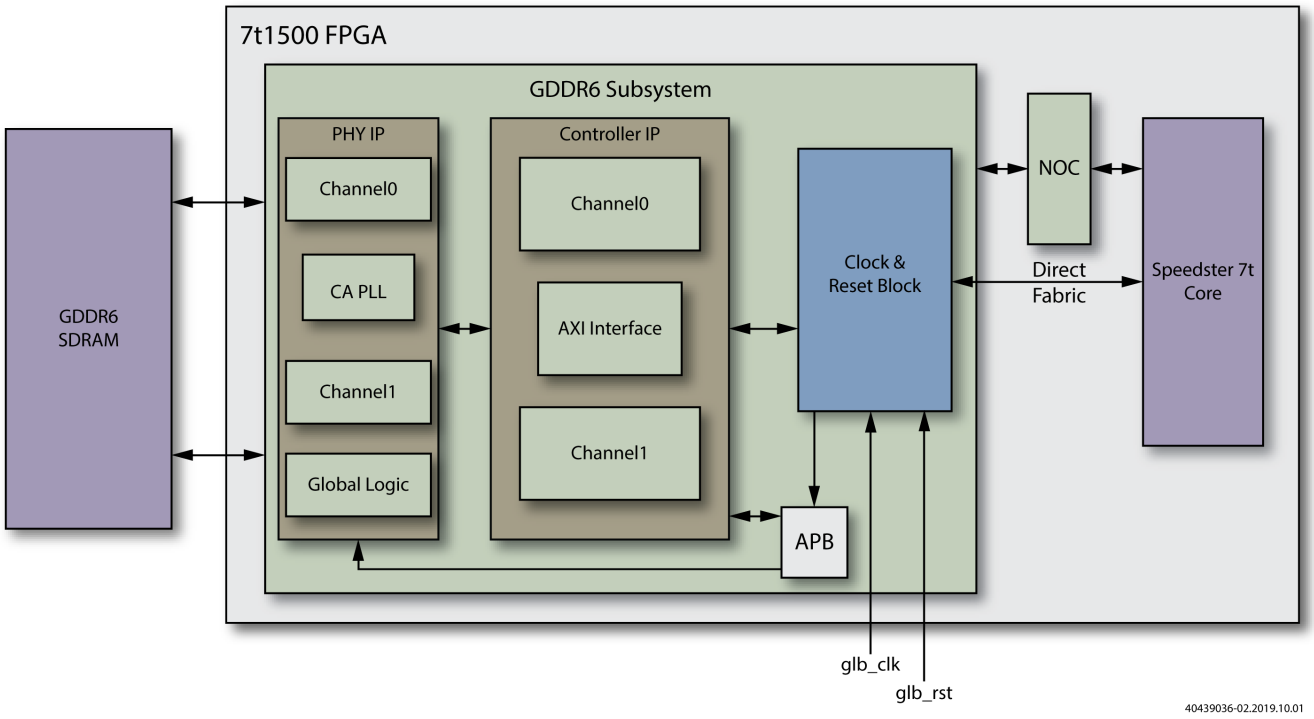

Figure 2: ***Speedster7t GDDR6 Subsystem Block Diagram***

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, Channel0 and Channel1 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 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 described below:
    - Memory read – To perform a read, a 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 such a burst of data into multiple consecutive transactions.
    - Memory write – To perform a write, a 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 a data request sent to the PHY
    - **AXI4 Slave Interface** – The AXI4 slave 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 a to the Speedster7t 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 enables the user to configure the GDDR6 subsystem registers. The subsystem registers are configurable through the APB slave interface where the master can be from the fabric or FPGA configuration unit (FCU) through the NoC. The FCU configures the subsystem registers during boot-up, and the user can configure these registers from the fabric during user mode.

Supported Frequency Table

The table below charts out the rates at which each of the interfaces in the GDDR6 subsystem operate:

Table 1: ***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 Speedster7t 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's 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 enables the controller to look ahead into the queue to perform operations and precharges in advance and 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 2: *GDDR6 Controller Features*

Feature	Description
Maximum frequency	The controller supports GDDR6 operation at up to 16 Gbps.
Controller clock rate	Controller operates at half the rate of the command address clock.
Number of channels	Two independent channels; Individual channels can also be disabled.
Memory Initialization	The memory controller supports initialization using PHY-independent mode.This mode is selected by setting a register bit; each training is performed by the PHY without intervention by the controller. The controller performs power-up sequence and then directs the PHY to perform CA training, and finally it enables word clocks (WCK) and initializes the mode registers of the memory devices. The memory controller then directs the PHY to perform the remaining training steps at the end of which the PHY asserts training completion, and then the controller can begin issuing commands.
Queue depth	Reorder queue depth fixed to 64, it is 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 that monitors the user interface queue and examines the access requests, issuing activate, precharge and auto-precharge commands as soon as possible in order to maximize memory bandwidth and minimize latency.

Feature	Description
Data width	Supports GDDR6 ×16 mode or ×8 clamshell modes.
Write masks	Supports write single-mask and write double-mask operations.
Bus inversion	Supports GDDR6 data bus inversion (DBI) and CA bus inversion (CABI).
Refresh	Per-bank and all-bank refresh support.
User-controlled refresh	Optional user-control of per-bank refreshes.
Auto-precharge	Read/write commands may be issued with or without auto-precharge.
Bus utilization optimization	Read/write commands automatically promoted to use auto-precharge to improve bus utilization: • if page-miss requests are detected in the queue • if no page-hits are detected in the queue (programmable option)
Page-hit mitigation	Memory refresh operations are delayed to prevent interrupting page-hit operations so that command requests to same page are not separated for long by refresh cycles.
Error detection	Supports GDDR6 error detection code 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 to allow the controller to determine if the data transfer was completed correctly.
Error interrupt	Mask-able interrupt outputs for all detected error conditions, with corresponding CSR read and clear-on-write registers.
Error retry	If the controller determines that an error in a read or write data transfer has occurred, the retry logic is enabled. The read or write request will be retried, and the Error Detection and Correction (EDC) results will be rechecked until the 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 figure below shows the memory controller and its sub-modules multi-port front-end, reorder, read-modify-write, memory test and memory test analyzer cores. These blocks offer higher efficiency and throughput by re-ordering controller commands and also provide for test and debug capability:

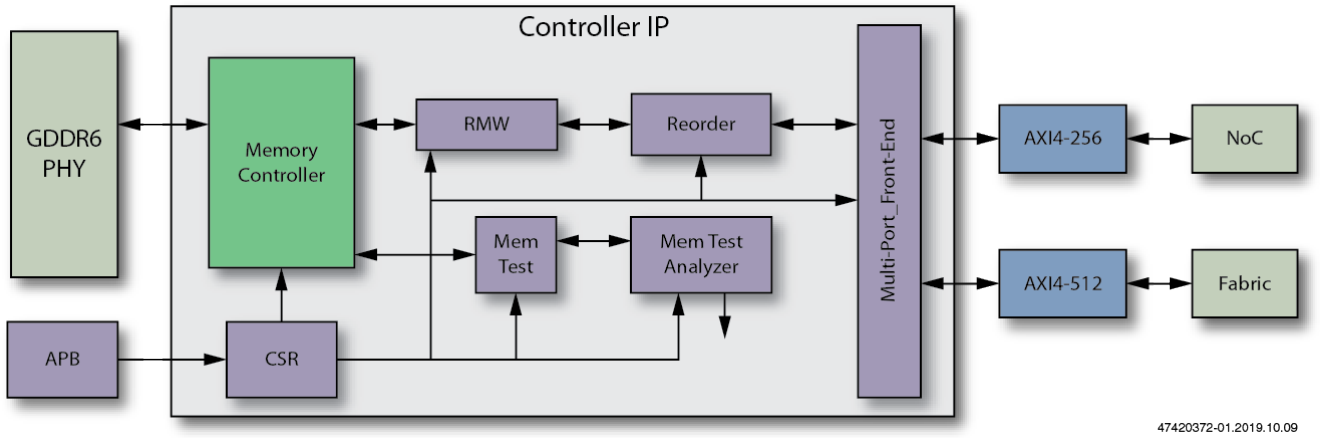

Figure 3: ***GDDR6 Controller IP Block Diagram***

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 Channel0 and Channel1 controllers where one port is driven by the NoC interface and the other is driven directly by the fabric.
- **Reorder Core** – This submodule is used in conjunction with 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 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. It 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 a status to the user. It can also be used to capture memory test signals of interest.
- **Memory Controller Core** – This queue-based, high-performance interface helps the controller to perform 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** – Provides AXI4 interfaces. The controller can access either the 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 slaves in the GDDR6 subsystem, one per controller and one for the PHY. It also includes few register maps to enable clock and reset functionalities. The clock and reset of APB slaves are connected by CSR signals. The last APB slave is connected to IPCNTL components.

Modes of Operation

The Speedster7t 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. A block diagram of the dual-controller system using a single memory device in ×16 mode is shown below:

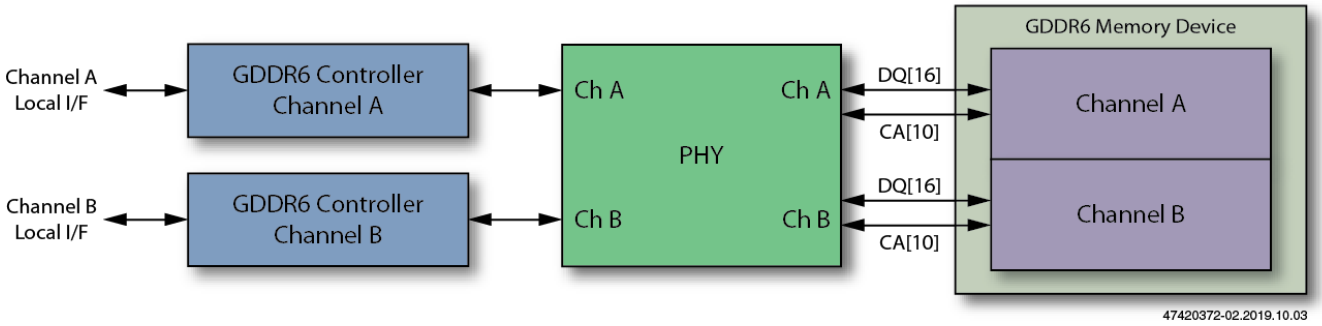

Figure 4: ***Dual-Controller ×16 Mode Block Diagram***

By 8 Clamshell Mode

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

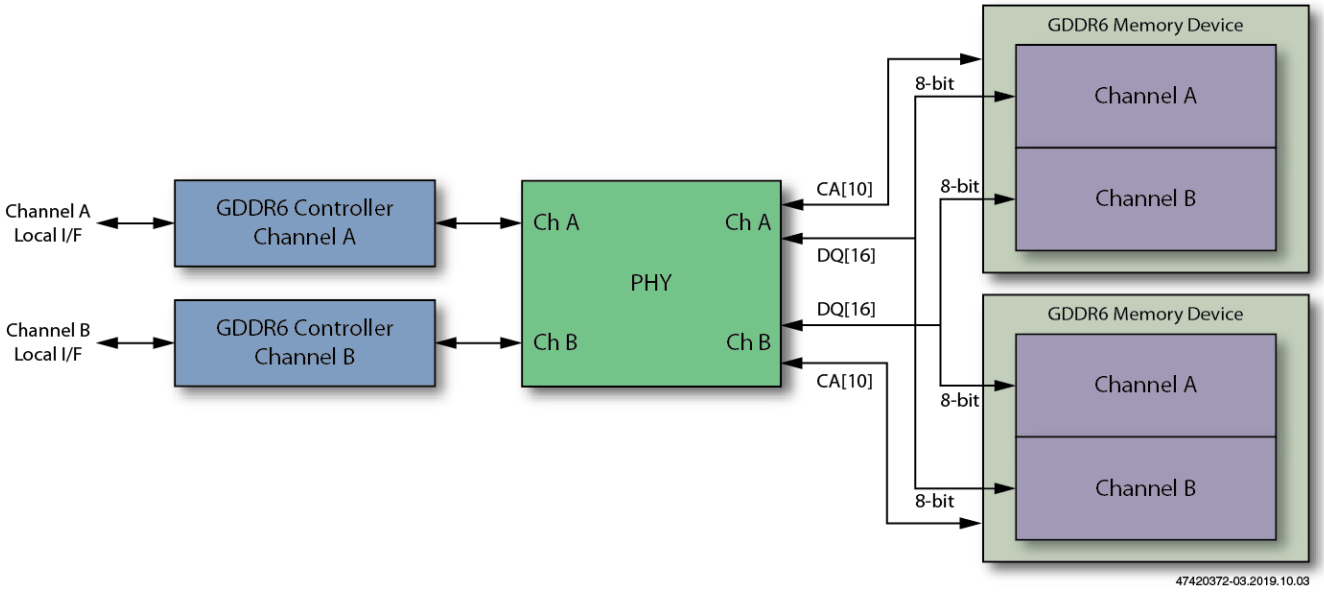

Figure 5: ***Dual-Controller ×8 Clamshell Mode Block Diagram***

Chapter - 3: GDDR6 PHY Architecture

PHY Overview

The embedded Speedster7t GDDR6 PHY supports the GDDR6 memory standard at the channel interface and DFI-4.0 interface on the FPGA side with the memory controller. It 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 figure below shows the PHY interfacing with the off-chip GDDR memory on one side and the memory controller on the FPGA side.

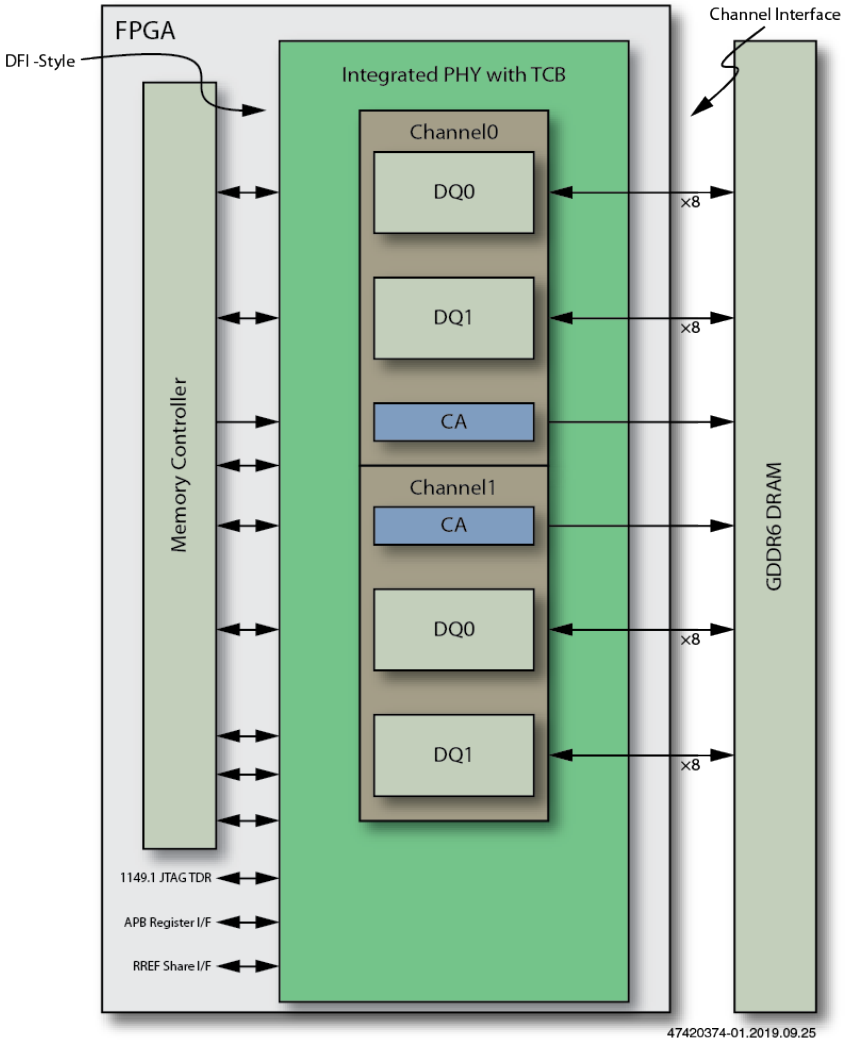

Figure 6: ***Speedster7t GDDR6 PHY Block Diagram***

PHY Features

Table 3: *Speedster7t GDDR6 PHY Features*

Feature	Description
DRAM density	SDRAM Density up to 16 Gb per component supported.
DRAM speeds	The 7t1500 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 an clock input to the PHY and PCLK: CK frequency ratio is fixed to 1:2
Command address bus inversion (CABI)	Supports CABI where each controller has a bit to enable 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 bit for DBI.
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)	There is a bit per DQ byte for EDC.

PHY Architecture

The PHY consists of the command/address (CA) block, DQ byte blocks, PLLs and the global logic block. 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 controller in 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 the channels of DRAM. Different DRAM data rates are supported with appropriate PLL multiplier and post-divider ratios. Reference clock and DFI clock is provided by the FPGA. The internal or local PCLK is aligned to FPGA PCLK using a DLL.

DQ PLL

The DQ PLL (one per ×16 interface) handles high-speed clock (WCK/WCKN) generation using PLL. WCK /WCKN is present per byte or per word as 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 using DFI status interface.

The figure below shows a high-level PHY I/O diagram:

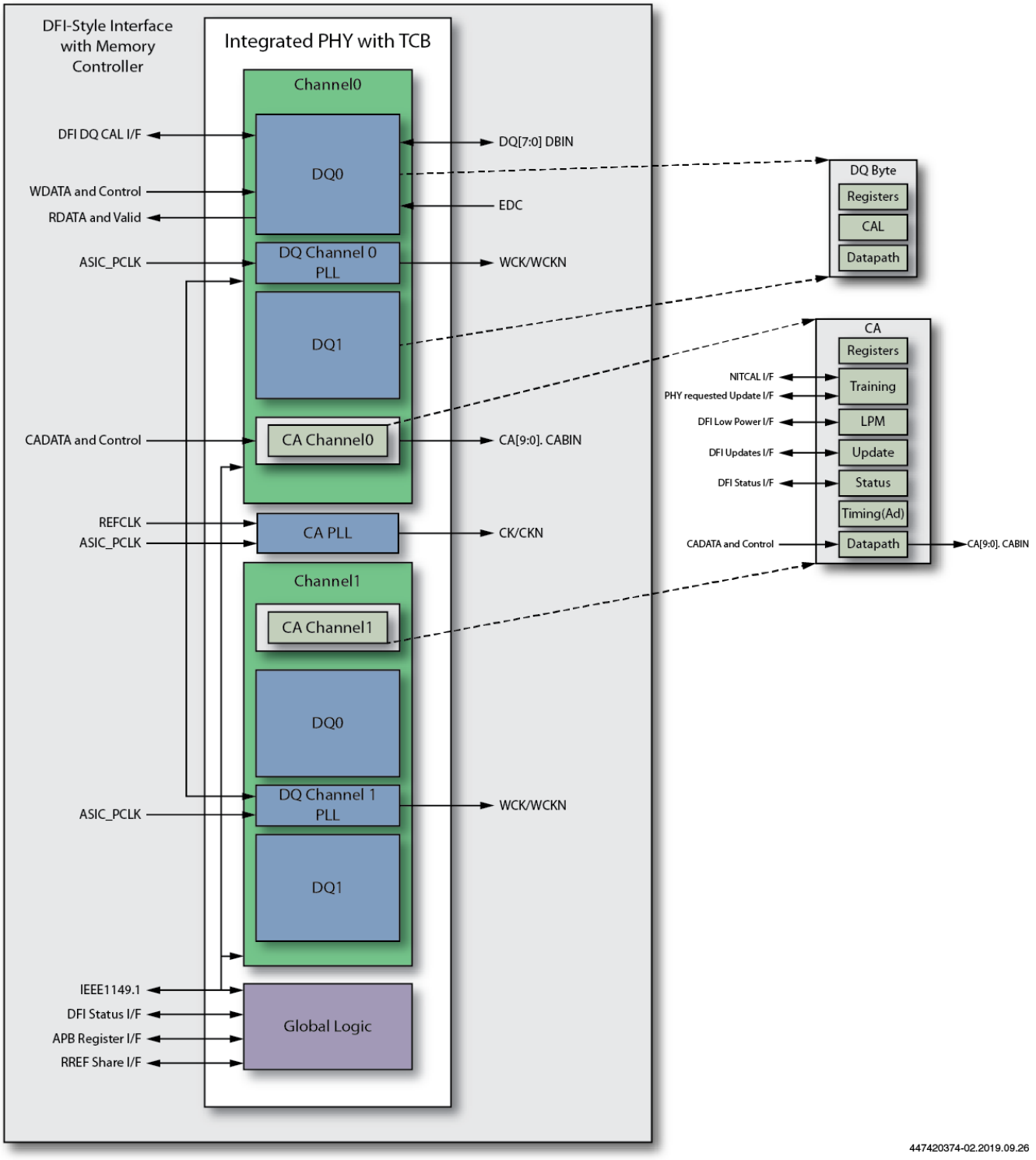

Figure 7: ***High-Level PHY I/O Block Diagram***

Chapter - 4: GDDR6 Clock and Reset Architecture

The Speedster7t GDDR6 subsystem requires an external input reference clock and reset signals to drive the subsystem. The clock and reset generator module in the FPGA, 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 device. Each clock and reset generator has four PLLs, with each PLL capable of generating up to four clock outputs; hence, 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 user can choose the subsystem clocks 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 device 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:

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

Similarly, there are 32 global resets generated by the clock and reset generators. In addition, 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:

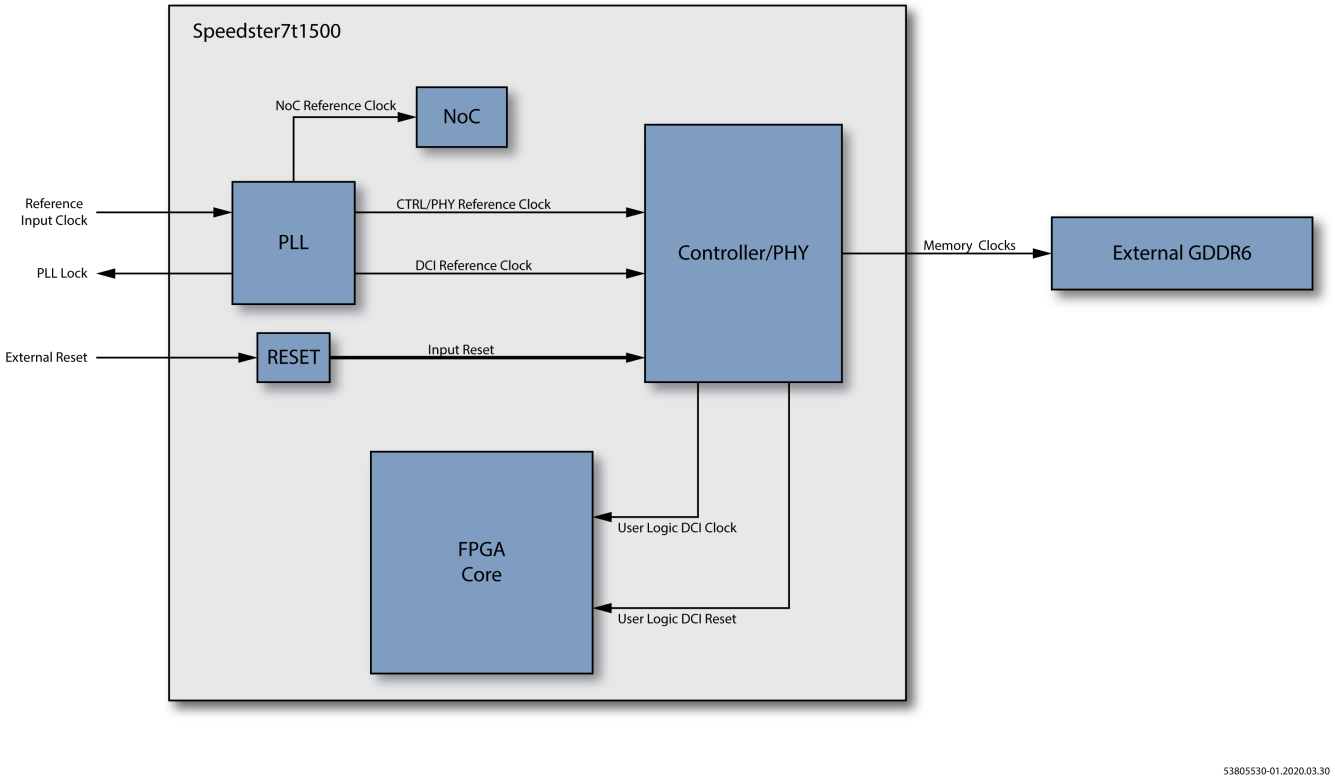

Figure 8: ***Clock and Reset Architecture of GDDR6 subsystem***

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 also comes from the same clock source as the controller clock. The PHY's internal PLL generates the CA and WCK memory clocks. The PHY's maximum rate of operation is 500 MHz.

The AXI4 interface requires two asynchronous clocks selected separately by the user in ACE I/O designer. These clocks drive the 256-bit AXI4 interface connected to the peripheral NoC and the 512-bit AXI4 interface connected to the fabric. The clock driving the user logic for the GDDR6 NoC interface is handled internally in the 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 NoC and DC interface reference input clock rates are independent of the controller/PHY clock rate; users can scale these clocks based on their throughput requirements.

The AXI interface clocks are routed through the GDDR subsystem and then made available to the 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 user exercises the NoC interface, 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.

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 generates clocks for the west GDDR6 controllers. There is no clock domain crossing between the east and west side GDDR6 controllers, which helps to achieve the maximum data rates across all eight GDDR6 interfaces.The resets for all controllers can be tied to an external reset.

The figures below shows how a user can connect the reference input clocks to drive all eight GDDR6 controllers. The preferred configuration is based on the jitter and skew assessments of the user design.

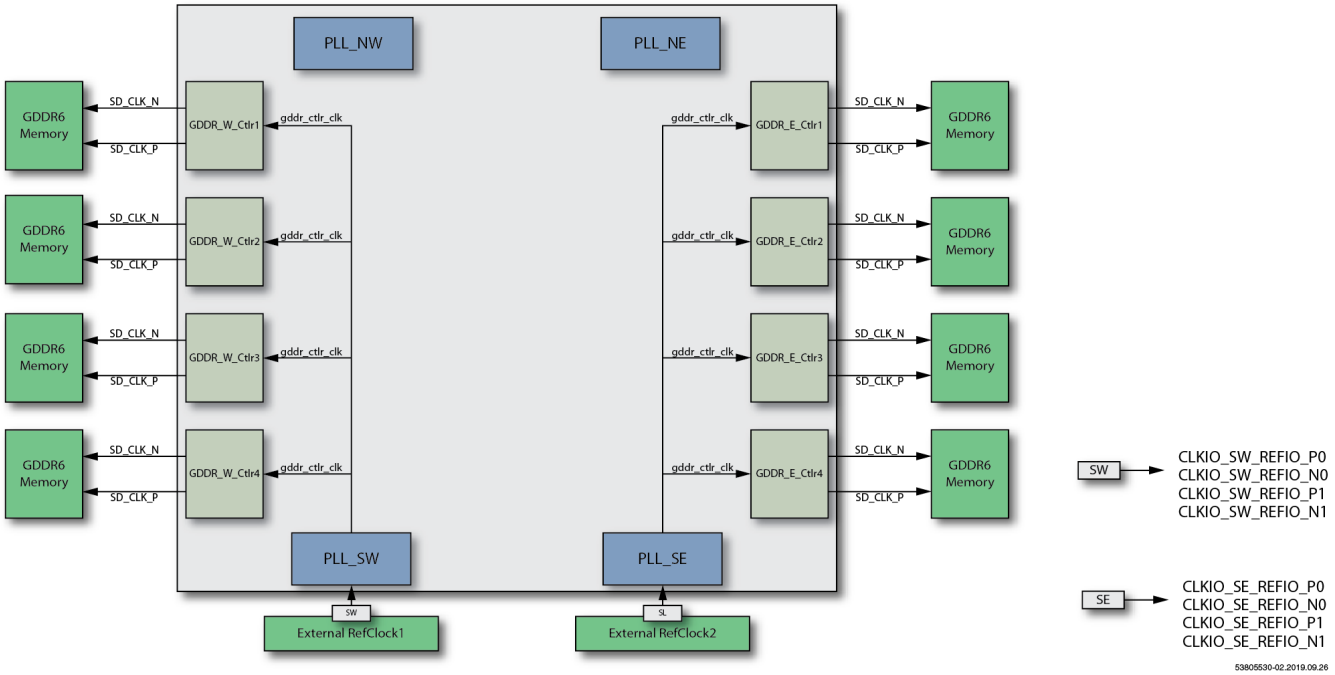

Figure 9: ***External Clock to South PLLs Driving the GDDR East/West Subsystems on the AC7t1500 device***

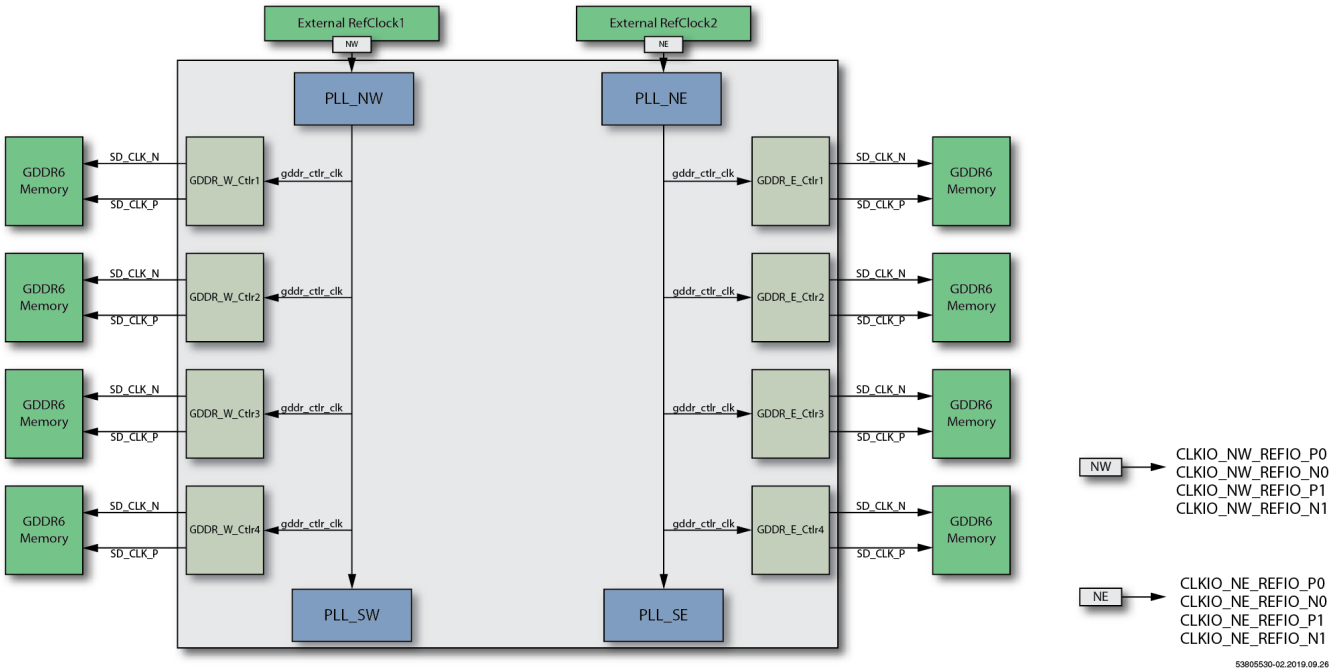

Figure 10: ***External Clock to North PLLs Driving the GDDR East/West Subsystems on the AC7t1500 device***

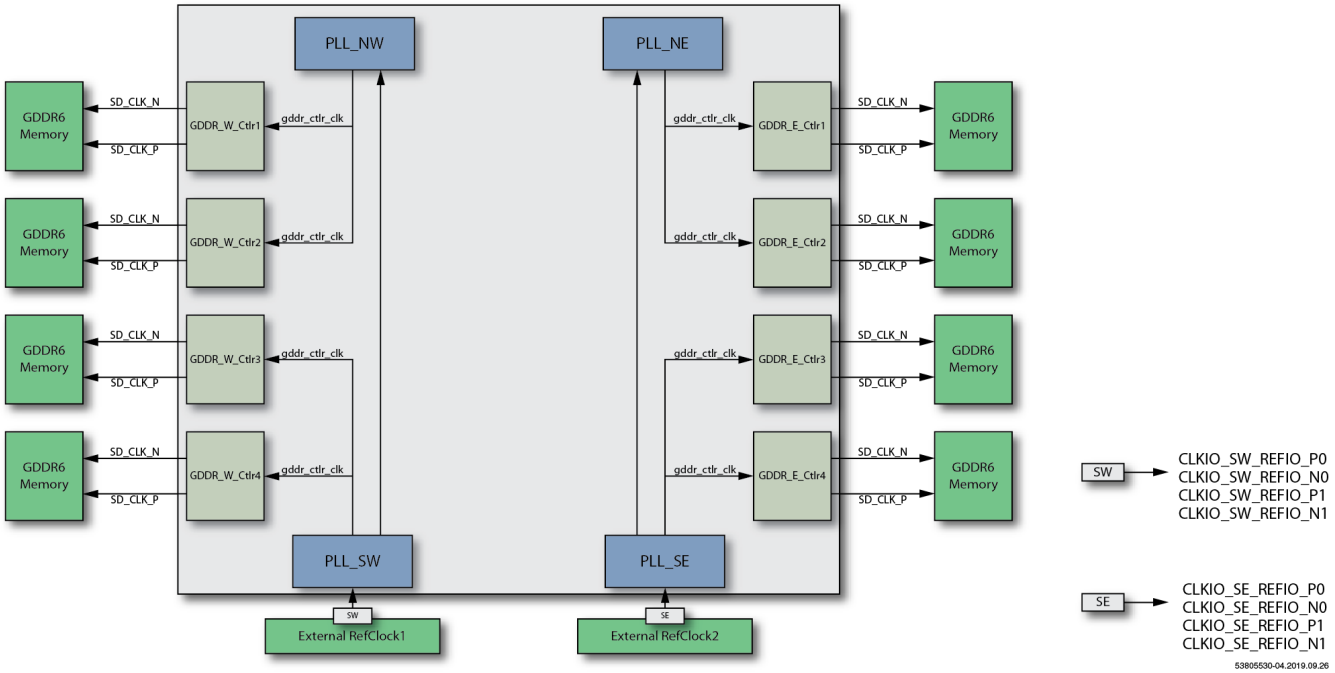

Figure 11: ***South PLLs Driving Some of the GDDR East/West Subsystems and the North PLLs on the AC7t1500 device***

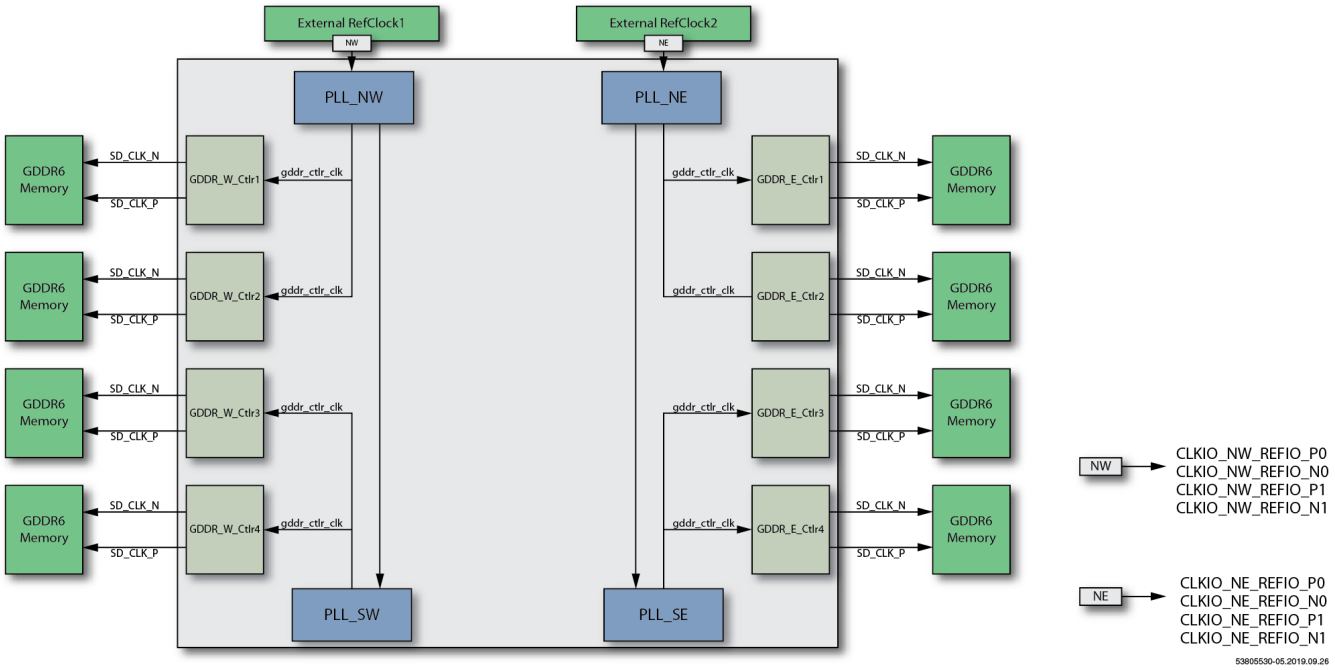

Figure 12: ***North PLLs Driving Some of the GDDR East/West Subsystems and the South PLLs on the AC7t1500 device***

Chapter - 5: GDDR6 Interface Connectivity

The following sections describe the two interfaces supported by the GDDR6 subsystem to connect to the user logic hosted in the FPGA fabric, namely the NoC and the DC interfaces.

Connectivity to the Peripheral NoC

The Speedster7t FPGA family of devices has a network hierarchy that enables extremely high-speed dataflow 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 of 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. Achronix's network on chip (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 user can connect the GDDR6 subsystems to enable transaction with PCIe or FPGA fabric through the NoC interface. PCIe can initiate transactions to any GDDR6 channel using NoC. In this case, the PCIe endpoint is the master with the GDDR6 acting as slave. Similarly, the FPGA master logic can issue a transaction to it's local Network Access Point (NAP), which carries it to the east or west side of the FPGA core, where it is presented to the NoC. From then on, the NoC carries data to the appropriate GDDR6 interface. The responses follow the same path in reverse.

In addition, the 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 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.

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

- AXI4 256b is a 256-bit slave interface which is connected to NoC master AXI.
- The AXI4 256b works in single-clock mode, user needs to provide synchronous clock to both read and write port; it has 256-bit write and read data width.
- AXI4 converts AXI transactions to the local bus transactions which connect directly to the MPFE and has independent command, write and read data FIFO
- Supports all burst sizes, types, and lengths including incremental and wrapping bursts.

The figure below shows the I/O diagram of an AXI4-256b Interface. For more details on AXI transactions, see the AMBA AXI Protocol Specification

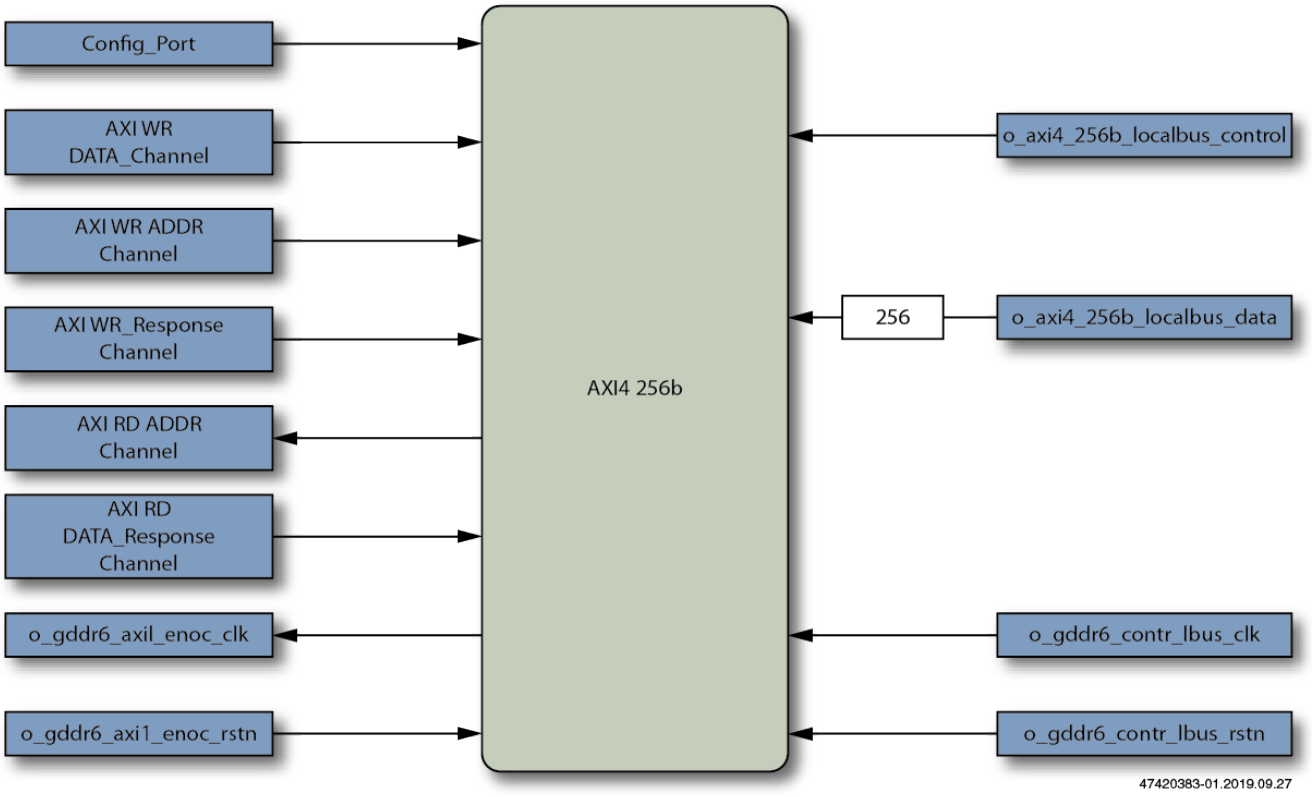

Figure 13: ***AXI4-256b I/O Diagram***

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

Table 4: ***AXI4-256b Interface Parameters***

Parameter	Value	Description
AXI_DATA_WIDTH	256	Data width of AXI4 interface
AXI_SLAVE_ID_WIDTH	7	Width of awid, wid, bid, arid, rid ports
AXI_CMD_FIFO_AWIDTH	3	Sets the depth of the read and write command FIFOs. The depth is equal to $2^{AXI\_CMD\_FIFO\_AWIDTH}$.
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 in the AXI4 Interface.

Parameter	Value	Description
ENABLE_INTRAPORT_REORDER	1	If set in this mode, 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 how they are issued by the AXI4 interface. In this case, for reads the AXI core must reorder the data coming from the memory so that it is presented to the AXI4 interface in the proper order. For writes the AXI core must present data to the memory controller in the order requested by the controller.
ADDR_MAP_SIG_BITS	10	Sets the number of significant bits that are used in the address value comparison (typically set to 10)
BURST_SIZE_WIDTH	13	Width of burst size
AXI_READ_ONLY	0	When set, disables write accesses through port and removes write logic
AXI_WRITE_ONLY	0	When set, disables read accesses through port and removes read logic
REQ_PRIORITY_WIDTH	3	Width of request priority attribute
SINGLE_CLOCK_MODE	1	Set when aclk and l_clk are the same clock domain, to optimize latency

The figure below shows PCI Express master issues a transaction to the NoC, which transmits it directly to the GDDR6 interface without involving any resources in the FPGA fabric at all.

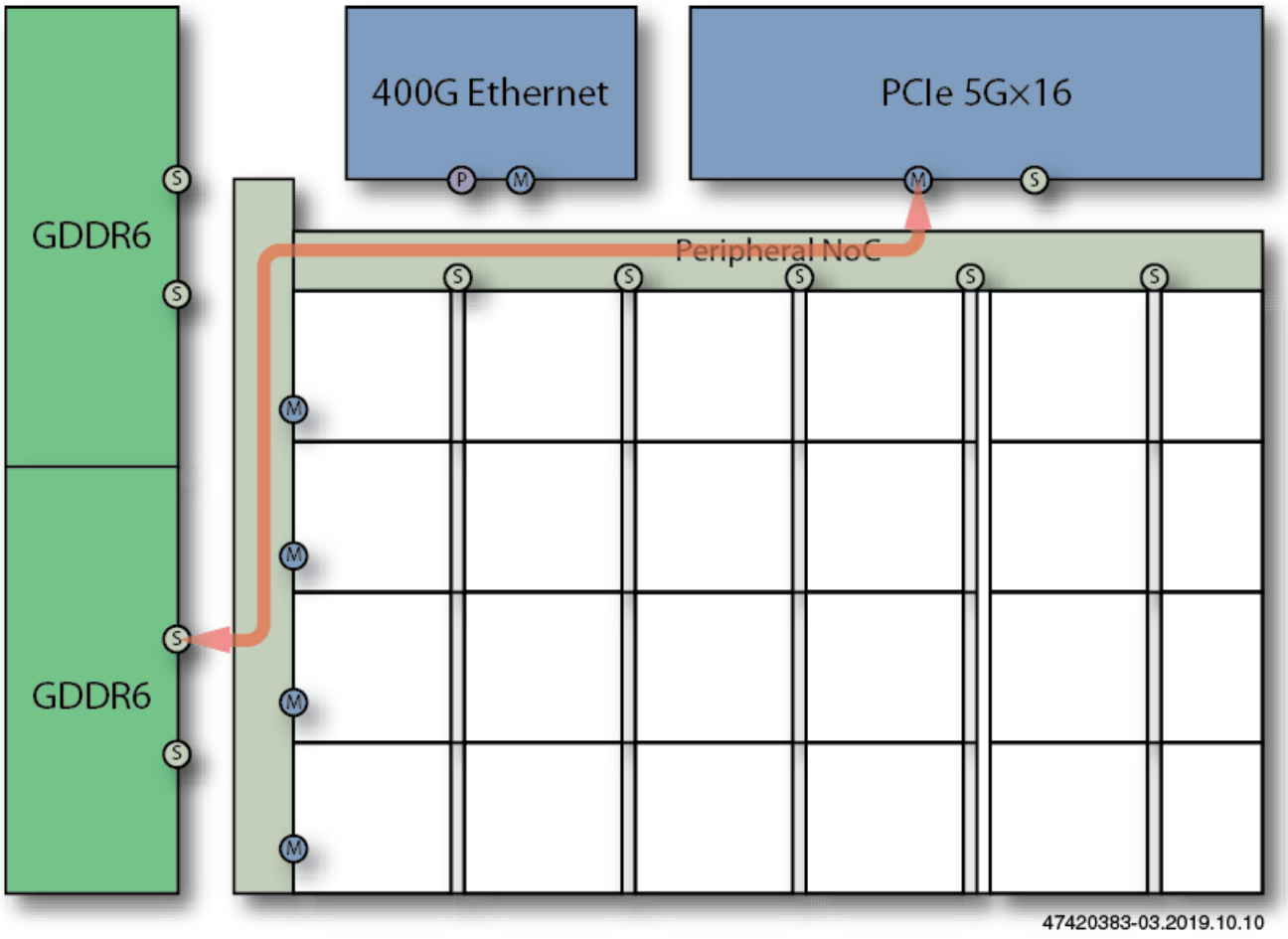

Figure 14: ***Data Flow from the PCIe Interface to GDDR6 Subsystem Through the NoC***

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

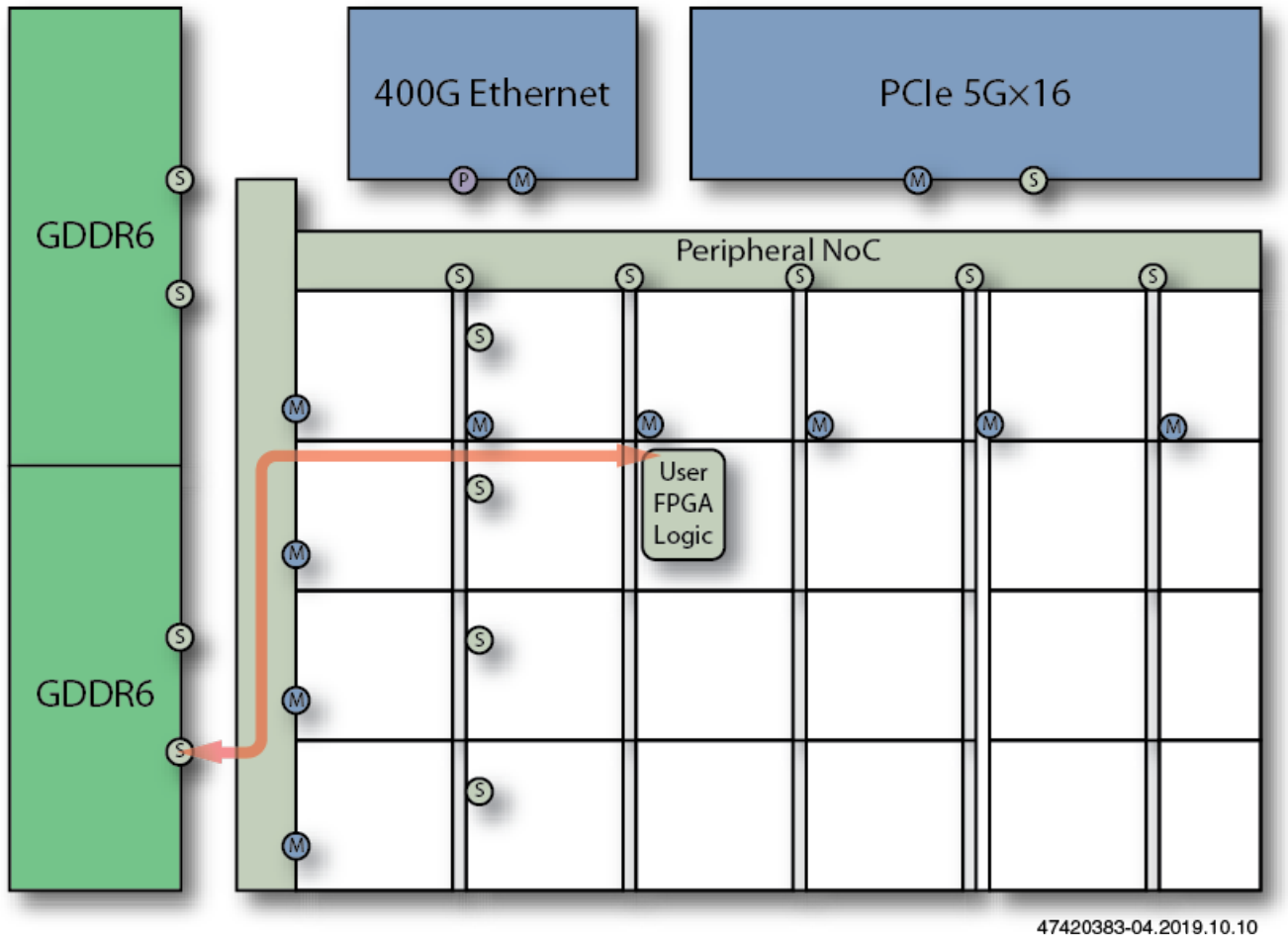

Figure 15: ***Data Flow from the FPGA Fabric to the GDDR6 Subsystem Through the NoC***

NoC Addressing for GDDR6

The table below shows how the GDDR6 NoC addressing is established.

Table 5: ***GDDR6 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 NoC GDDR6 addressing representation is as shown below:

- Addr[41:37] = `5'b00000`
- Addr[36:33] = **Ctrl ID** - Selects which of the eight GDDR6 controllers the transaction is destined for. The three most significant bits of this field select the controller, the least significant bit selects between the two channels on each controller.
- Addr[32:0] = **Memory Address** - The memory address for the specific controller and channel.

The mapping of the addresses is explained in the section GDDR6 Memory Address Mapping to AXI Addresses (see page 29).

Connectivity Through the DC Interface

There are only four GDDR6 subsystems, the middle two, on the east and west sides of the chip 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 supporting data rates of up to 16 Gbps:

- AXI4 512b is a 512-bit slave interface which is connected to fabric master AXI
- 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.
- The write clock is 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.
- AXI4 converts AXI transactions to the local bus transactions.

The figure below shows the I/O diagram of an AXI4-512b Interface:

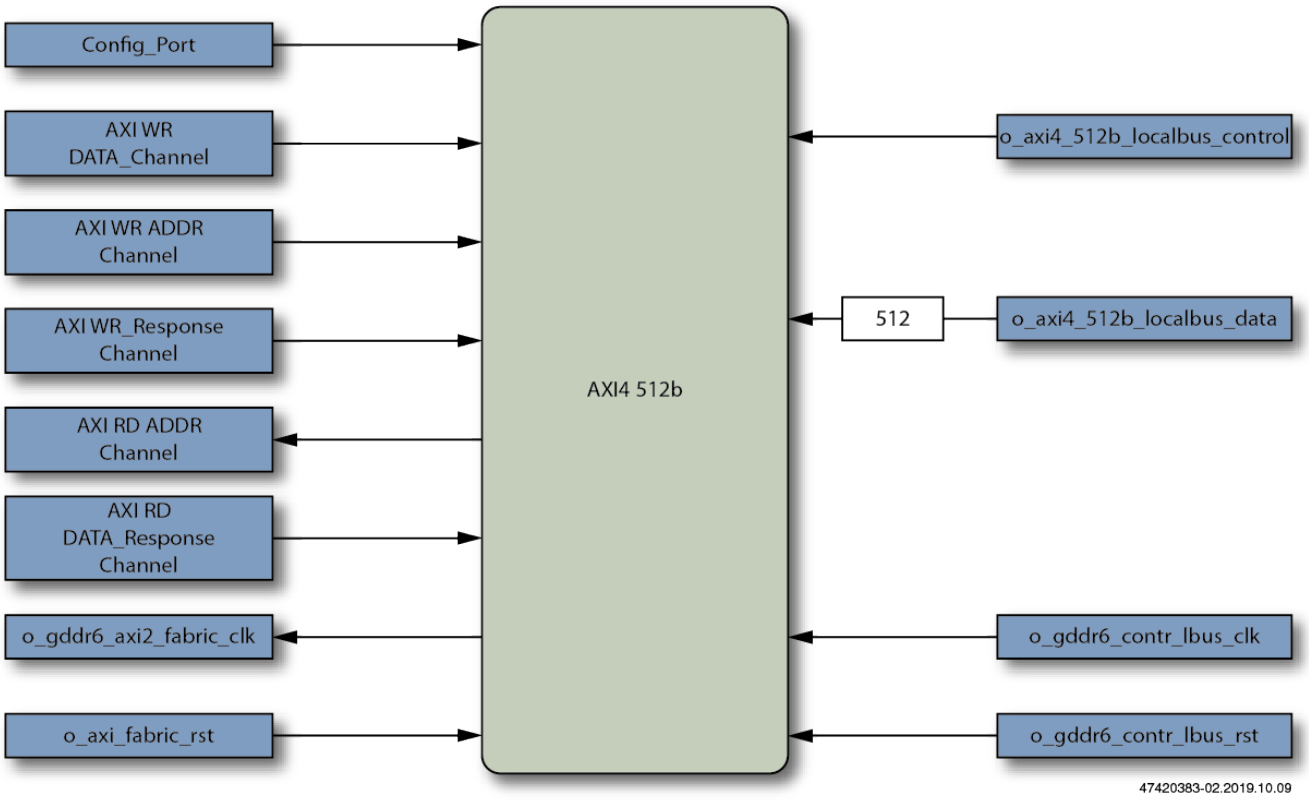

Figure 16: ***AXI4-512b I/O Diagram***

The following table provides the parameters to be given to the AXI4 Interface.

Table 6: ***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, rid ports.
AXI_CMD_FIFO_AWIDTH	3	Sets the depth of the read and write command FIFOs. The depth is equal to $2^{AXI\_CMD\_FIFO\_AWIDTH}$.
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 in the AXI4 interface.
ENABLE_INTRAPORT_REORDER	1	If set to this mode, 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 how they are issued by the AXI4 interface. In this case, for reads the AXI core must reorder the data coming from the memory so that it is presented to the AXI4 interface in the proper order. For writes the AXI core must present data to the memory controller in the order requested by the controller.
ADDR_MAP_SIG_BITS	10	This sets the number of significant bits that are used in the address value comparison (typically set to 10)
BURST_SIZE_WIDTH	13	Width of burst size.
AXI_READ_ONLY	0	When set, disables write accesses through the port and removes the write logic
AXI_WRITE_ONLY	0	When set, 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

GDDR6 Memory Address Mapping to AXI Addresses

The Speedster7t device 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. Below is an example of how the address mapping is done for a 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 will be set to zeroes. For the same 8 gb/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 will be set to zeroes. 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 the 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 driven into the fabric core so that the associated logic use these signals for clock and reset purposes. Each GDDR6 subsystem has a clock and reset signal per channel.

Table 7: *Clock and Reset Signals*

Pin Name	Direction	Width	Description
<prefix>_chan[0/1]_clk	Output	1	Output clock from GDDR6 subsystem to the fabric. This clock is 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 GDDR6 subsystem to the fabric. This reset is 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 8: *Error and Interrupt Signals*

Pin Name	Direction	Width	Description
<prefix>_chan[0/1]_crc_error	Output	1	Read or write CRC error signal for the corresponding channels to indicate an error when there is a CRC mismatch.
<prefix>_chan[0/1]_interrupt	Output	3	Interrupt output signal from the controller to perform further actions based on the error condition.
<prefix>_chan[0/1]_interrupt_or	Output	1	ORed output of all the interrupt signals connected to the fabric, where the user can take further decision on the errors.

AXI Interface Signals

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

Table 9: *Controller to AXI Interface Signals*

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

Pin Name	Direction	Width	Description
<prefix>_chan0/1_wlast	Input	1	Input signal to set the write last. This signal indicates the last transfer in a write burst.
<prefix>_chan0/1_wvalid	Input	1	Input signal to indicate if the write data channel signals are valid.
<prefix>_chan0/1_wready	Output	1	Indicates that a transfer on the write data channel can be accepted.
<prefix>_chan0/1_bid	Output	7	Identification tag for a write response.
<prefix>_chan0/1_bresp	Output	2	Write response output signal that indicates the status of a write transaction.
<prefix>_chan0/1_bvalid	Output	1	Indicates that the write response channel signals are valid.
<prefix>_chan0/1_bready	Input	1	Indicates that a transfer on the write response channel can be accepted.
<prefix>_chan0/1_arid	Input	7	Input signal to set the identification tag for a read transaction.
<prefix>_chan0/1_araddr	Input	33	Input signal for the address of the first transfer in a read transaction.
<prefix>_chan0/1_arlen	Input	4	Input signal to set the exact number of data transfers in a read transaction.
<prefix>_chan0/1_arsize	Input	3	Sets the number of bytes in each data transfer in a read transaction.
<prefix>_chan0/1_arburst	Input	2	Input signal to set the burst type. Indicates how address changes between each transfer in a read transaction.
<prefix>_chan0/1_arlock	Input	2	Input signal that provides information about the atomic characteristics of a read transaction.
<prefix>_chan0/2_arcache	Input	4	This signal indicates how a read transaction is required to progress through a system.
<prefix>_chan0/1_arprot	Input	3	This signal is used to set the following protection attributes of a read transaction: privilege, security level, and access type.
<prefix>_chan0/1_arvalid	Input	1	This signal indicates that the read address channel signals are valid.
<prefix>_chan0/1_arready	Output	1	This signal indicates that a transfer on the read address channel can be accepted.
<prefix>_chan0/1_arqos	Input	3	QoS identifier for a read transaction.

Pin Name	Direction	Width	Description
<prefix>_chan0/1_rid	Output	7	Identification tag for read data and response. The slave must ensure that the RID value of any returned data matches the ARID value of the corresponding address.
<prefix>_chan0/1_rdata	Output	512	256-bit wide read data from memory.
<prefix>_chan0_rresp	Output	2	Read response signal that indicates the status of a read transfer.
<prefix>_chan0_rlast	Output	1	This signal indicates whether this is the last data transfer in a read transaction.
<prefix>_chan0/1_rvalid	Output	1	This signal indicates that the read data channel signals are valid.
<prefix>_chan0/1_rready	Input	1	This signal indicates that a transfer on the read data channel can be accepted.

PHY – Memory Signals

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

Table 10: *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 the chosen IP for the user design. It also allows the user to select the placement for each individual IP and visualize package routing. Once 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 these bitstreams into a single configurable bitstream targeting a Speedster7t device.

The following steps provide a brief description on creating a GDDR6 IP interface design:

Step 1 – Create a Project

Create a project in ACE, and then in the 'Project perspective', select the target device **AC7t1500ES0** which ensures that the appropriate IP options are available in the IP Perspective window in ACE.

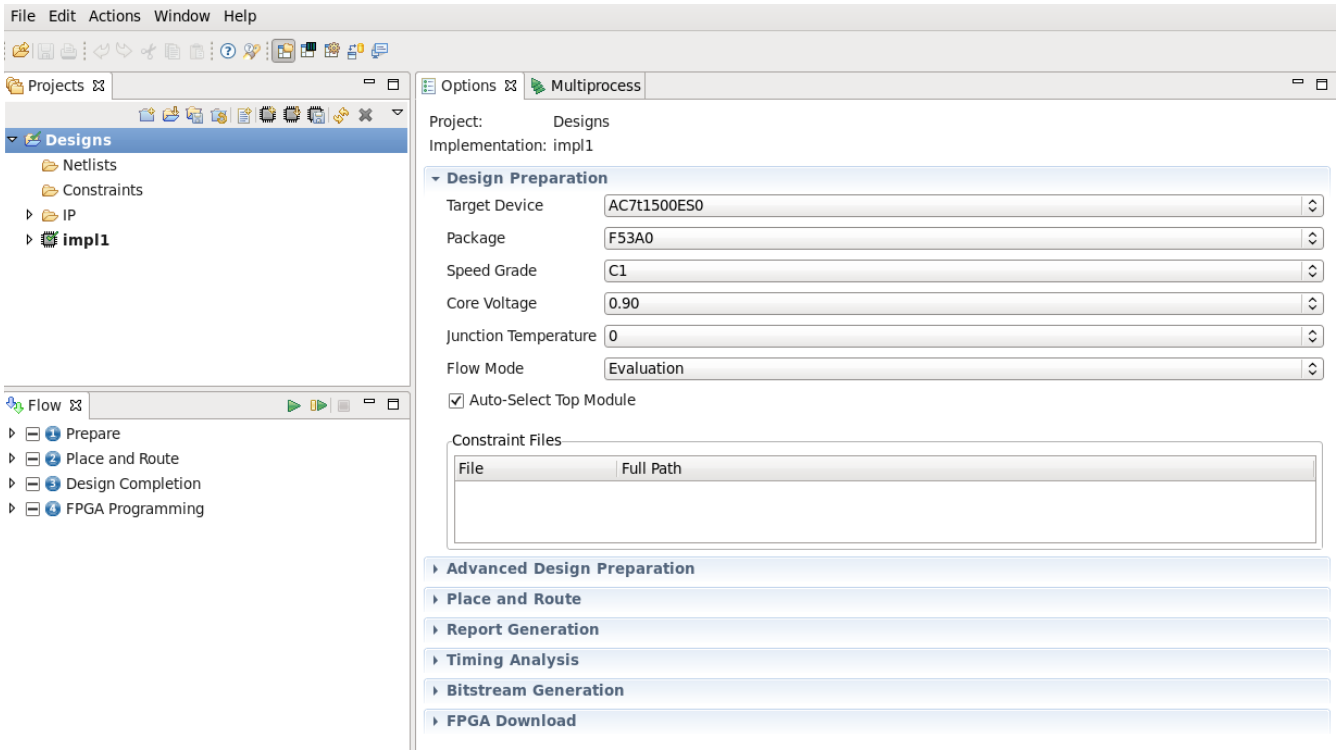

Figure 17: ***Design Preparation Options in the ACE Project***

Step 2 – IP Configuration and Placement

Switch to the 'IP Configuration' perspective and select **Programmable I/O IP** from the **Speedster7t → IO Ring** drop-down menu in the IP Libraries window. This selection creates an `.acxip` file that can be used to create the external input clock source and other top level I/O pins. Then select the ball placement for each of these pins and desired frequency for the clock input. Once 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 GPIO are highlighted in the 'IP Problems' window.

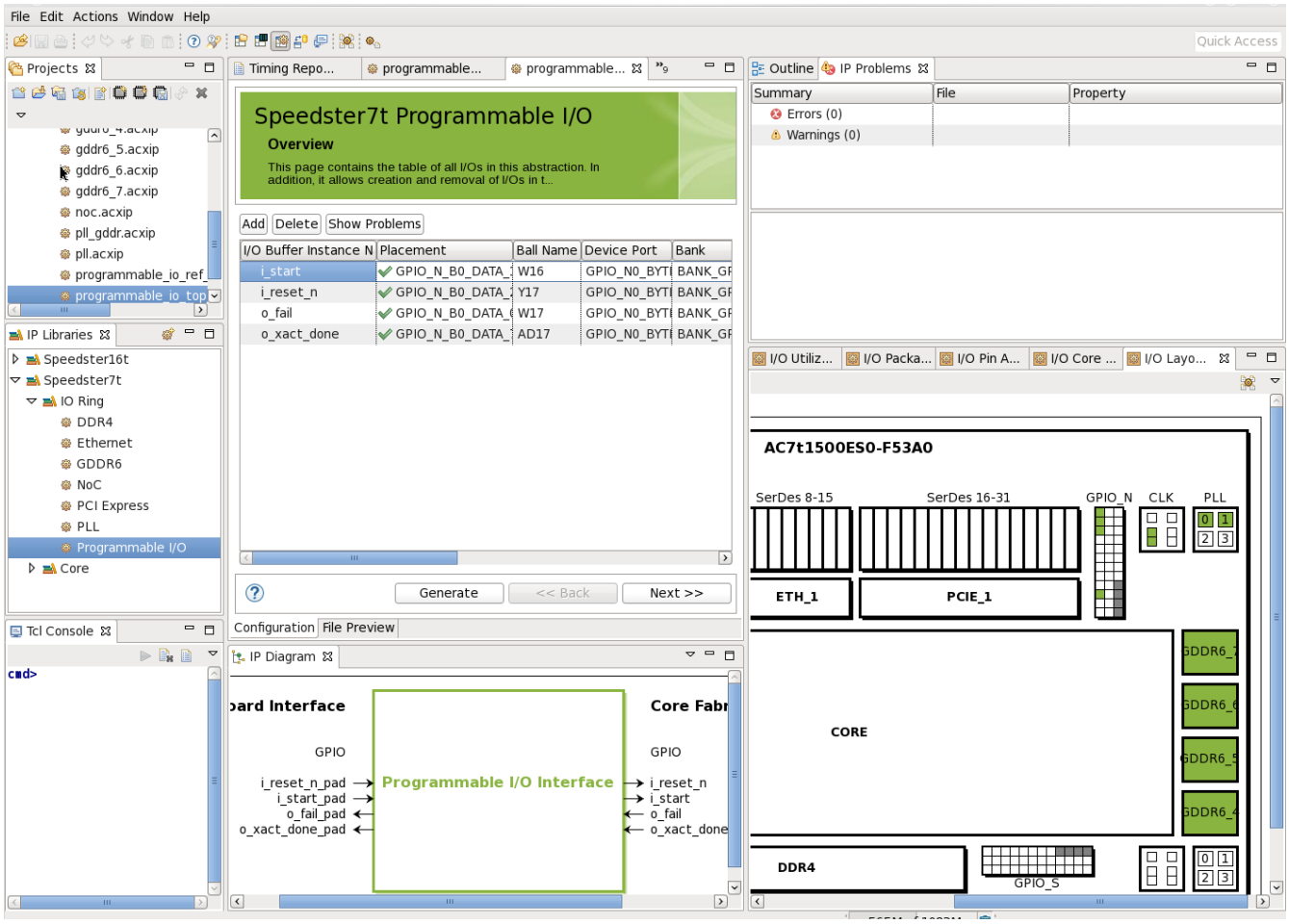

Figure 18: ***Programmable IO IP Configuration in ACE I/O Designer***

Step 3 – Configure the PLL

Next, 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 NoC clock for the NoC interface, and if the GDDR6 subsystem uses the direct-connect (DC) interface, then 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 reference clock, NoC clock and DC interface clock need not be exposed to the fabric core unless these clocks are used by other parts in the core logic. In this case the user needs to enable the 'Expose Clock Output to Core Fabric' option that will let ACE connect this clock input to the fabric core.

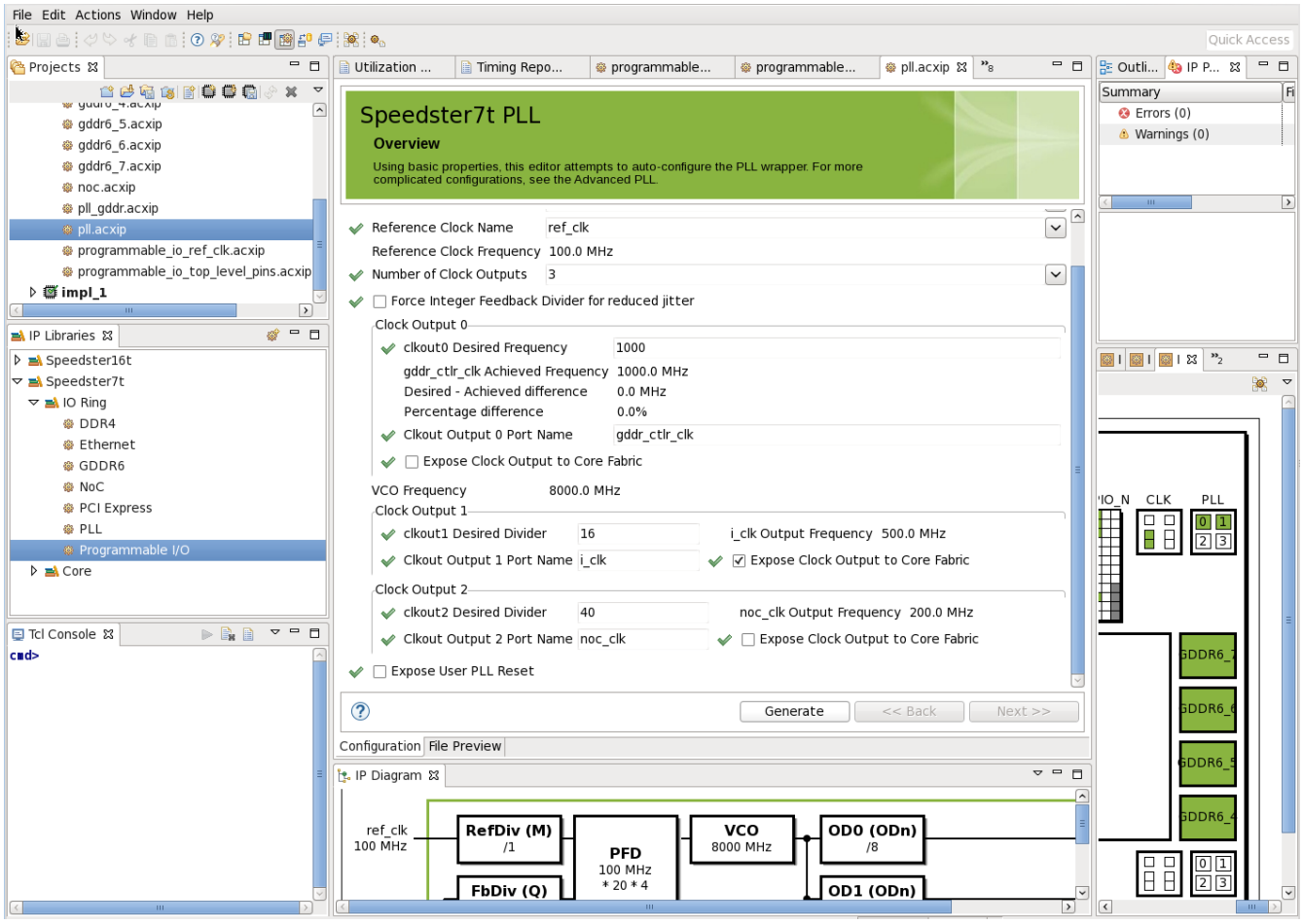

Figure 19: ***PLL IP Configuration in ACE I/O Designer***

Step 4 – Configuring the NoC

If the GDDR subsystem also interfaces with the NoC, then the user also must instantiate NoC IP with the appropriate clock setting. The appropriate NoC reference clock input needed by the PLL must be selected.

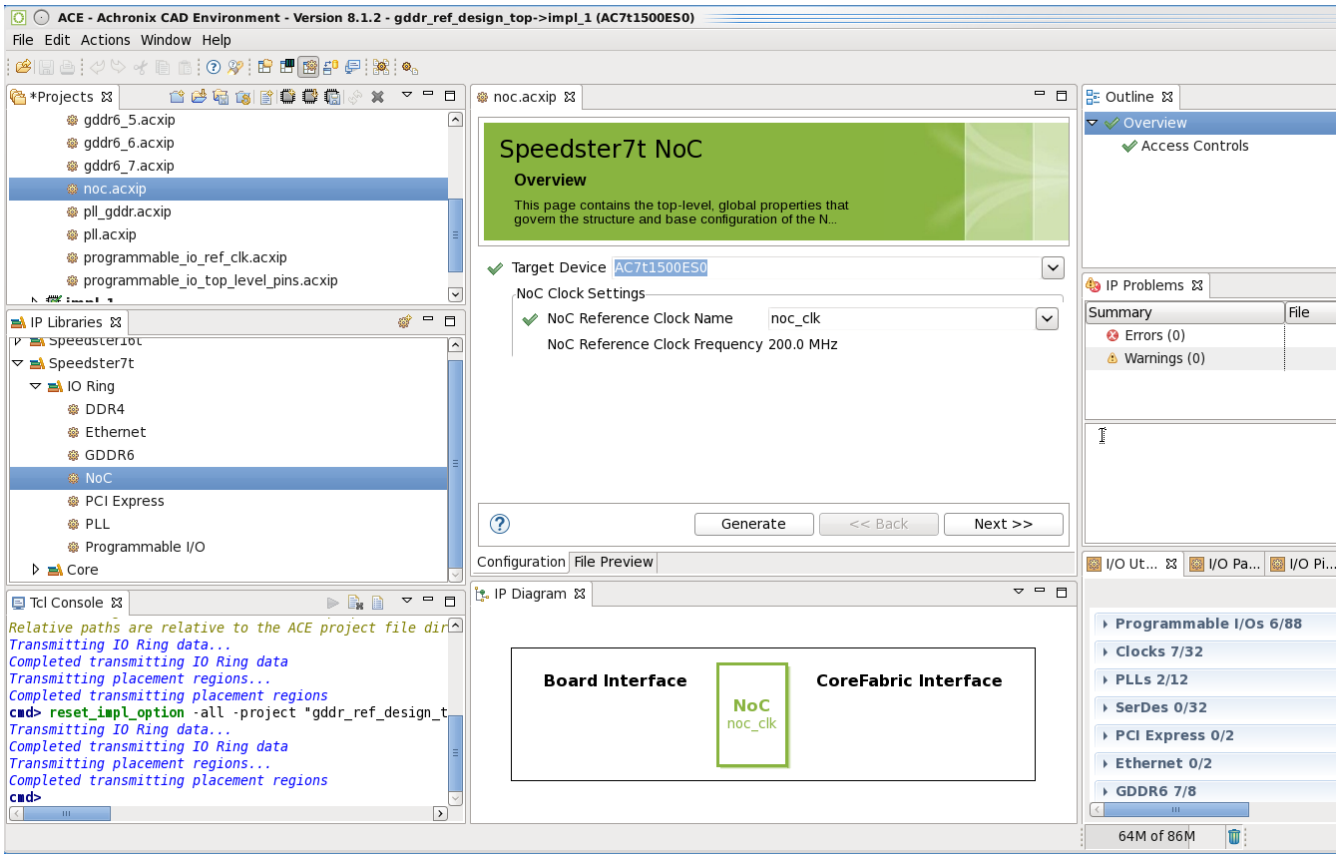

Figure 20: ***NoC IP Configuration in ACE I/O Designer***

Step 5 – Configure the GDDR6 Subsystem

Next, the user must configure each GDDR6 subsystem that will be used in the design. Select the GDDR6 IP in the IP libraries section to create a `.acxip` file. Then, select the desired placement for that particular GDDR6 interface along with the memory part number, data rate and mode of operation. The GDDR6 clock settings will 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 7t1500 FPGA has eight GDDR6 subsystems, with a subset of them 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 the 'Expose Channel 0/1 AXI Interface to Fabric Pins' option. 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 clock and reset signals for each channel and error/interrupt signals. The user design needs to use these clock and reset outputs from the GDDR6 subsystem to drive any GDDR6 DC interface logic in the fabric core.

Note

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

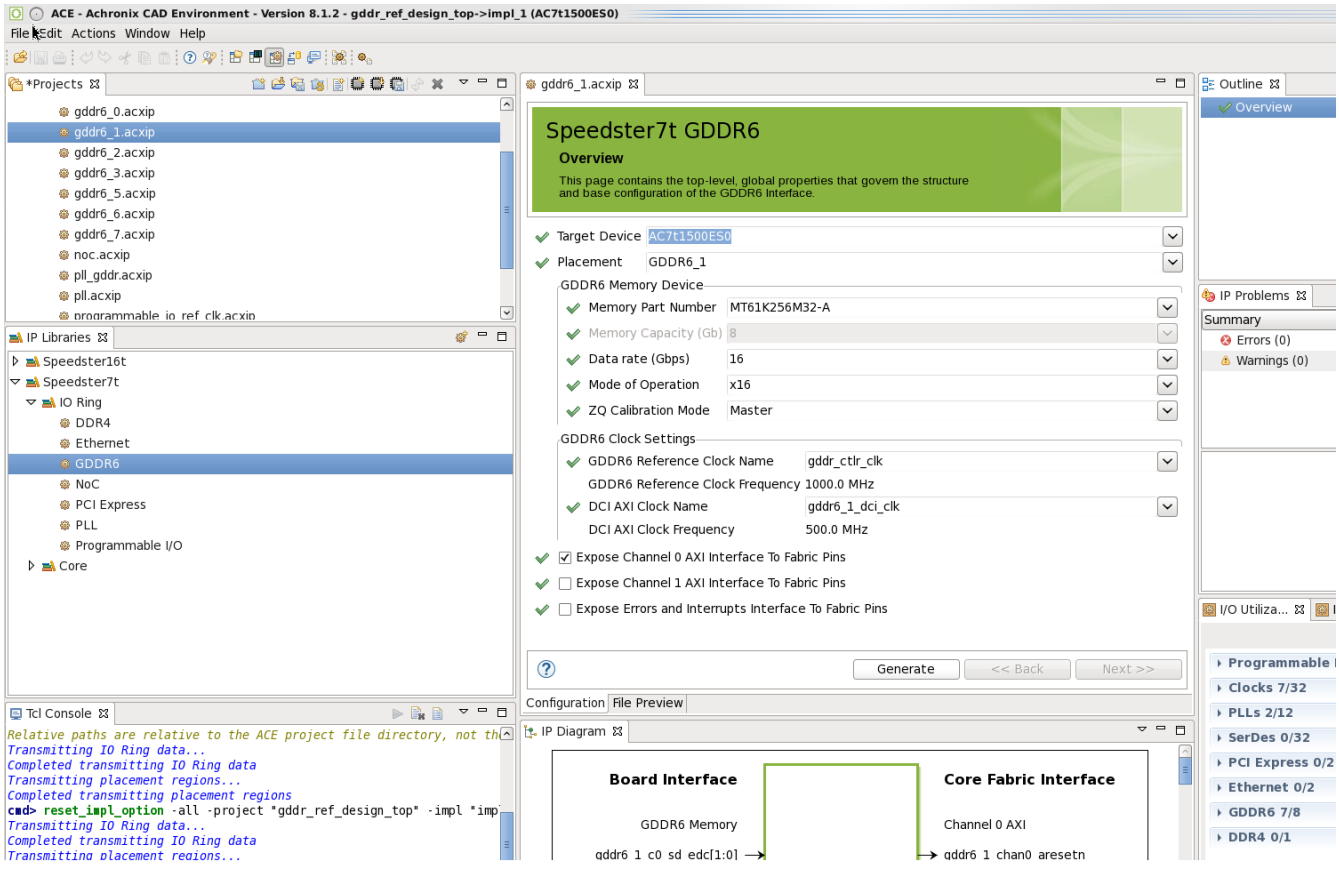

Figure 21: ***GDDR6 Subsystem Configuration in ACE I/O Designer***

Step 6 – Cloning a GDDR6 Instance (Optional)

If the user intends to build a design with multiple GDDR6 subsystems, then an existing GDDR6 subsystem can be cloned by selecting the **Clone IP** option in the right-click menu to match the required instance count. The cloned subsystem instance(s) have to be configured individually later.

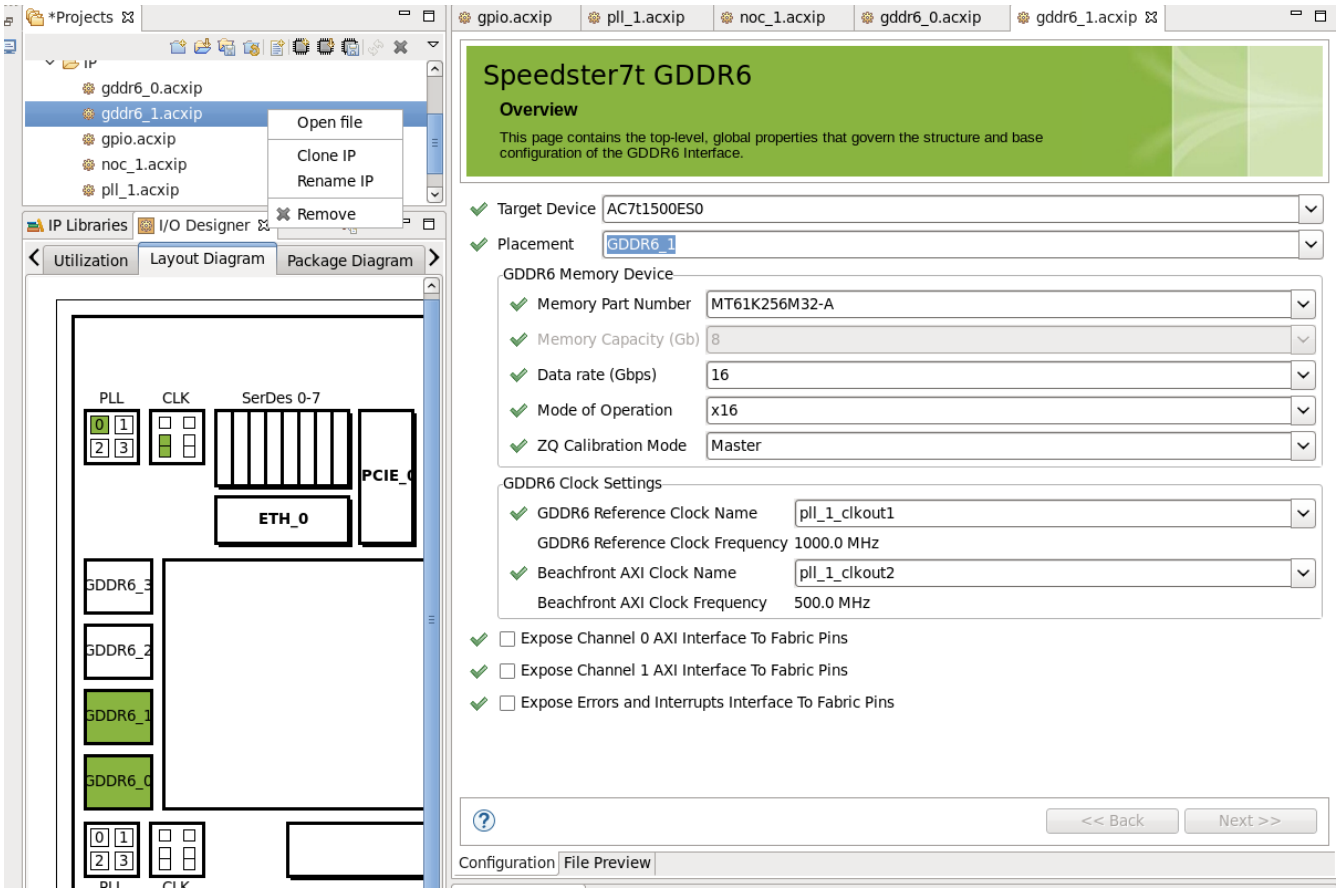

Figure 22: ***Cloning an Existing GDDR6 Subsystem***

Step 7 – Checking for Errors and Generating Files

After all the configuration options are selected, the 'IP Problems' window reports any errors or warnings that occurred with the configuration. If there are no errors or warnings reported, the user can be assured that the entire I/O interface with all the required IP are integrated properly and will close timing at the required clock frequency. Once these checks are done, click **Generate IO Ring Design Files** in the I/O Designer window or the **Generate** option in any of the `.acxip` files to generate all the necessary IP files. The following files are generated as output:

- SDC file with timing constraints for all clocks exposed to the fabric and the GDDR6 channel clock outputs (if the DC interface is enabled for any GDDR6 subsystem)
- PDC file with pin placements for all GPIO pins and GDDR6 DC interface pins if enabled

This step completes the I/O ring configuration. The user can now switch to the core design. This core design will be integrated with the bitstream generated for the I/O interface to obtain the final full-chip integrated bitstream. This output file generation will be enabled in a future ACE release.

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 35 35) and GDDR6 Core and Interface Signals (see page 31), to align with latest tool capabilities. • Added details on the NoC Addressing Scheme for the GDDR6 interfaces to chapter, GDDR6 Interface Connectivity. (see page 23) • Other minor updates and edits.