
Speedster7t DDR User Guide (UG096)

Speedster FPGAs



Copyrights, Trademarks and Disclaimers

Copyright © 2020 Achronix Semiconductor Corporation. All rights reserved. Achronix, Speedcore, Speedster, and ACE are trademarks of Achronix Semiconductor Corporation in the U.S. and/or other countries. All other trademarks are the property of their respective owners. All specifications subject to change without notice.

NOTICE of DISCLAIMER: The information given in this document is believed to be accurate and reliable. However, Achronix Semiconductor Corporation does not give any representations or warranties as to the completeness or accuracy of such information and shall have no liability for the use of the information contained herein. Achronix Semiconductor Corporation reserves the right to make changes to this document and the information contained herein at any time and without notice. All Achronix trademarks, registered trademarks, disclaimers and patents are listed at <http://www.achronix.com/legal>.

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	6
DDR4 Feature Highlights	6
Architecture Overview	7
DDR4 Subsystem Overview	8
Clock and Reset	9
DDR4 Controller	9
DDR4 PHY	10
DDR4 DRAM Interface	10
APB Interface	10
Chapter - 2: DDR4 Controller Architecture	11
DDR4 Controller Features	11
Controller Architecture Overview	13
Port Arbiter (PA) Block	14
Controller Core	14
RAM Interfaces	14
AXI4 Slave Interface	15
APB Interface	15
Chapter - 3: DDR4 PHY Architecture	16
PHY Overview	16
PHY Features	17
PHY Architecture	17
DDRPHYACX4	18
DDRPHYBYTE	18
MASTER PHY	18
PHY Utility Block (PUB)	18
ICCM and DCCM Memories	18
PHY Clocking	18
Chapter - 4: DDR4 Clock and Reset Architecture	19
DDR4 Subsystem Clocks	20
DDR4 Subsystem Resets	20
External Clock and PLL Connection	21

Chapter - 5: DDR4 NoC and Fabric Connectivity	23
Connectivity to the NoC	23
NoC Addressing for DDR4	27
Connectivity to the Direct Connect Interface	27
Direct Connect interface Addressing for DDR4	27
Bypass Interface	27
Chapter - 6: DDR4 Core and Interface Signals	28
Clock and Reset Signals	28
Error and Interrupt Signals	28
DDR4 Subsystem-to-Core AXI4 Interface Signals	30
DDR4 Subsystem to Memory Interface Signals	34
Chapter - 7: DDR4 IP Software Representation in ACE	36
Overview	36
Step 1 - Create a Project	36
Step 2 - Configure the Programmable I/O	37
Step 3 - Configure the PLL IP	38
Step 4 - Configuring the NoC	38
Step 5 - Configure the DDR4 Interface	39
Step 6 - Check for Errors	40
Step 7 - Generate the Design Files	41
Revision History	42

Chapter - 1: Introduction

The Achronix Speedster7t FPGA family provides DDR subsystems that enable the user to fully utilize the low latency and high-bandwidth efficiency of these interfaces for critical applications such as high-performance compute and machine learning systems. The number of DDR subsystems varies within the Speedster7t device family. The DDR subsystem supports memory devices and features compliant with JEDEC Standard JESD79-4B.

The Speedster7t1500 FPGA contains a single high-speed DDR4 interface which can be used to talk to and control off-chip DDR4 memory devices, including a variety of dual in-line memory modules (DIMM) and memory components. The DDR4 controller IP in Speedster7t devices provides a low-power, low-latency and high-performance interface solution for external DDR4 SDRAM devices. The memory controller core clock runs at a maximum of 800 MHz to support data rates up to 3.2 Gbps, which is achieved when the SDRAM clock operates at 1.6 GHz. The controller supports up to 72 bits of data including 8 bits of error correction code (ECC). The DDR4 controller and PHY in the subsystem are implemented as hard IP blocks inside the I/O ring of the Speedster7t1500 FPGA. For resource counts of other Speedster7t family members, refer to the *Speedster7t FPGA Datasheet* (DS015).

DDR4 Feature Highlights

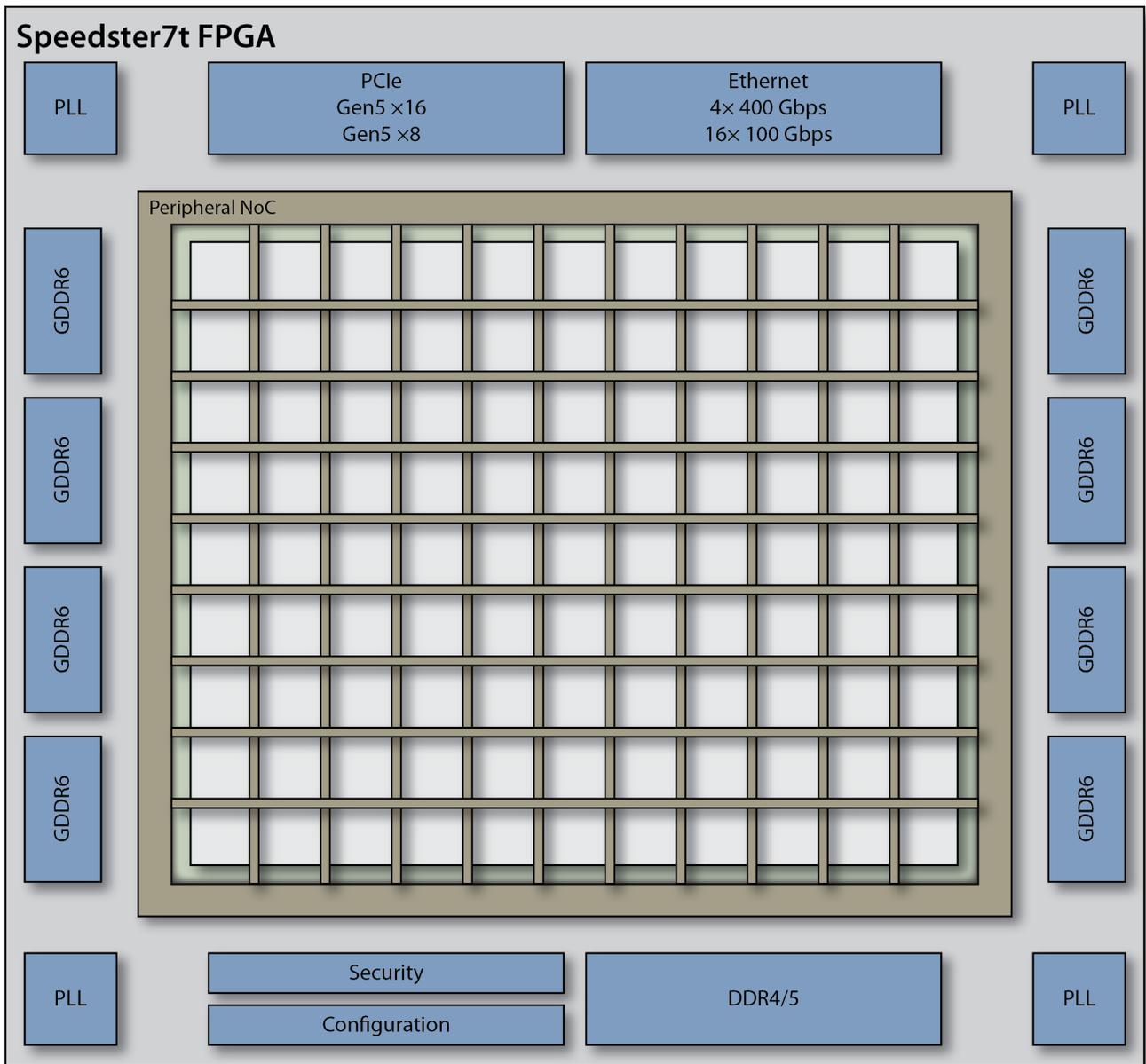
- **Data Rate** – Supports a data transfer rate per pin of up to 3200 Mbps, providing up to 25.6 GBps of memory bandwidth per chip.
- **Interface Width** – Up to 72-bit wide interface; data path widths of 64, 32, 16 and 8 bits are supported. There are 8 check bits to support ECC, 1 bit per DQ byte.
- **Memory Density** – Supports memory densities that are compliant with JEDEC Standard JESD79-4B .
- **Memory Type** – Component mode: SODIMM, UDIMM, RDIMM, LRDIMM, and 3DS-4H DIMM.
- **Multi Rank** – Supports single-, dual- and quad-rank memories.
- **Data Bit Width** – ×4, ×8 and ×16 configuration supported.
- **DQ format** – Double data rate; data latches on the rising and falling edge of the data strobe.
- **Banks** – Four bank groups in the DDR4 SDRAM, providing 16 internal banks that aid faster burst accesses.
- **Burst Modes** – Supports sequential BL8 burst mode and burst chop. Also supports partial reads and writes.
- **DQ Bus** – POD12 configuration that reduces I/O noise and power.
- **Data Mask and Data Bus Inversion** – Supports data mask and data bus inversion.
- **Low Power Modes** – Supports self-refresh and low-power modes.
- **PHY Bypass Mode** – The user has the option to bypass the DDR4 PHY and utilize the associated DDR I/O pins for driving other interfaces. If the user does not require all 72 bits of the data bus, the unused byte lanes can also be bypassed to leverage the use of the designated I/O for other purposes.

Architecture Overview

The figure below shows the architecture of 7t1500 FPGA. The DDR4 memory interface resides on the south edge of the device. There are integrated PLLs in each of the four corners of the device that supply the external reference clocks to all high-speed peripheral interfaces.

The DDR4 subsystem can access the FPGA core logic in the following two ways:

- NoC interface – By accessing the NoC that allows high-speed data to flow between the FPGA fabric and high-speed interfaces. For details, refer to the *Speedster7t Network on Chip User Guide Revision History* (UG089).
- Direct Connect interface – By using the direct fabric connection where the memory controller can be connected directly to the FPGA fabric, similar to existing (or traditional) solutions.



40438623-01.2019.12.18

Figure 1: Speedster7t Architecture Overview of DDR4 Interface

DDR4 Subsystem Overview

A DDR4 subsystem consists of a DDR4 controller, a PHY, a clock and reset block, and an APB interface that updates the control and status (CSR) registers, plus an AXI4 interface to connect to the user logic through either the NoC interface or directly on the FPGA fabric.

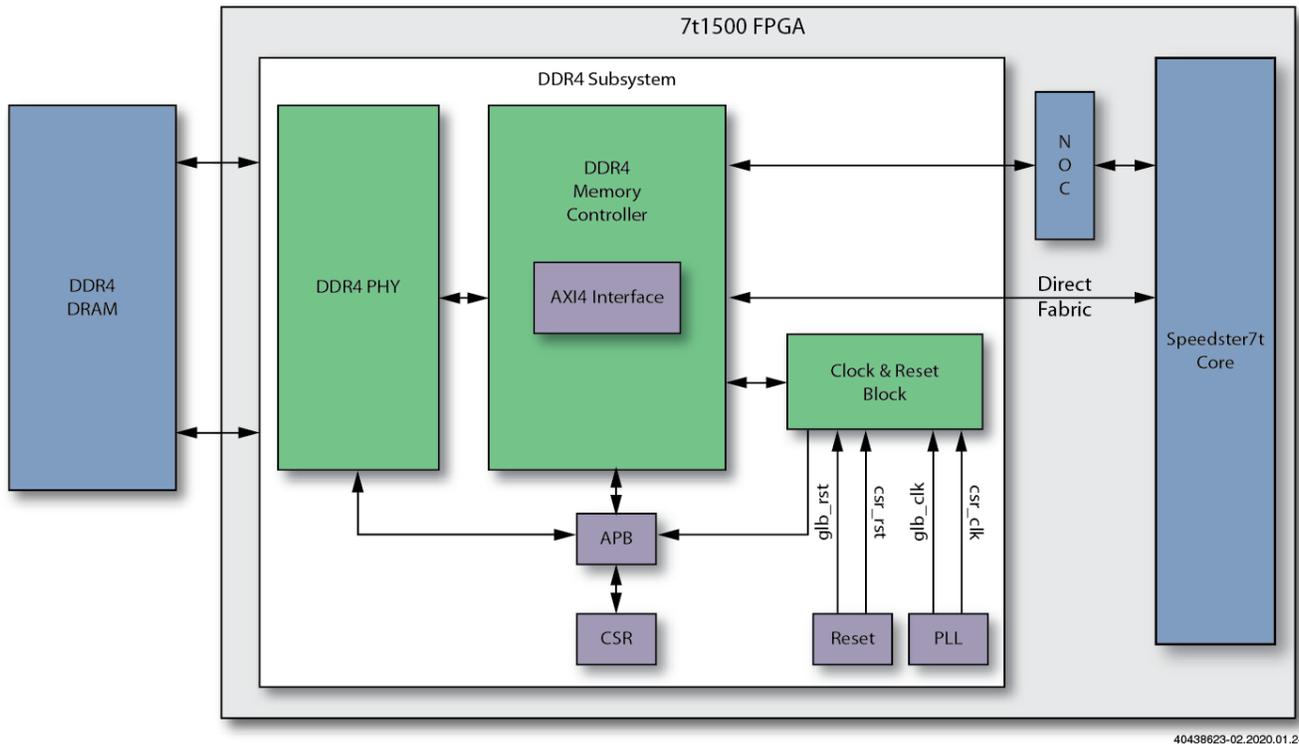


Figure 2: DDR4 Subsystem Block Diagram

The following are key components that are required for a DDR4 memory interface operation:

Clock and Reset

The Speedster7t device has PLLs integrated in the four corners of the FPGA. The PLLs receive external reference clock inputs and generate global clocks used to drive the high-speed interfaces. The DDR4 PHY has an internal PLL that generates the memory clock used at the interface (for DQ/DQS/CA). When the DDR PHY clock runs at 800 MHz, the DDR4 DRAM clock operates at a maximum frequency of 1.6 GHz, generating data transactions at the maximum DDR4 data rate of 3.2 Gbps. The DDR4 memory uses a DDR protocol where data is latched at the rising and the falling edges of the data strobe. The reset circuitry generates global resets and at reset, the controller performs the required initialization of the external DDR4 memory, including calibration and programming of the internal mode registers.

DDR4 Controller

The DDR4 controller has a single clock input, usually driven by the device PLLs and runs at a maximum frequency of 800 MHz. At reset, the controller performs the required initialization and training of the external memory, and programs the internal mode registers of the memory with calibrated settings. Then read and write leveling operations are performed that match the PHY to the byte lane delays of the DIMM module.

- **Memory Read** – To perform a read, a user design signals a read request via the NoC or fabric logic, together with an address and data burst size. The controller responds with an acknowledgement before the data is made available. The controller then translates such a burst of data into multiple consecutive transactions.

- **Memory Write** – To perform a write, a user design signals a write request via the NoC or fabric logic, together with an address and burst size. When the DDR4 memory is ready to receive the data, the controller generates a data request that is sent to the PHY and the transaction is terminated when the data is written to the memory.
- **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 AXI4 256-bit interface that talks to NoC interface and the AXI4 512-bit interface that connects the signals from controller directly to the user logic in the core through the direct-to-fabric interface.

DDR4 PHY

The DDR4 PHY enables the communication between the integrated memory controller and the external DDR4 memory. The PHY supports data width of 64 bits and speeds up to 3.2 Gbps per pin, delivering a maximum bandwidth of up to 25.6 GBps for a single-rank DIMM.

DDR4 DRAM Interface

The DDR4 PHY and the controller IP manages the memory transactions such as precharges, activates and refreshes. The controller issues commands as efficiently as possible, subject to the timing requirements of the DDR4 memory to achieve maximum efficiency

APB Interface

The APB interface operates at 250 MHz and enables the user to configure the DDR4 subsystem registers. The APB interface can be controlled either by the FPGA configuration unit (FCU), or from the fabric when the FPGA is in user mode. During device power-up, initialization, and bitstream loading, the FCU will configure many of the DDR4 subsystem registers. The user design may subsequently reprogram further registers as part of DDR4 calibration and training (with FPGA in user mode).

Chapter - 2: DDR4 Controller Architecture

The DDR4 controller IP in Speedster7t devices contains the logic necessary to accept read and write requests to off-chip DDR4 memory and translates these requests into command sequences. The memory controller ensures proper SDRAM initialization; performs address mapping from system addresses to SDRAM addresses with correct rank, bank group, bank addresses; accepts requests with system addresses and associated data for writes; prioritizes requests to minimize the latency of reads (especially high-priority reads); and maximize page hits. The controller also ensures that refresh functions and other memory and PHY maintenance requests are carried out as required. When the SDRAM enters and exits various power-saving modes, the controller ensures that these operations are executed appropriately.

DDR4 Controller Features

Here is a list of features that the DDR4 controller in Speedster7t devices support:

Table 1: DDR4 Controller Features

Feature	Description
Supported maximum data rate	Memory controller supports DDR4 operation at up to 3.2 Gbps per channel
Controller clock rate	Runs at a clock rate of 800 MHz to support maximum data rate 3.2 Gbps
Memory protocol	DDR4; 3DS(DDR4)
Memory type	Component mode: UDIMM, RDIMM, LRDIMM, SODIMM
Memory configurations	Supports ×4, ×8 and ×16 bit-width configurations
Number of channels	Supports single-channel mode with 64 data bits
Burst modes	<ul style="list-style-type: none"> • BL8 • Burst chop • Sequential burst mode • Partial reads and writes
Data bus width mode	Supports full, half and quarter bus width modes
Multi ranks	<ul style="list-style-type: none"> • Supports up to 4 physical ranks • Supports up to 16 logical ranks (for 3DS 4H only)

Feature	Description
Power saving	<ul style="list-style-type: none"> • Clock gating • Power down • Idle-low-power-down controller switches off clock; related to maximum power saving mode • Self refresh • Dynamic tri-stating
Refresh control	<ul style="list-style-type: none"> • Supports per bank and all bank refresh • Automatic refresh • Software Initiated refresh - APB writes to registers to initiate refresh
Data mask (DM) and data bus inversion (DBI)	Supports data masking and data bus inversion
Page policy	<ul style="list-style-type: none"> • Per command auto-precharge control • Open-page
Memory error correcting code	Supports sideband ECC
Command address parity	Supports command address parity
Transaction service control	<ul style="list-style-type: none"> • Transaction store • Bus turn-around control • Multi rank control
QOS and efficiency	<ul style="list-style-type: none"> • AXI/DDR controller – variable priority read/write (VPR/VPW) and associated timeout • AXI – traffic class mapping from following QoS signals: <ul style="list-style-type: none"> • HPR – high priority read • LPR – low priority read • VPR – variable priority read • NPW – normal priority write • VPW – low priority write • AXI – dual read address queue • AXI – urgent signals (sideband) • Port throttling • Transaction re-ordering for improving bus efficiency.

Feature	Description
ZQ control	<ul style="list-style-type: none"> • ZQ long calibration supported after self-refresh exit (through register writes) • Automatic ZQ command(periodical) enable/disable • Automatic ZQ command enable/disable after SRX,MPSMX • ZQ resistor shared • SW initiated ZQ command
Programmable DRAM bus width (new HBW/QBW in XPI)	<ul style="list-style-type: none"> • Programmable DRAM bus width (new HBW/QBW in XPI) • Enhanced read/write switching
DDR4-specific features	<ul style="list-style-type: none"> • MPR writes/read • Fine granularity refresh (FGR) • On-die termination • CA parity (except retry feature) • Maximum power saving mode (MPSM) • Power-down auto-precharge (PDA) • Programmable preamble (2tCK) • Gear-down mode to facilitate signal integrity

Controller Architecture Overview

The figure below gives an overview of the DDR4 memory controller IP and shows its sub blocks, the controller core, port arbiter, AXI port interface, SRAM memories to implement the read reorder buffer (RRB), write-data RAM and the APB Interface.

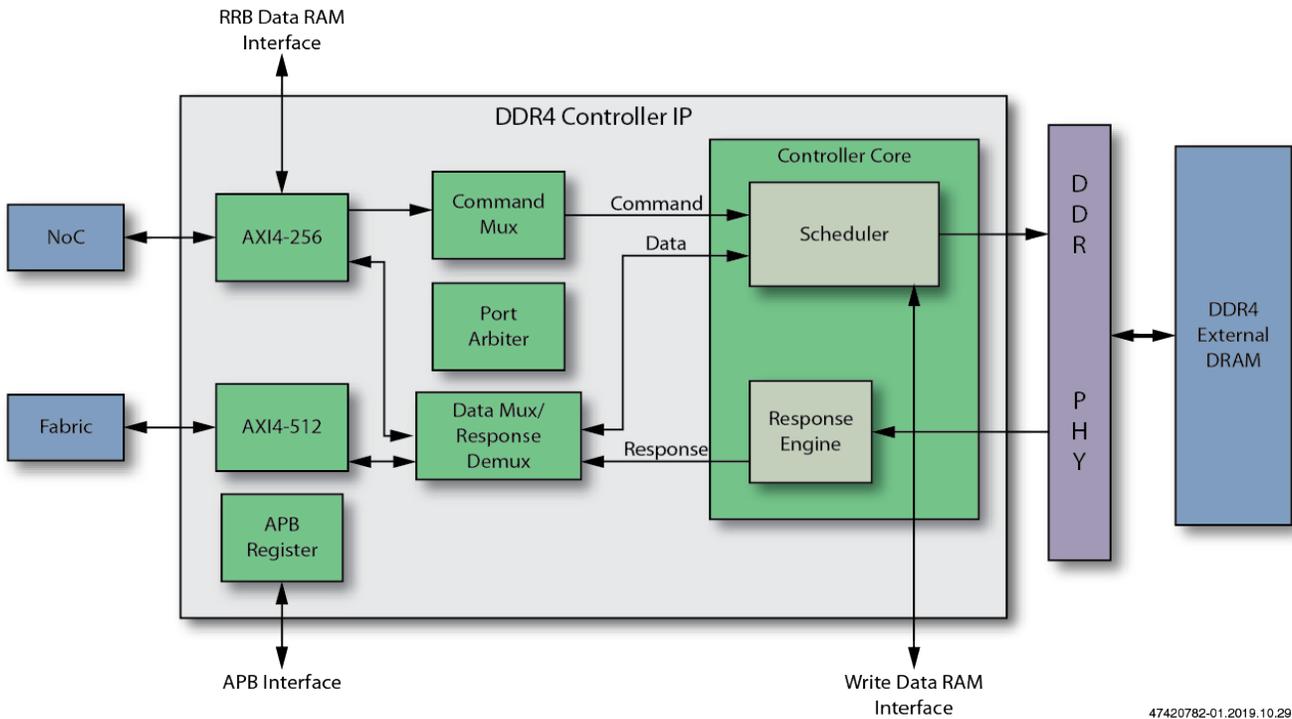


Figure 3: DDR4 Memory Controller Block Diagram

The DDR4 memory controller IP contains the following main architectural blocks.

Port Arbiter (PA) Block

This block provides latency-sensitive, priority-based arbitration between the addresses issued by the AXI ports for command requests from AXI ports to the DDR controller.

Controller Core

This block contains a logical content-addressable memory (CAM), that holds information on the commands, that is used by the scheduling algorithms to optimally schedule commands to be sent to the PHY, based on priority, bank/rank status and DDR timing constraints. The write data is stored in an embedded SRAM internal and external to the controller until its associated command is issued to the DDR4 PHY. The read data is handled by the response engine in the controller core and is returned in the order of scheduled read commands on the host interface. ECC handling is an optional function, which is handled by logic modules within the controller core in the write data path and in the response engine.

RAM Interfaces

The memory controller implements a read reorder buffer so that the read data can be returned from the controller core in a different order from which the read commands are forwarded from the AXI interface. This reordering of read commands in the controller helps maximize DRAM bandwidth. There is also a write-data RAM for storing data until the time the data can be made available to the controller core.

AXI4 Slave Interface

The AXI transactions are input to the controller through the NoC interface via the 256-bit AXI4 interface or directly to an adjacent fabric cluster using the 512-bit AXI4 interface. The AXI clocks for the NoC and direct connections (to the fabric) are asynchronous to each other. The resets for these AXI interfaces, are synchronized for de-assertion with their respective AXI clocks before they are input to the memory controller.

This burst-based interface enables read and write request channels that specify the host ID for the request, start byte address, burst length, burst size, and burst type. This information is processed by the interface and is used subsequently by the DDR controller core. The AXI ports interface with the controller core to perform read/write address and data/response generation. Read data is stored in a buffer and returned in order to the AXI ports.

APB Interface

The APB interface is used to configure the clocks and resets in the DDR4 subsystem. The APB interface also makes the internal registers of both the controller and the PHY accessible to the user logic. The APB interface to the DDR4 subsystem consists of a 28-bit address and a 32-bit data bus. The subsystem control and status registers address (CSR) and memory controller address are used as-is from the APB interface. The available configuration space through the APB bus to which the memory controller is mapped is in the address range 0x000_0000 to 0xFFF_FFFF.

The table below describes the configuration space register address allocation for the controller, PHY and the DDR subsystem.

note



A detailed description of these register bits will be provided in a future release of this guide.

Table 2: Control and Status Register Address Map

Address Range	Accessible Space
0x000_0000 to 0x0FF_FFFF	Memory controller register space
0x100_0000 to 0x1FF_FFFF	PHY register space and PHY training FW memory access
0x200_0000 to 0x2FF_FFFF	Subsystem control and status register space

Chapter - 3: DDR4 PHY Architecture

PHY Overview

The embedded Speedster7t DDR4 PHY supports the DDR4 memory standard at the channel interface and the DFI-4.0 interface on the FPGA side with the memory controller. It supports a maximum data rate of 3.2 Gbps and is targeted for systems that require low-latency and high-bandwidth memory solutions.

The figure below shows the PHY interfacing with the off-chip DDR4 memory on one side and the embedded DDR4 memory controller on the FPGA side.

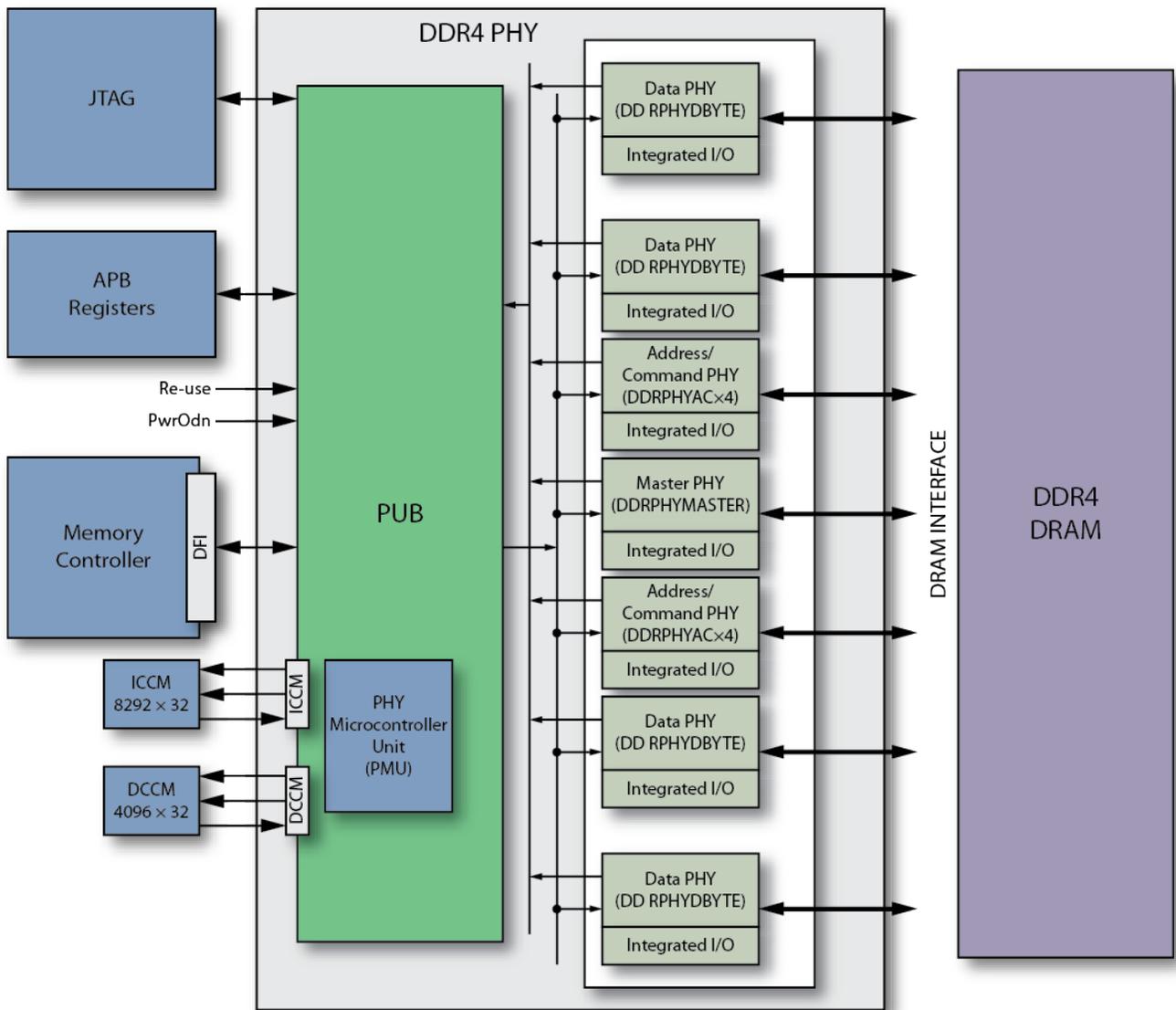


Figure 4: DDR4 PHY Block Diagram

PHY Features

Feature	Description
PHY rate	The PHY operating frequency is at a max of 800 MHz.
DRAM maximum data rate	High-performance DDR4 PHY supporting data rates up to 3.2 Gbps per data pin
Memory protocol	DDR4, 3DS(DDR4)
Memory type	Component mode: UDIMM, RDIMM, LRDIMM, SODIMM
Memory configurations	Supports ×4, ×8 and ×16 bit width configurations
Number of channels	Supports single-channel mode with 64 data bits
Multi-rank support	<ul style="list-style-type: none"> • Supports up to 4 physical ranks • Supports up to 16 logical ranks (for 3DS 4H only)
PHY training	PHY independent, firmware-based training using an embedded calibration processor
Embedded DLLs	Voltage and temperature compensated delay lines used for DQS centering; DDR4 read/write leveling and per-bit deskew
Embedded PLL	Includes a low-jitter PLL for both PHY clock generation and SDRAM clock generation
Other DDR4 features	<ul style="list-style-type: none"> • DRAM addressing up to 16Gb • Supports data bus inversion (DBI) • Command address parity • Direct programming of DQ VREF level setting; automatic training of DQ VREF level setting separately for PHY per nibble and SDRAM per device • MPR writes/reads • Per DRAM addressability • Programmable input on-die termination (ODT) and output impedance • Programmable preamble (2tCK) • Gear-down mode for non-registered systems

PHY Architecture

The embedded DDR4 PHY IP in Speedster7t FPGAs consists of multiple components shown in the block diagram above. The sections that follow describe each of these components,

DDRPHYACX4

The DDRPHYACX4 is the PHY component for the address/command lane. Features of this component include:

- High-speed digital logic pipeline for transmitting address/command to the SDRAM.
- I/O component that includes PVT-compensated output impedance.

DDRPHYDBYTE

The DDRPHYDBYTE is the data receive/transmit building component for the DDR4 PHY, statically configurable as one byte-wide data lane. Features of the DDRPHYDBYTE include:

- High-speed digital logic pipeline for transmit and receive datapaths to and from the SDRAM.
- I/O component that includes PVT-compensated, on-die termination (ODT) and output impedance.

MASTER PHY

The master PHY is the location of the high-speed PLL used to generate the high-speed clock for data transmit and high-speed digital pipelines. The PLL has the following modes of operation:

- The PLL block generates the actual DRAM bit-rate clock.
- Power on reset (POR) circuit, I/O macros for DRAM reset, DRAM alert_n signals.

PHY Utility Block (PUB)

The PHY utility block provides configuration, control, and other utility functions for all PHY hard components. This block handles a variety of functions for the PHY, including driving initialization sequences, providing all training functions, implementing the DFI protocol, implementing registers and a register access bus.

ICCM and DCCM Memories

The microcontroller in the utility block, which runs the training firmware, requires two memories: DCCM and ICCM memories. These firmware memories are accessible on APB interface as a contiguous CSR space. The ICCM is mapped on PHY CSR address space to the address range 0x0005_0000 to 0x0005_7FFF. The DCCM is mapped on PHY CSR address space to the address range 0x0005_4000 to 0x0005_3FFF. These memories are loaded during bitstream configuration via APB bus and are both word addressable.

PHY Clocking

One of the global clocks generated by the device PLLs is selected as the core clock that drives both the memory controller core and the PHY. The PHY clock operates at a maximum of 800 MHz and is required to be synchronous to the controller core clock. This PHY clock generates the external DDR4 SDRAM memory clock at 1.6 GHz that helps achieve the data rate of 3.2 Gbps on the memory interface.

Chapter - 4: DDR4 Clock and Reset Architecture

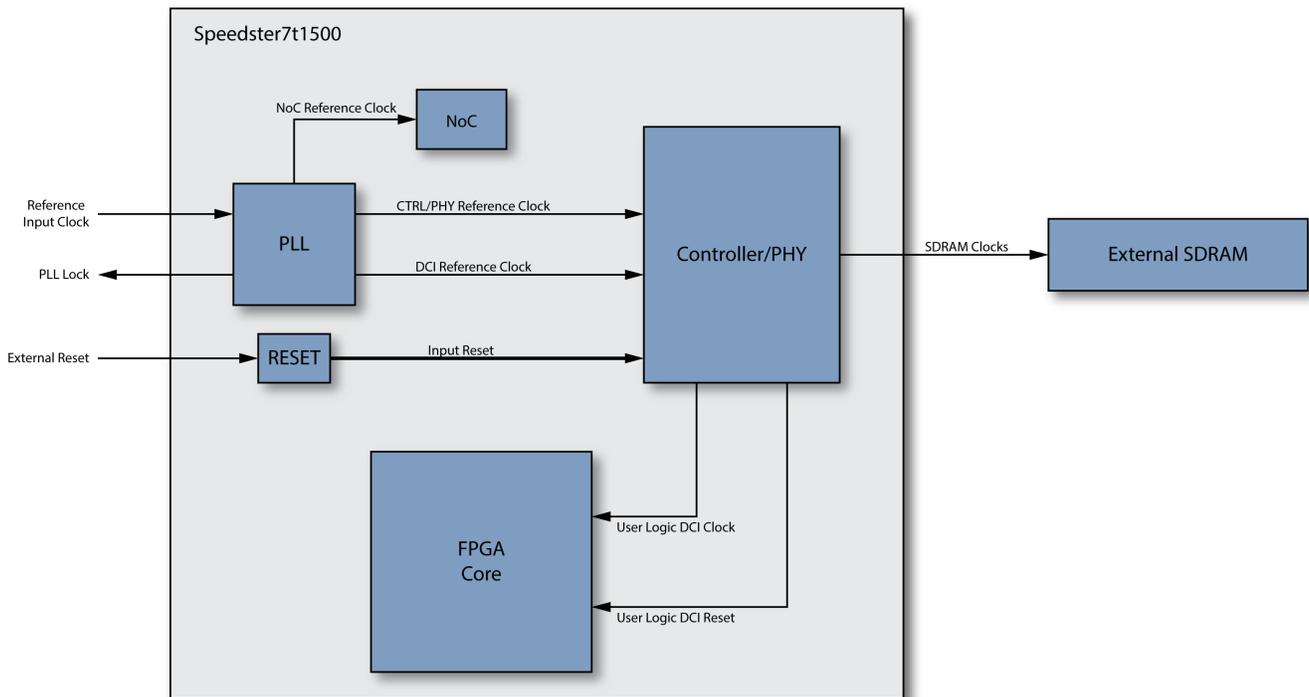
A Speedster7t device requires external input reference clock and reset signals to drive the DDR4 memory interface. The Speedster7t device clock and reset generator module consists of PLLs, DLLs and reset circuitry to generate these signals. There is a clock and reset generator placed in each corner of the 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. Refer to the *Speedster7t Clock and Reset Architecture User Guide (UG083)* for further details.

The DDR4 subsystem requires three input clocks selected from the global clock network. These clocks can be sourced from either a single or multiple PLLs. The DDR4 subsystem requires the following clocks:

- DDR memory controller and PHY reference clock
- Input reference clock for the 256-bit AXI4 NoC interface
- Input reference 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 resets. Any of these resets can be selected to provide the reset input to the DDR subsystem.

The diagram below depicts the flow of all the clocks and resets required for the DDR4 subsystem.



55345240-01.2020.03.16

Figure 5: Clock and Reset Architecture of DDR4 Subsystem

DDR4 Subsystem Clocks

The DDR4 memory controller clock is selected from one of the global clocks generated by the PLL and runs at a maximum of 800 MHz to support maximum data rate of 3200 Mbps (achieved when DDR4 SDRAM memory clock runs at 1600 MHz). The DDR4 PHY clock is also selected from the same clock source as the controller clock. The PHY's internal PLL generates the external SDRAM memory clock. When the PHY runs at 800 MHz, the DDR4 memory operates at the maximum supported data rate.

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 NoC reference input clock always operates at 200 Mhz and is selected from the global clock outputs. The clock driving the DDR user logic for the NoC interface is handled internally in the NoC and can operate at a max rate of 800 Mhz. The direct-to-fabric connection (DC) AXI clock, also chosen from a global clock can run at a maximum frequency of 400 MHz and drives the DDR user logic for the 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.

Both the AXI interface clocks are routed through the DDR subsystem and then fed to the NoC or the DC interface for driving user logic to avoid clock divergence issues. All clock domain crossings are also handled internally within the DDR4 subsystem.

The table below shows the maximum operating frequencies and the relationships between the different clock domains in the DDR4 subsystem. The user can scale the clocks maintaining the ratios and achieve desired rates of operation.

Table 3: DDR4 Subsystem Clocks

DDR4 Data Rate	AXI-256 NoC Clock	AXI-512 Direct-to-Fabric Connection Clock	Controller Clock ⁽¹⁾	PHY Clock ⁽¹⁾	SDRAM Clock ⁽²⁾
3200 Mbps	800 MHz	400 MHz	800 MHz	800 MHz	1600 MHz
2666 Mbps	800 MHz ⁽³⁾	400 MHz ⁽³⁾	667 MHz	667 MHz	1333 MHz
2400 Mbps	800 MHz ⁽³⁾	400 MHz ⁽³⁾	600 MHz	600 MHz	1200 MHz

Table Notes



1. The controller and PHY clock always operate at half rate of the SDRAM clock.
2. The external SDRAM clock always operates at DATA_RATE/2 of the DDR4 memory. For example, if the DDR4 memory is operating at 3200 Mbps, memory clock frequency will be 1600 MHz.
3. These are suggested rates, they can be scaled down as long as it meets desired interface data rates.

DDR4 Subsystem Resets

The DDR4 subsystem has access to 80 available resets: 32 global resets and 48 active-low startup resets. The controller reset is asynchronous to the controller clock and is synchronized for de-assertion in the subsystem before it is sent to the memory controller. All resets can also be configured through IPCNTL reset selection registers.

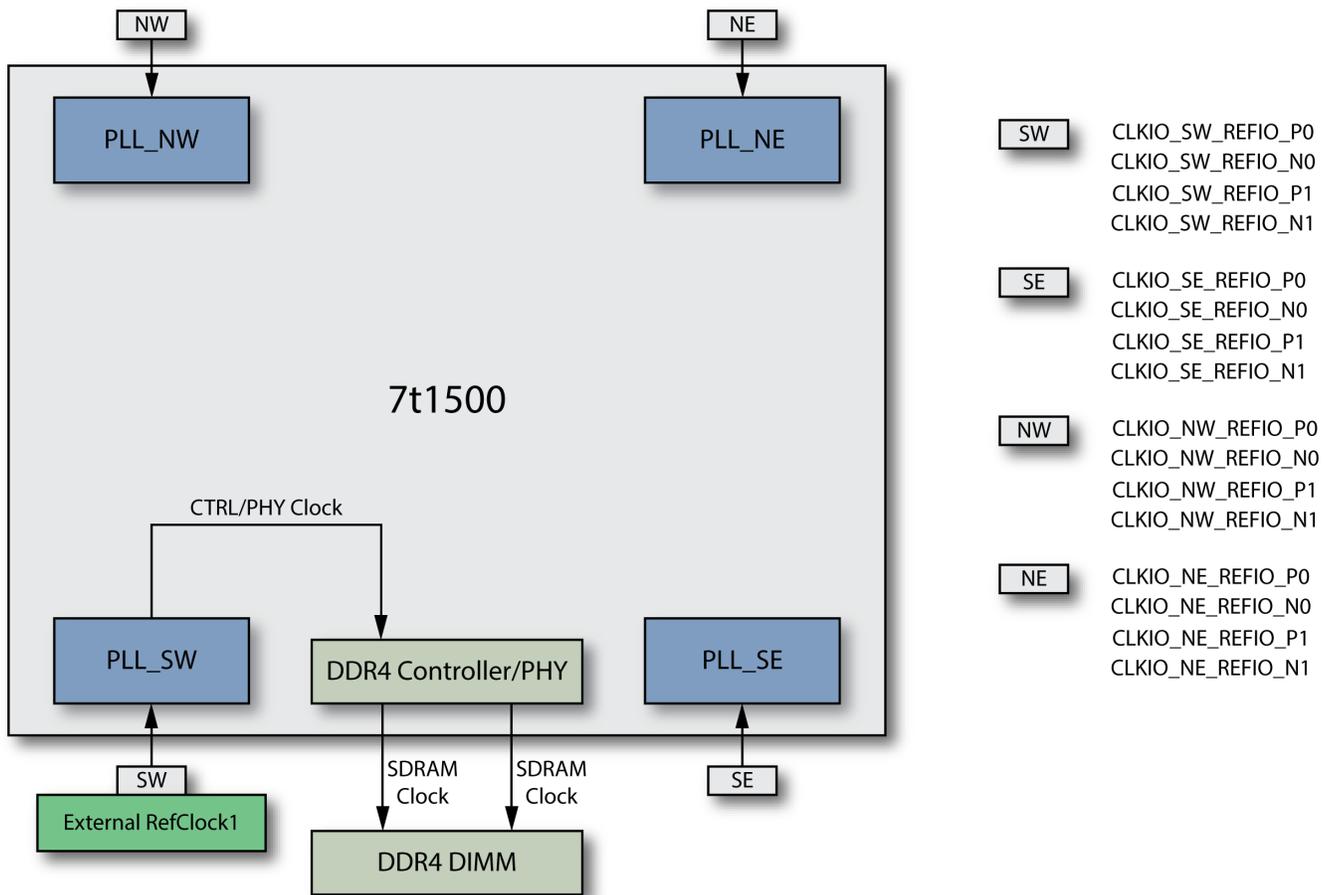
Note

The details of these register settings to be programmed using the APB interface will be covered in future releases of the user guide.

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-to-fabric connect is utilized, the DDR4 subsystem outputs the reset to the FPGA fabric (DCI Reset as shown in the block diagram).

External Clock and PLL Connection

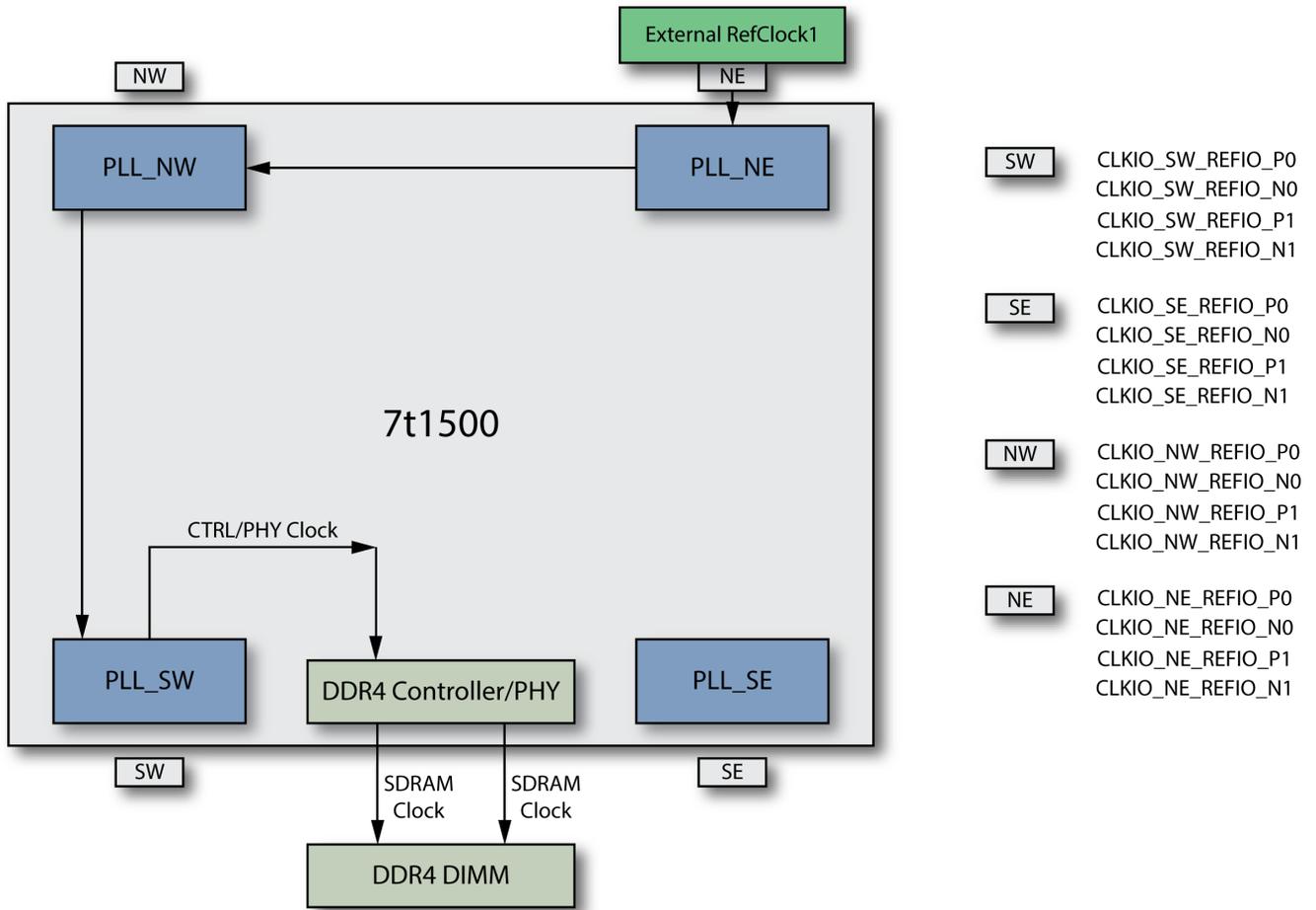
All the corner PLLs can be utilized to generate clocks for the DDR4 subsystem to perform at maximum data rates. The following figures illustrate how a user can connect the external reference input clock to drive the DDR4 subsystem. In the figure below, the south-west PLL generates clocks that drive the DDR4 subsystem clocks and the DDR4 interface SDRAM clock (ck_p and ck_n).



55345240-02.2019.12.26

Figure 6: South PLLs Driving the DDR4 Subsystem

The figure below shows the north-east PLL generates clocks that drive the north-west PLL which subsequently can be used to generate DDR4 clocks:



55345240-03.2019.11.13

Figure 7: North PLLs Driving the DDR4 Subsystem

Chapter - 5: DDR4 NoC and Fabric Connectivity

The following sections describe the two interfaces supported by the DDR4 subsystem to connect to the user logic, namely the network on chip (NoC) and the direct connect (DC) interfaces. The DDR4 subsystem can also be configured to utilize both interfaces at the same time. If a transaction arrives on the NoC, the DDR subsystem will respond to it via the NoC interface. Similarly, if a transaction arrives on the DCI, the DDR subsystem will respond to it via the DCI. This capability is demonstrated in the DDR4 user reference design. Both NoC and DC interfaces use a standard AXI4 protocol; details of the specification can be found at [AXI Protocol Specification](#).

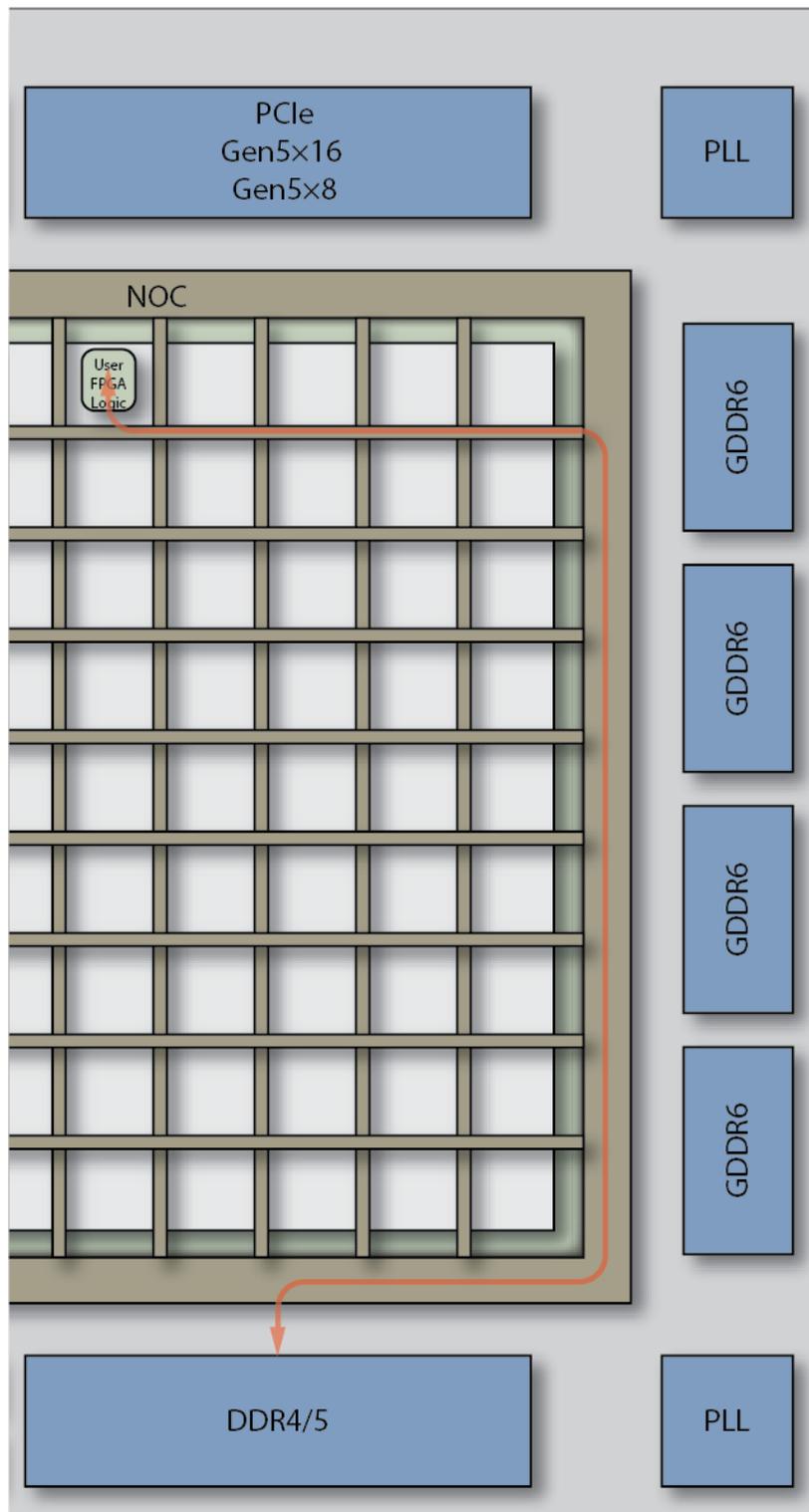
Connectivity to the NoC

The Speedster7t FPGA architecture has a network on chip that enables extremely high-speed data flow between the FPGA's core and its interfaces, as well as between logic within the FPGA fabric itself. The DDR4 subsystem can receive transactions initiated by the FPGA fabric or by PCIe via the NoC. For more details on the NoC, refer to the *Speedster7t Network on Chip User Guide* (UG089)

The NoC interface connectivity to the DDR4 subsystem is the default connection in ACE I/O Designer toolkit :

- This interface exists as part of the controller interface and the user does not need to create it.
- The user will need to only establish a connection from master logic in the fabric to the NoC by instantiating the ACX_NAP_AXI_SLAVE macro in the user design (refer to the section, NAP_AXI_SLAVE, in the *Speedster7t IP Component Library User Guide* (UG086)). Once a connection is established the master logic can send AXI transactions through the NoC interface to the DDR memory.
- For transactions from the PCIe subsystem via the NoC interface, the user needs to configure the PCIe subsystem, the NoC and the DDR4 subsystem in the I/O Designer Toolkit and requires the PCIe subsystem to send AXI commands to the DDR4 memory addresses.
- The interface runs at a maximum rate of 800 MHz, supporting a DDR4 data rate of 3.2 Gbps.

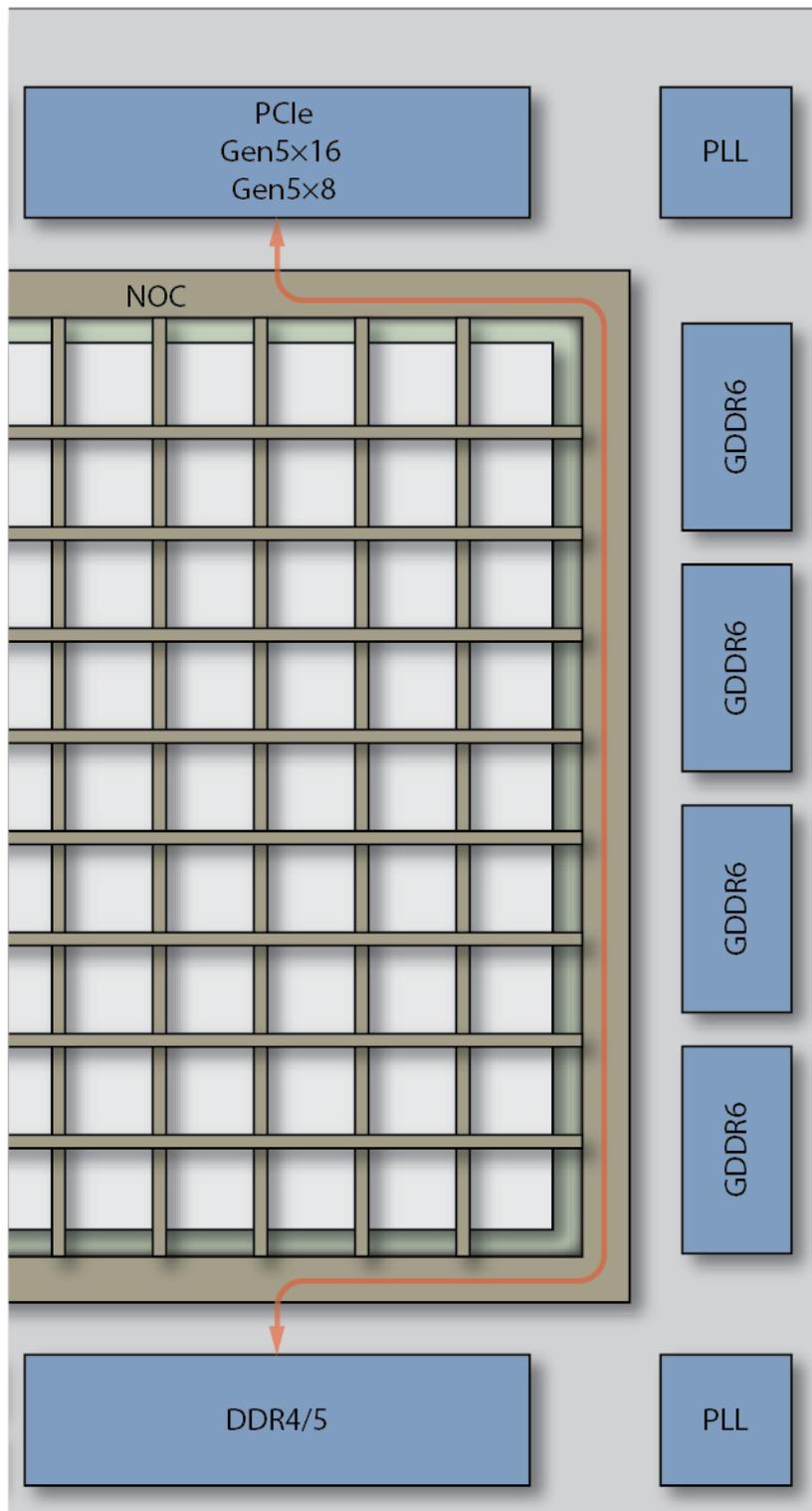
The following figure shows how the master logic in the FPGA fabric sends and receives transactions from the DDR4 interface utilizing the NoC.



47420789-03.2020.01.21

Figure 8: Data Flow from Core Logic in FPGA Fabric to DDR4 Through the NoC

The figure below shows the PCI Express master issuing a transaction to the NoC, which is transmitted directly to the DDR4 interface without involving resources within the FPGA fabric.



47420789-02.2019.12.18

Figure 9: Data Flow from PCIe Interface to DDR4 Through the NoC

NoC Addressing for DDR4

The table below shows how DDR4 NoC addressing is established.

Table 4: DDR4 NoC Addressing Scheme

Address Bit	41	40	39	38	37	36	35	34	33	32	31	30	29	28	..	0
DDR4	0	1	Memory Address													

The target ID for DDR4 is represented by two most significant bits, and the remaining bits, Addr[39:0], represent the DDR4 external memory address.

The address depends on the device density. For example for a 128-Gb device (16 GB) requires a 34-bit address, Addr[33:0], on the AXI ports because AXI uses a byte addressing scheme. The remaining bits, Addr[39:34], should be set to zero.

Connectivity to the Direct Connect Interface

The DDR4 subsystem also enables a connection directly to the fabric through the direct connect interface, a 512-bit AXI slave interface which connects the DDR memory controller to master logic in the FPGA. This AXI interface:

- Exposes the clocks and resets which are outputs from the DDR subsystem and are connected to the fabric cluster
- Runs at 400 MHz, supporting a DDR data rate of 3.2 Gbps

Direct Connect interface Addressing for DDR4

The address bus width for DCI is set to 40 bits and is represented exactly the same way as for the NoC Addr[39:0] bits.

Bypass Interface

The DDR4 PHY has the provision of being bypassed, allowing the FPGA core to control its input and output pins directly as GPIO. PHY bypass is achievable by turning on the bypass mode enable control bits. There are a total of 156 DDR4 signals that can be utilized as GPIO in addition to 64 GPIO on the 7t1500 device.

Chapter - 6: DDR4 Core and Interface Signals

The Speedster7t1500 FPGA has a single DDR4 hard IP core that can be enabled and configured by the user in ACE. The DDR4 subsystem for direct connect (DC) interface connection comprises:

- [Clock and Reset Signals \(see page 28\)](#)
- [Error and Interrupt Signals \(see page 28\)](#)
- [DDR4 Subsystem-to-Core AXI4 Interface Signals \(see page 30\)](#)
- [DDR4 Subsystem to Memory Interface Signals \(see page 34\)](#)

This section below lists all the signals that a user requires to bring up in a design that interfaces with the DDR4 subsystem on the Speedster7t FPGA.

Note



The DDR4 ports have a logical and consistent naming scheme within ACE, consisting of a <prefix>_function. The <prefix> is user defined.

Clock and Reset Signals

The following table summarizes the clock and reset signals from DDR4 subsystem when DCI is enabled.

Table 5: DDR4 Subsystem Clock and Reset Signals

Pin Name	Direction	Width	Description
<prefix>_clk	Output	1	Clock from DDR4 subsystem to the FPGA core at 400 Mhz max rate
<prefix>_rstn	Output	1	Reset from DDR4 subsystem to the FPGA core

Error and Interrupt Signals

The following table lists the error and interrupt signals in the DDR4 subsystem.

Table 6: DDR4 Subsystem Error and Interrupt Signals

Pin Name	Direction	Width	Description
<prefix>_ecc_corrected_err_irq	Output	1	ECC corrected error interrupt. This interrupt is asserted when a correctable ECC error is detected at the DFI
<prefix>_ecc_corrected_err_irq_fault	Output	1	ECC corrected error fault. This signal is a version of ecc_corrected_err_intr. ^(†)

Pin Name	Direction	Width	Description
<prefix> _ecc_uncorrected_err_irq	Output	1	ECC uncorrected error interrupt. This interrupt is asserted when a uncorrectable ECC error is detected
<prefix> _ecc_uncorrected_err_irq_fault	Output	1	ECC uncorrected error fault. This signal is a version of ecc_uncorrected_err_intr. ^(†)
<prefix>_dfi_alert_err_irq	Output	1	DFI alert interrupt from the DDR4 PHY
<prefix>_par_waddr_err_irq	Output	1	AXI write address parity error interrupt
<prefix> _par_waddr_err_irq_fault	Output	1	AXI write address parity error fault. This signal is a version of par_waddr_err_intr. ^(†)
<prefix>_par_raddr_err_irq	Output	1	AXI read address parity error interrupt
<prefix> _par_raddr_err_irq_fault	Output	1	AXI read address parity error fault. This signal is a version of par_raddr_err_intr. ^(†)
<prefix>_par_wdata_err_irq	Output	1	On-chip write data parity error interrupt
<prefix> _par_wdata_err_irq_fault	Output	1	On-chip write data parity error fault. This signal is a version of par_wdata_err_intr. ^(†)
<prefix>_par_rdata_err_irq	Output	1	On-chip read data parity error interrupt
<prefix> _par_rdata_err_irq_fault	Output	1	On-chip read data parity error fault. This signal is a version of par_rdata_err_intr. ^(†)
<prefix>_phy_irq_n	Output	1	Interrupt from PHY indicating alert conditions. User must treat this signal as asynchronous signal and must use a simple 2-d synchronizer.

Table Note

† This signal cannot be disabled or forced through register.

DDR4 Subsystem-to-Core AXI4 Interface Signals

The table below shows the controller-to-AXI4 interface signals

Table 7: DDR4 Subsystem Controller-to-AXI4 Interface Signals

Pin Name	Direction	Width	Description
<prefix>_awid	Input	8	AXI write address ID. Identification tag for the write address group of signals.
<prefix>_awaddr	Input	40	AXI write address. The address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in a burst.
<prefix>_awlen	Input	8	AXI write burst Length. The number of transfers in a burst associated with the write address.
<prefix>_awsize	Input	3	AXI write burst size. The size of each transfer in a burst. Byte lane strobes indicate exactly which byte lanes to update.
<prefix>_awburst	Input	2	AXI write burst type. Coupled with the size, burst type details how the address for each transfer within a burst is calculated.
<prefix>_awlock	Input	1	AXI write lock type. Provides additional information about the atomic characteristics of the transfer.
<prefix>_awcache	Input	4	AXI write cache type. Indicates the buffer-able, cache-able, write-through, write-back, and allocate attributes of the transaction. ⁽¹⁾
<prefix>_awprot	Input	3	AXI write protection Type. Indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access. ⁽¹⁾
<prefix>_awqos	Input	4	AXI write quality of service. Sideband signal to indicate the quality-of-service attributes of the write transaction. The awqos signalling is sticky, that is, it must remain stable when awvalid is asserted and awready is de-asserted. This signal determines the transaction priority for port arbitration. Higher values signify higher priority.
<prefix>_awregion	Input	4	AXI4 write address region signal. ⁽¹⁾

Pin Name	Direction	Width	Description
<prefix>_awex_auto_precharge	Input	1	AXI auto-precharge signal for write command. This port is for write address channel signal. This signal is valid when awvalid is high.
<prefix>_awex_parity	Input	1	AXI write address parity. Parity is calculated for the whole address. Type of parity should match the configuration in memory controller register for parity
<prefix>_awex_poison	Input	1	AXI write poison. Sideband signal to indicate an invalid write transaction. When asserted, no data is written to the memory. If not needed, the signal must be tied to low.
<prefix>_o_ddr_memc_axi_s_awpoison_intr_1	Output	1	Write transaction poisoning interrupt. Cleared by writing into the relevant memory controller interrupt clear register.
<prefix>_awex_urgent	Input	1	AXI write urgent. Sideband signal to indicate a write urgent transaction. When asserted, if wr_port_urgent_en register is set, causes the port arbiter to switch immediately to write. It can be asserted anytime and is not associated to any particular command.
<prefix>_awvalid	Input	1	AXI write address valid. Indicates that the valid write address and control information are available.
<prefix>_awready	Output	1	AXI write address ready. Indicates that the slave is ready to accept an address and associated control signals.
<prefix>_wdata	Input	512	AXI write data.
<prefix>_wstrb	Input	64	AXI write data byte strobe. Indicates which byte lanes to update in memory.
<prefix>_wlst	Input	1	AXI write last. Indicates the last transfer in a write burst.
<prefix>_wex_parity	Input	64	AXI write data parity.
<prefix>_wvalid	Input	1	AXI write valid. Indicates that valid write data and strobes are available.
<prefix>_wready	Output	1	AXI write ready. Indicates that the slave can accept the write data.
<prefix>_bid	Output	8	AXI write response ID. Must match the awid value of the write transaction to which the slave is responding

Pin Name	Direction	Width	Description
<prefix>_bresp	Output	2	AXI write response. Indicates the status of the write transaction
<prefix>_bvalid	Output	1	AXI write response Valid. Indicates that a valid write response is available
<prefix>_bready	Input	1	AXI write response Ready. Indicates that the master can accept the write response information.
<prefix>_arid	Input	8	AXI read address ID. Identification tag for the read address group of signals.
<prefix>_araddr	Input	40	AXI read address. The address of the first transfer in a read burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in a burst.
<prefix>_arlen	Input	8	AXI read burst length. The number of transfers in a burst associated with the read address.
<prefix>_arsize	Input	3	AXI read burst size. The size of each transfer in a burst.
<prefix>_arburst	Input	2	AXI read burst type. Coupled with the size, burst type details how the address for each transfer within a burst is calculated.
<prefix>_arlock	Input	1	AXI read lock type. Provides additional information about the atomic characteristics of the transfer.
<prefix>_arcache	Input	4	AXI read cache type. Provides additional information about the cacheable characteristics of the transfer. ⁽¹⁾
<prefix>_arprot	Input	3	AXI read protection Type. Provides protection unit information for the transaction. ⁽¹⁾
<prefix>_arqos	Input	4	AXI read quality of service. Sideband signal to indicate the quality of service attributes of the read transaction. The arqos signalling is sticky, that is, it must remain stable when arvalid is asserted and arready is de-asserted. This signal determines the transaction priority for port arbitration. Higher values signify higher priority. This signal determines also the priority of the read transfer going into the CAM depending on the programming of the PCFGQOS0_n register in memory controller
<prefix>_arregion	Input	4	AXI 4 read address region signal. ⁽¹⁾

Pin Name	Direction	Width	Description
<prefix>_arex_auto_precharge	Input	1	AXI auto-precharge signal for read command. This port is for read address channel signal. This signal is valid when arvalid is high. ⁽²⁾
<prefix>_arex_parity	Input	1	AXI read address parity. Parity is calculated for the whole address. Type of parity should match the configuration in memory controller register for parity. ⁽²⁾
<prefix>_arex_poison	Input	1	AXI read poison. Sideband signal to indicate an invalid read transaction. When asserted, all zeros are returned at the output. If not needed, signal must be tied to zero. ⁽²⁾
<prefix> _o_ddr_memc_axi_s_arpoison_intr_1	Output	1	Read transaction poisoning interrupt. Cleared by writing into the relevant interrupt clear register in memory controller.
<prefix>_arex_urgent	Input	1	AXI read Urgent. Off-band signal to indicate a read urgent transaction. When asserted, if rd_port_urgent_en register is set, causes the port arbiter to switch immediately to read. It can be asserted anytime, it is not associated to any particular command. ⁽²⁾
<prefix>_arvalid	Input	1	AXI read address Valid. Indicates that valid read address and control information are available.
<prefix>_arready	Output	1	AXI read address Ready. Indicates that the slave is ready to accept an address and associated control signals
<prefix>_rid	Output	8	AXI read ID. Must match the arid value of the read transaction to which the slave is responding
<prefix>_rdata	Output	512	AXI read Data.
<prefix>_resp	Output	2	AXI read Response. Indicates the status of the read transfer.
<prefix>_rlast	Output	1	AXI read Last. Indicates the last transfer in a read burst.
<prefix>_rex_parity	Output	64	AXI read parity/ECC. Generated by the memory controller. Byte wise parity is generated based on parity type selection in configuration of memory controller register for parity.

Pin Name	Direction	Width	Description
<prefix>_rvalid	Output	1	AXI read Valid. Indicates that the required read data is available and the read transfer can complete.
<prefix>_rready	Input	1	AXI read Ready. Indicates that the master can accept the read data and response information.

Table Note

1. This signal is not used by the controller and must be tied off to 0.
2. Not available on the NoC interface.

DDR4 Subsystem to Memory Interface Signals

The table below summarizes the DDR external memory to PHY interface signals:

Table 8: DDR4 Subsystem to Memory Interface Signals

Pin Name	Direction	Width	Description
<prefix>_bp_memreset_l	Output	1	SDRAM reset pin controlled by PHY.
<prefix>_a[13:0]	Output	14	SDRAM address bus.
<prefix>_a17	Output	1	SDRAM address signal.
<prefix>_act_n	Output	1	SDRAM activate input command.
<prefix>_ba[1:0]	Output	2	SDRAM bank address select within a bank group.
<prefix>_bg[1:0]	Output	2	SDRAM bank group select.
<prefix>_alert_n	input	1	SDRAM alert signal.
<prefix>_cas_n	Output	1	SDRAM CAS control signal.
<prefix>_cid	Output	1	SDRAM chip Id select; These inputs are used only when devices are stacked for x4 and x8 configurations using TSV.
<prefix>_ck_n[3:0]	Output	4	SDRAM differential clock signal.
<prefix>_ck_p[3:0]	Output	4	SDRAM differential clock signal.
<prefix>_cke[3:0]	Output	4	SDRAM clock enable control signal
<prefix>_cs_n[3:0]	Output	4	SDRAM chip select.

Pin Name	Direction	Width	Description
<prefix>_dq_0[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 0.
<prefix>_dq_1[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 1.
<prefix>_dq_2[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 2.
<prefix>_dq_3[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 3.
<prefix>_dq_4[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 4.
<prefix>_dq_5[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 5.
<prefix>_dq_6[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 6.
<prefix>_dq_7[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 7.
<prefix>_dq_8[7:0]	Bidir.	8	SDRAM bidirectional data bus, byte lane 8.
<prefix>_udqs_n[8:0]	Bidir.	9	SDRAM differential strobes, used for ×4 mode.
<prefix>_udqs_p[8:0]	Bidir.	9	SDRAM differential strobes, used for ×4 mode.
<prefix>_ldqs_n[8:0]	Bidir.	9	SDRAM differential strobes, used for ×8 mode.
<prefix>_ldqs_p[8:0]	Bidir.	9	SDRAM differential strobes, used for ×8 mode.
<prefix>_odt[3:0]	Output	4	SDRAM on-die termination control signal.
<prefix>_par	Output	1	SDRAM parity bit for command and address.
<prefix>_ras_n	Output	1	SDRAM RAS control signal.
<prefix>_we_n	Output	1	SDRAM write enable control signal.
<prefix>_dm_dbi_udqs_p[8:0]	Bidir.	9	SDRAM data mask/data bus inversion signal.

More information on the DDR4 device-level pins can be found in the *Speedster7t Pin Connectivity User Guide* (UG084), and the power-level requirements for the DDR signals can be found in the *Speedster7t Power User Guide* (UG087).

Chapter - 7: DDR4 IP Software Representation in ACE

Overview

The DDR4 Interface IP Generation in ACE is a GUI that helps generate and integrate the DDR4 subsystem in a Speedster7t FPGA based on specific inputs provided by the user. The Speedster7t I/O Ring Toolkit in ACE supports the integration of the chosen IP for the user design and allows the user to select the pin placements and visualize package routing. Once the desired configuration is achieved via the IO Ring Toolkit, the ACE tool generates a bitstream for the IP interface which is independent of the bitstream generated from the FPGA fabric. The tool further integrates both these bitstreams into a single full-chip bitstream that enables configuration of a Speedster7t device.

The following steps provide a brief description on creating a DDR4 IP interface user design:

Step 1 – Create a Project

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

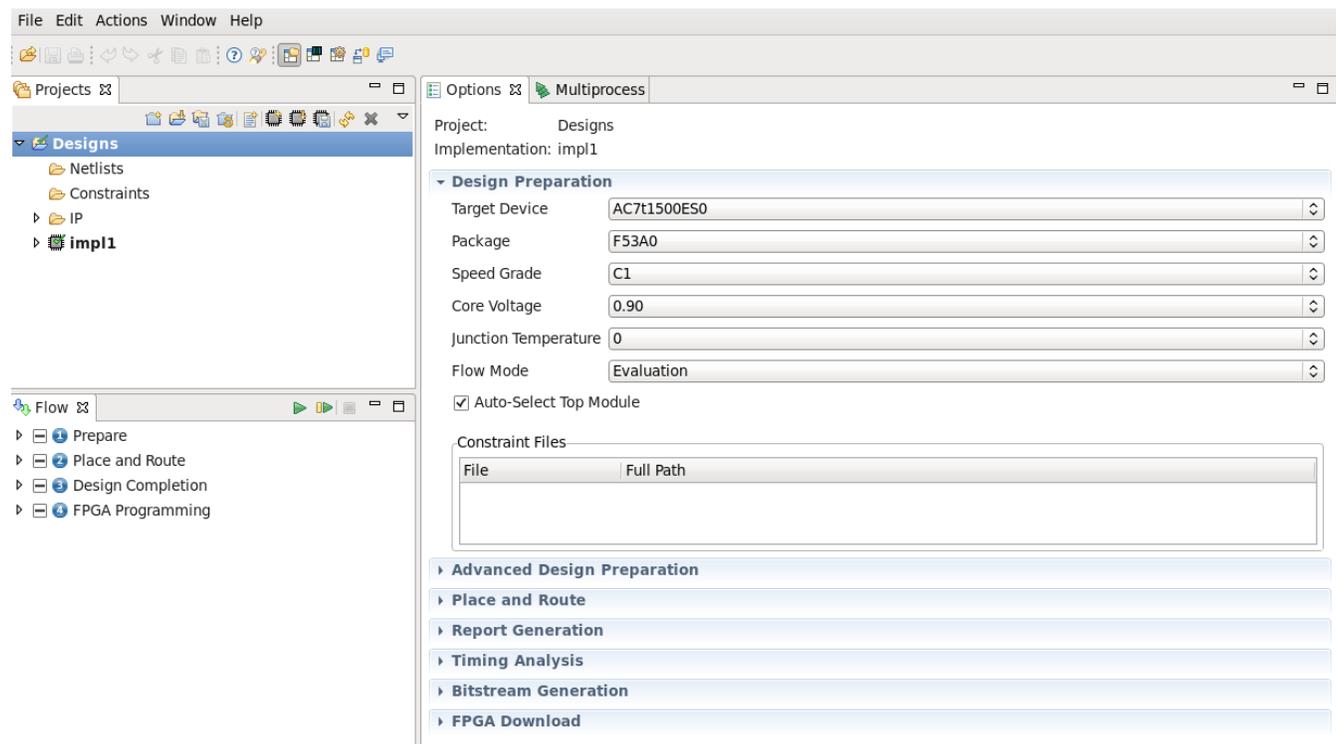


Figure 10: Design Preparation Options in the ACE Project

Step 2 – Configure the Programmable I/O

Switch to the 'IP Configuration' perspective and select the **Programmable I/O IP** from the Speedster7t **IO Ring** to create the external input clock source; select **Add** new I/O and provide the instance name, bank type, signal type, I/O standard, the desired ball placement and frequency for the I/O. Once the selection is made, the Layout Diagram highlights the chosen I/O. The figure below shows a CLKIO being created. The programmable I/O ACXIP file will also be used for any other top-level GPIO or CLKIO, and multiple I/O can be added in the same ACXIP file.

Note



The external reset inputs also have to be configured in similar fashion. This capability will be available in future ACE releases.

The 'IP Problems' window highlights any errors or warnings that occurred while configuring the programmable I/O. Since the DDR4 interface is on the south edge of the FPGA, the south-west or south-east PLL clocks are preferable.

Note



Actual signal names are prefixed with whatever was provided at IP configuration.

I/O Buffer Instance	Na	Placement	Ball Na	Device Port	Bank	I/O Standard	VDD
sys_clk							
sys_clk_pad_n		✓ CLK_SW_REFIO_0_N	AW34	CLKIO_SW_REFIO_N_0	BANK_CLKIO_SW	LVCMOS_15	1.5
sys_clk_pad_p		✓ CLK_SW_REFIO_0_P	AY34	CLKIO_SW_REFIO_P_0	BANK_CLKIO_SW	LVCMOS_15	1.5

Figure 11: Programmable I/O IP Configuration

Step 3 – Configure the PLL IP

Next, configure the PLL IP with the desired placement and appropriate clock output frequencies based on the user-specified data rate for the DDR4 interface. The DDR4 IP requires a clock running at 200 MHz to drive the NoC interface and an input reference clock at maximum of 800 MHz for the controller and PHY operations which are generated by the PLL. If the user opts for direct connect option, a third AXI clock is required to be used from the global clock signals for the direct connect AXI interface to the FPGA fabric. The number of PLL clock outputs and their respective frequencies should match the number of required clock inputs and respective rates for the subsystem to operate correctly.

Figure 12: PLL IP Configuration

Step 4 – Configuring the NoC

If the DDR4 subsystem interfaces with the NoC, then the user must set the NoC clock to 200 MHz. The global PLL clock output drives the NoC interface, which has been setup already in step 3; hence, the clocks with the expected range of frequency for NoC shows up at the tooltip. The NoC IP GUI also allows the user to set the NoC network access control configuration options.

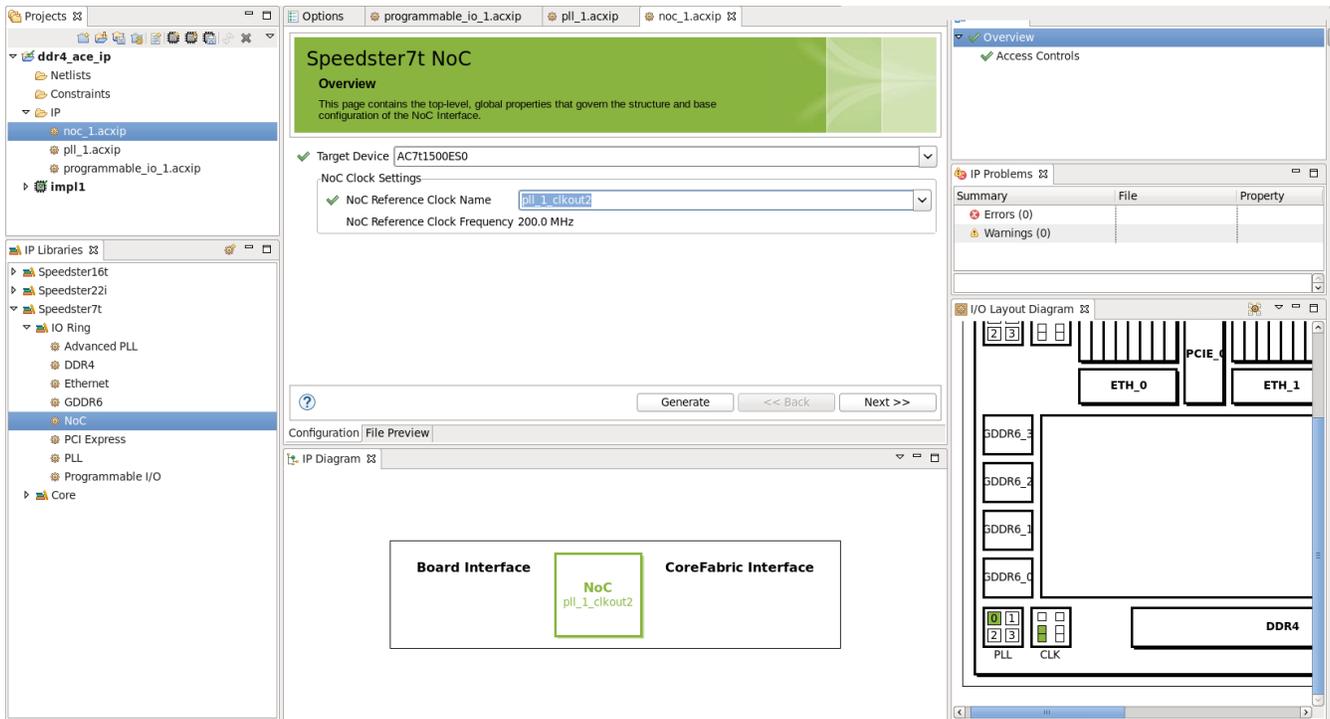


Figure 13: NoC IP Configuration

Step 5 – Configure the DDR4 Interface

Next, configure the DDR4 subsystem interface by selecting DDR4 IP in the IP library section to create an ACXIP file. Choose the desired memory part number and the DIMM data width. ACE will populate some fields such as DIMM type, rank and data rate based on the memory device chosen. The DDR4 clock settings will show the available valid clock input selections for the DDR4 reference clock, the NoC clock and the the direct connect interface AXI clock based on the clock outputs generated by the PLL. As the Speedster7t DDR4 subsystem can also be connected directly to fabric interface, there is the option of enabling the interface-to-fabric pins that helps check all signals to the core. The IP diagram shows the signals originating from the board interface connecting to the package balls on one side and those that connect to the core fabric interface on the other side.

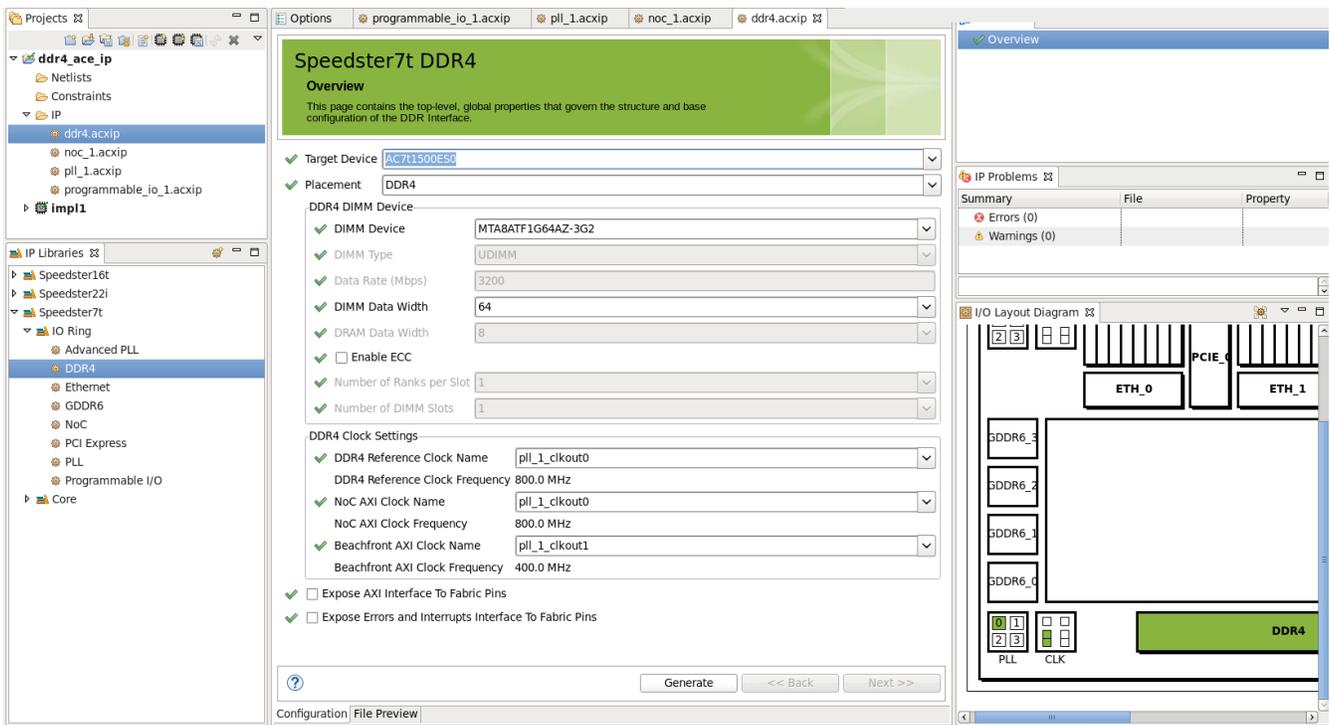


Figure 14: DDR4 Subsystem Configuration

Step 6 – Check for Errors

After all the configuration options are selected, check the 'IP Problems' window report for any errors or warnings. If there are no errors reported, the entire I/O interface with all the required IP has been integrated properly and will close timing at the specified data rate.

There is also a fully interactive 'I/O Package Diagram' view. If the user hovers over the package diagram where the DDR4 pins were placed, all information related to these pins will show up at the tooltip. The 'I/O Pin Assignments' is also an interactive map that lists all pins utilized for the DDR4 design and any other IP present in the user design.

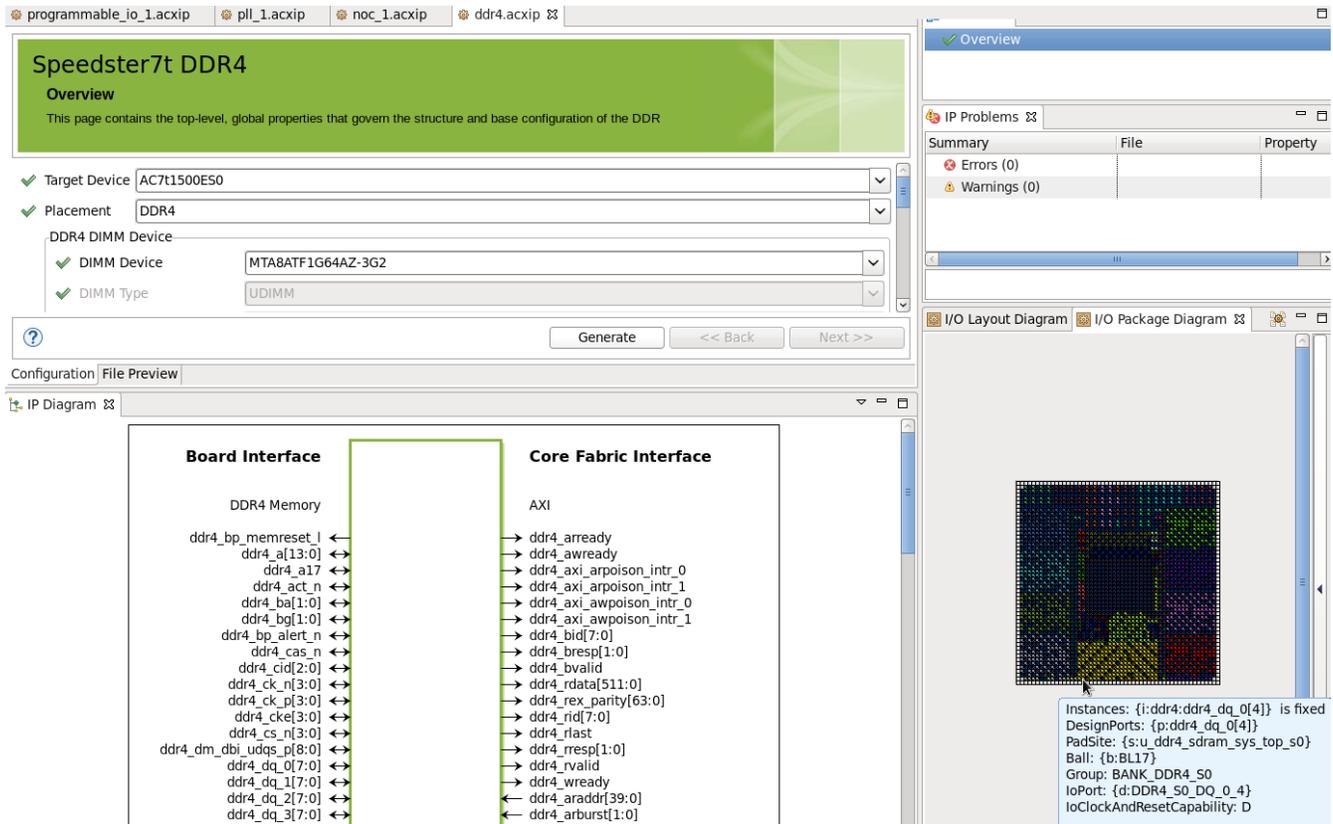


Figure 15: DDR4 Subsystem's Interactive Tools

Step 7 – Generate the Design Files

Once the configurations are verified, click the **Generate** option in any of the ACXIP files to generate all the necessary files, including the bitstream for the entire I/O ring. Clicking this icon also generates the necessary simulation models and placement files required for integration with the core design, completing the I/O ring configuration. The files generated are:

- SDC file with timing constraints for all clocks exposed to the FPGA fabric
- PDC file with pin placements for DDR4 DC interface pins (if enabled) and all GPIO pins used (by the user design)

The user can now switch to building 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.

Note

Full-chip output file generations will be enabled in future ACE releases.

Revision History

Version	Date	Description
1.0	19 Mar 2020	Initial release.