Speedster22i HD1000 FPGA

achronix
SEMICONDUCTOR CORPORATION

DS005 Rev. 1.2 – December 28, 2017

Highlights

- Advanced highest-density and highest-bandwidth FPGA
  - Abundant embedded hard IP for communications applications
  - Fully re-programmable, SRAM based
  - Synchronous core and I/O
  - Built on Intel's advanced 22-nm 3-D Tri-Gate process technology
- Large capacity
  - Up to 1 million effective look-up-tables
  - Up to 82 Mb of block RAM
  - 80 Kb block RAMs running at 750 MHz
  - 640 bit logic RAM (LRAM) running at 750 MHz
- Industry-standard register transfer level (RTL) synthesis support using Synplify-Pro from Synopsys
  - Rapid timing closure yielding significant time-to-market advantages
- Embedded (hard IP)
  - 10/40/100 Gigabit Ethernet MAC
  - PCI Express Gen 1/2/3, ×1, ×4, ×8 with DMA engine
  - DDR 2/3 72 bits wide
  - Interlaken
- Up to 64 channels of embedded 10.3125 Gbps SerDes:
  - PCI Express Gen 1/2/3
  - 10/40/100 Gigabit Ethernet (XFI, XAUI, XLAUI, CAUI)
  - Interlaken
  - Fibre Channel
  - SATA/SAS
  - OC48
  - CEI-6 SR/LR, CEI-11 SR
  - GPON/EPON
  - CPRI/OBSAI

Features

Table 1: *Speedster22i HD1000 Features*

Features	
Logic capacity with embedded IP (effective LUTs)	1,045,000
Programmable LUTs	700,000
Number of BRAM Instances	1,026
Number of LRAM Instances	6,156
80 Kb BRAM (total Kb)	82,080
640-bit LRAM (total Kb)	3,940
Multiplier/accumulators (BMACs)	756
SerDes Lanes 10.3125 Gbps	64
10G Ethernet MAC	24
40G Ethernet MAC	6
100G Ethernet MAC	2
Interlaken LLC	2
PCI Express LLC	2
DDR3/DDR2 controller	6
Number of PLLs	16
User (programmable) I/O	960

Packaging Options

Speedster®22i FPGAs are available in a variety of package options. Most adjacent family members are available in the same package and are pin-compatible, allowing migration between the members without a board layout change.

The package options are listed in **Table 2**.

Table 2: *Speedster22i FPGA Packaging*

Package Options	Available SerDes and I/O	
	10.3125 Gbps SerDes	User I/O
FBGA2597 (52.5-mm × 52.5-mm, 1-mm ball pitch)	64	960
FBGA1932 (45-mm × 45-mm, 1-mm ball pitch)	40	684

Speed Grade Options

Speedster22i FPGAs are available in two speed grades, C2 and C3; C2 features approximately 7% better core performance than C3. The following table details the differences between the two speed grades.

Table 3: *Speedster22i Speed Grade Differences*

Feature	C2	C3
DDR3 speed	1600 Mtr/sec	1333 Mtr/sec
Ethernet speed/functions	• 1-12 × 10 Gigabit Ethernet channels • 1 × 100 Gigabit, 1-2 × 10 Gigabit Ethernet channels • 1-3 × 40 Gigabit Ethernet channels • 1-4 × 10 Gigabit, 1-2 × 40 Gigabit Ethernet channels • 1-8 × 10 Gigabit, 1 × 40 Gigabit Ethernet channels	• 1-12 × 10 Gigabit Ethernet channels • 1-3 × 40 Gigabit Ethernet channels • 1-4 × 10 Gigabit, 1-2 × 40 Gigabit Ethernet channels • 1-8 × 10 Gigabit, 1 × 40 Gigabit Ethernet channels Interfaces have additional latency over those in C2
SerDes	Up to 10.3125Gbps	
PCIe	Gen1/2/3 ×1, ×4, ×8	
Interlaken	3.125 Gbps, 6.25 Gbps and 10.3125 Gbps; from ×4 up to ×12	3.125 Gbps and 6.25 Gbps; from ×4 up to ×12
GPIO speed	1600 Gbps	1333 Gbps

Family Overview

Speedster22i HD devices run at a maximum rate of 750 MHz and have effective densities of up to one million LUTs. Based on the Intel 22nm process, Speedster22i HD devices are SRAM based and fully reconfigurable. Logic resources are provided using standard, synchronous, 4-input LUTs. A reconfigurable logic block (RLB) contains ten LUTs, and has ten registers. Speedster22i HD devices also contain block RAMs. Each block RAM is 80 kb in size and allows true dual port access.

The I/O frame contains embedded controller IP, configurable I/O, SerDes, clock generator blocks with phase lock loops (PLLs), and the device configuration logic. Speedster22i FPGAs contain up to sixty-four lanes of 10.3125 Gbps SerDes and up to an additional 960 high-speed reconfigurable I/O. Additional dedicated hard IP includes up to six DDR2/3 PHY and controllers, up to forty eight 10 Gb Ethernet controllers, up to twelve 40G Ethernet controllers and up to four 100G Ethernet controllers. There are also are up to four Interlaken controllers and two PCI Express controllers, all available as embedded hard IP and therefore use none of the reconfigurable logic fabric and achieve maximum performance without the need for timing closure/optimization.

There are also dedicated I/O for the embedded programming and configuration logic (CFG) designed to support a variety of programming options. Dedicated clock I/O pins are located near the corners of each Speedster device. **Figure 1** gives an overview of Speedster22i Devices

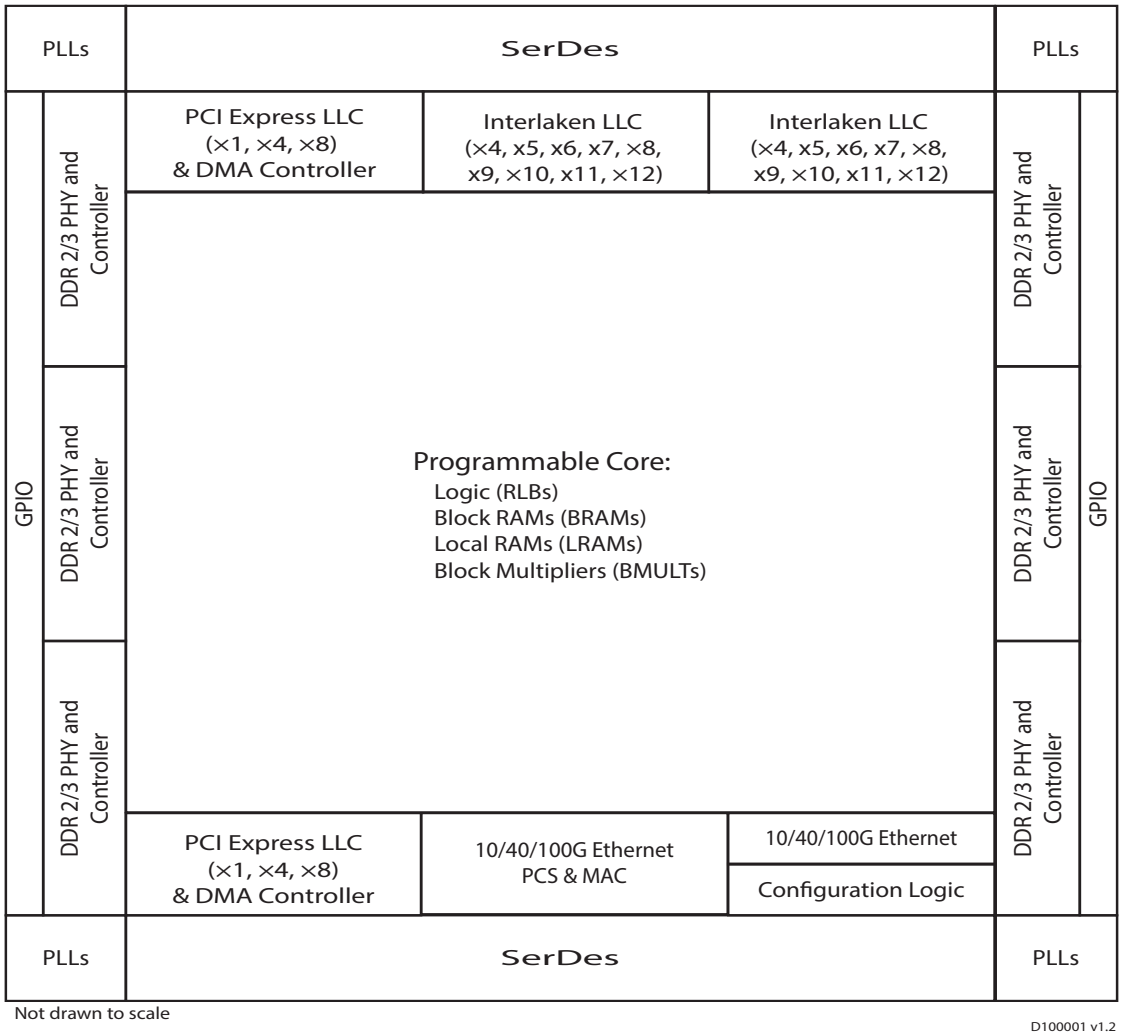


Figure 1: *Speedster22i HD Device Overview*

FPGA Core

The core of an Achronix Speedster22i HD FPGA contains columns of logic, memory and multiplier/accumulators (BMACs) connected with a global interconnect as shown in figure 2 below. Columns of reconfigurable logic blocks (RLBs) are interspersed with columns of block RAMs (BRAMs) and local RAMs (LRAMs) and BMACs. The core also includes global and local clock networks as well as reset networks. The columns of logic resources are shown in **Figure 2**:

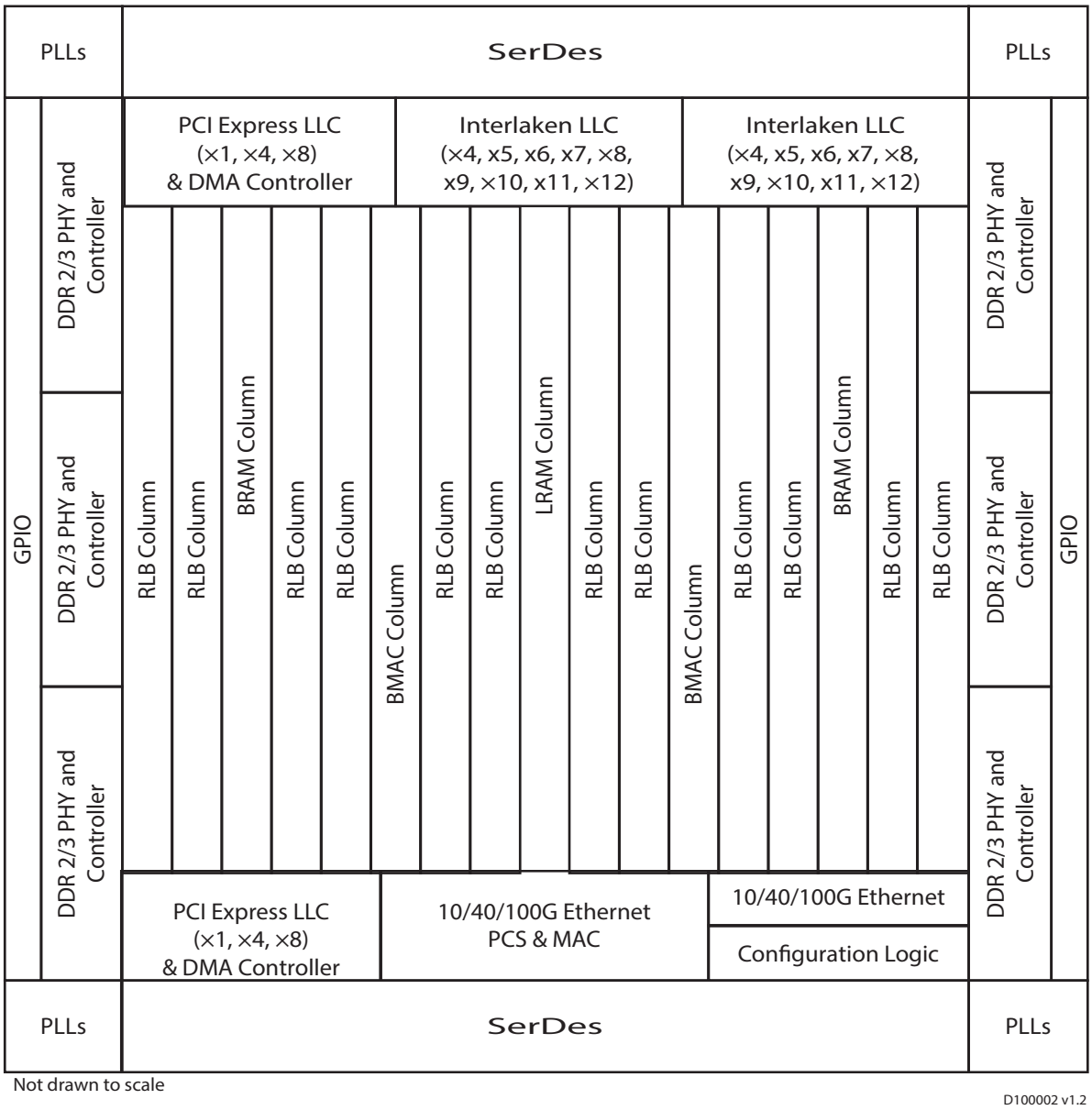


Figure 2: HD1000 highlighting Columns of RLBs, BRAMs, LRAMs and BMACs in the Programmable Core

Global Interconnect

The RLBs, BRAMs, LRAMs and BMACs are connected by a uniform global interconnect. This enables the routing of signals between core elements. Switch boxes make the connection points between vertical and horizontal routing tracks. Inputs to and outputs from each RLB/BRAM/LRAM/BMAC connect to the global interconnect. An example of an RLB with eight used inputs and two outputs is shown in **Figure 3**.

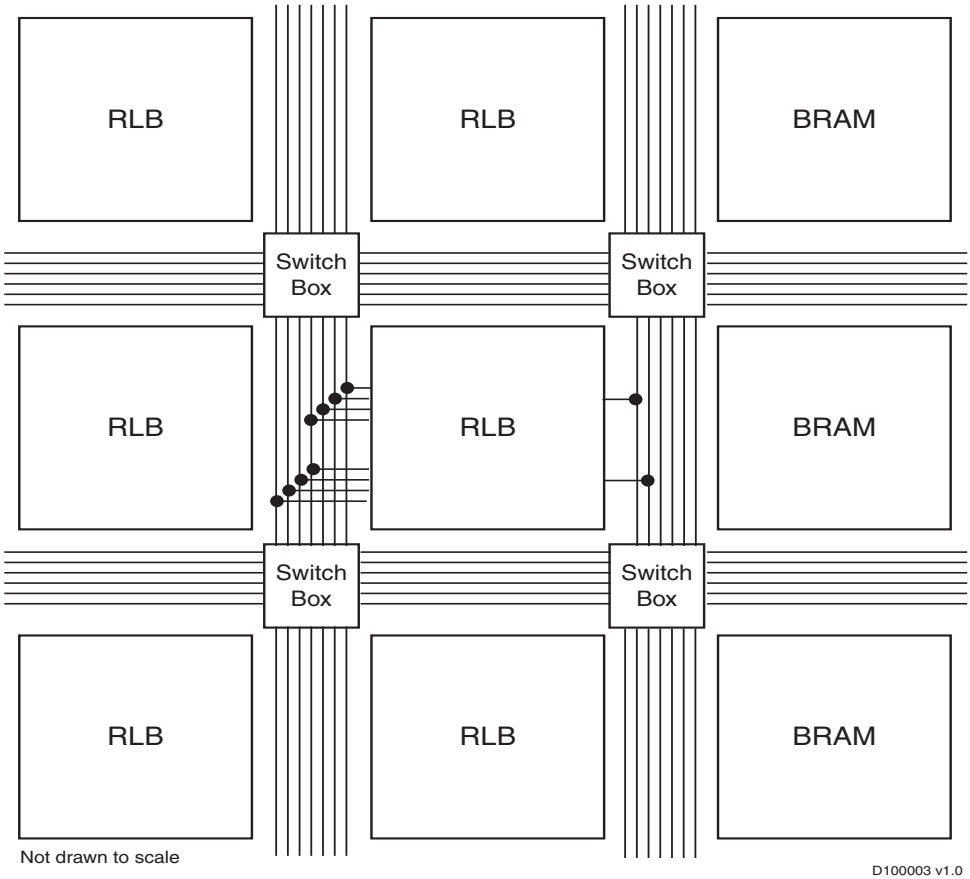


Figure 3: Global Interconnect Routing (Conceptual)

The Reconfigurable Logic Block (RLB)

The reconfigurable logic block (RLB), is comprised of five logic clusters, each of which contains two LUTs and two registers. This gives a total of ten, 4-input LUTs in a single RLB.

There are two types of logic cluster, the light logic cluster (LLC) and the heavy logic cluster (HLC). Each RLB has three LLCs and two HLCs. The HLC has greater functionality than the LLC, as it includes an advanced carry chain whereas the LLC has a MUX2.

The RLB is illustrated in **Figure 4**. Essentially, an RLB consists of five logic clusters, two of which contain a carry chain. This carry chain has dedicated carry-in and carry-out signals to allow chaining to be cascaded through multiple RLBs.

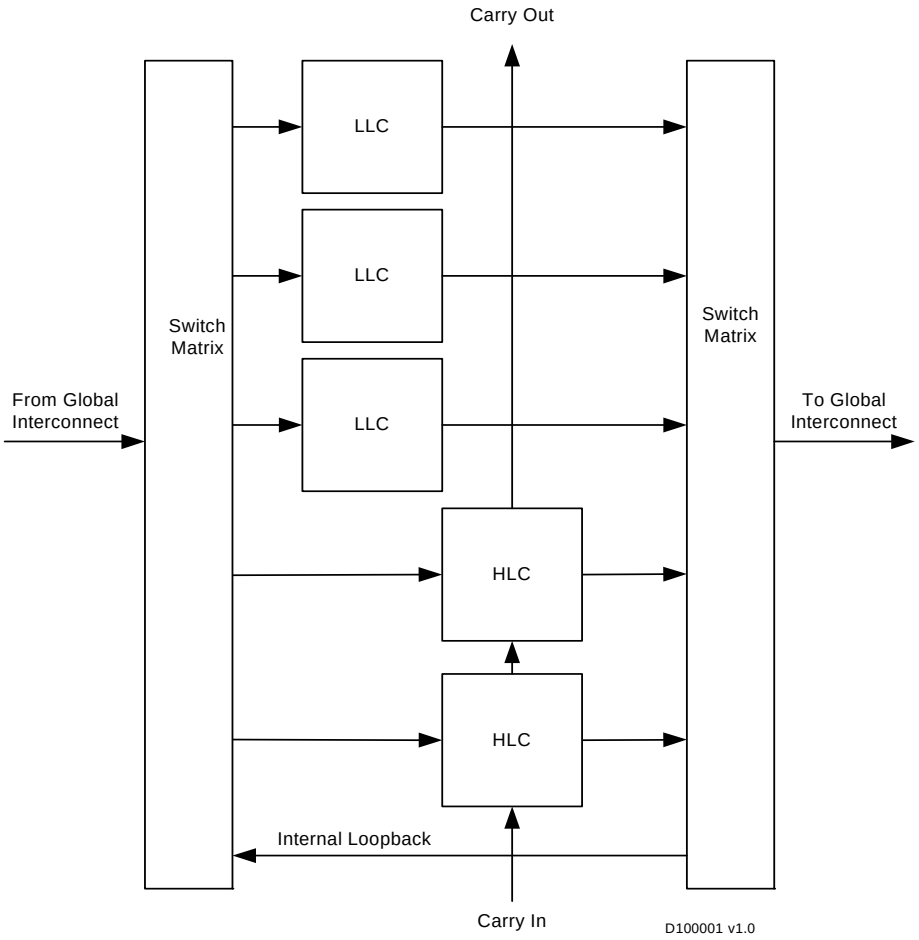


Figure 4: *The Reconfigurable Logic Block*

Efficient RLB Feedback

There are several feedback mechanisms within the RLB to allow efficient feedback, i.e feedback signals that stay within the RLB instead of having to use external routing resources.

A single signal matrix exists in the RLB. This multiplexes:

- Inputs – routes the RLB inputs and internal feedback signals to the logic clusters
- Outputs – routes the logic cluster outputs (registered and unregistered) to the RLB outputs and also back to the input matrix where feedback is required

The details of internal RLB feedback paths are shown in **Figure 5**:

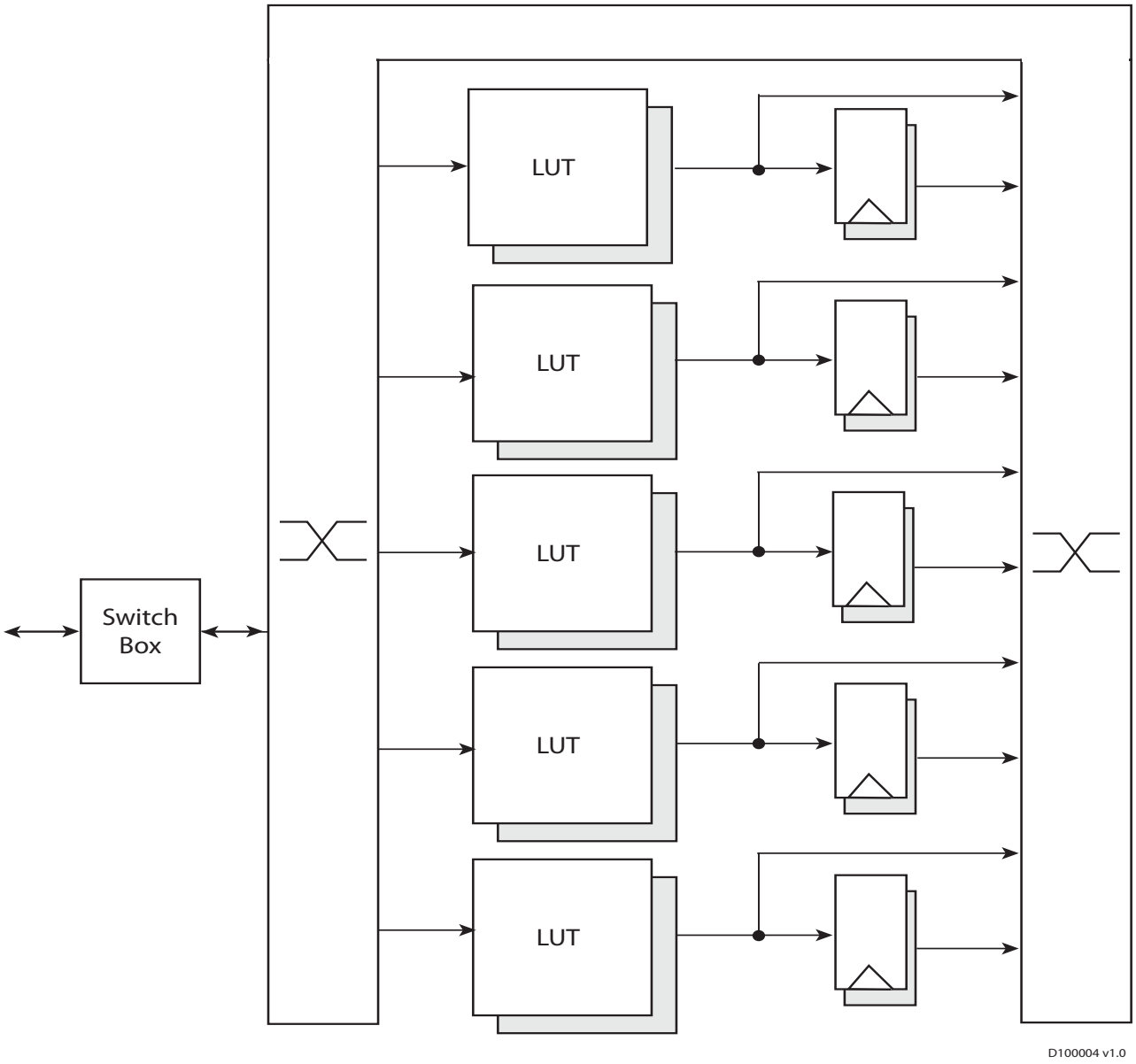


Figure 5: *The Reconfigurable Logic Block*

Both Registered and unregistered outputs from each LUT can also be routed from the output matrix back into the input matrix without leaving the RLB.

The outputs from the RLB can also be routed back into the inputs via the external Routing if needed (not shown).

The Light Logic Cluster

The light logic cluster (LLC) is illustrated in **Figure 6**.

The standard 4-input LUT is the fundamental logic building block of the fabric. Each LUT has four inputs and a single output, and can be configured to make the output reflect any combinatorial (truth table) function of the inputs. The two four-input LUTs can implement a single five-input LUT function with the utilization of the MUX2. The MUX2 also enables the implementation of certain six, seven, eight and nine-input functions.

Multiplexing blocks (shown in **Figure 6**) provide flexible access to the two register outputs.

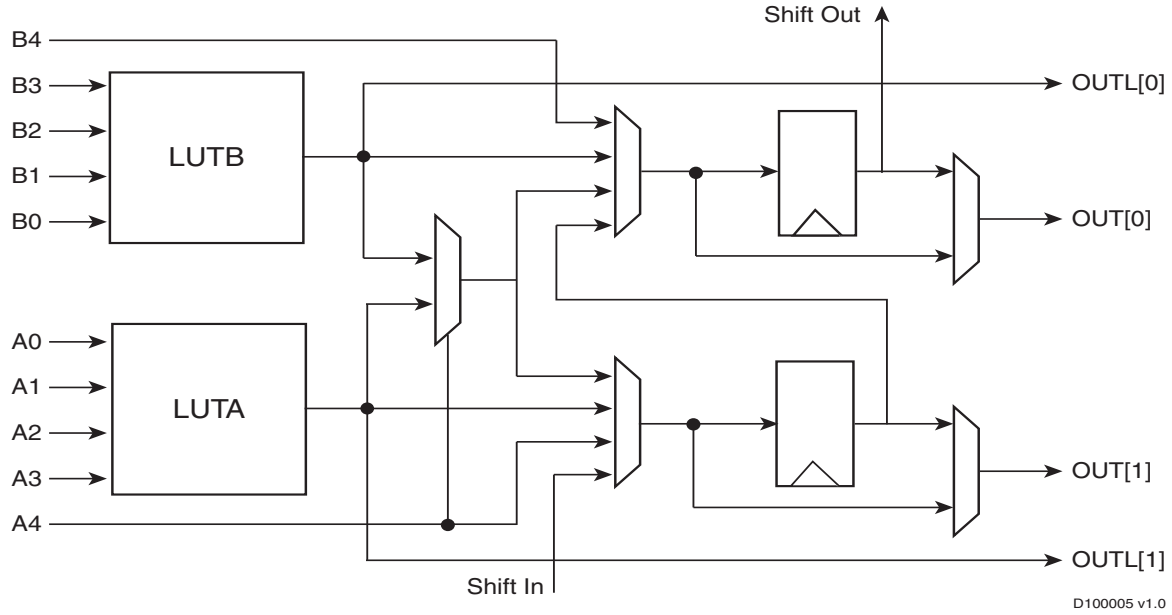


Figure 6: *The Light Logic Cluster*

The Heavy Logic Cluster

The heavy logic cluster (HLC) is illustrated in **Figure 7**. Most functionality possible with the light logic cluster can also be implemented in a HLC. In addition, each cluster has a 2-bit adder as well as the logic needed for generation of an arithmetic carry signal and propagation to the HLC to the north based on the RLB inputs and the carry in signal from its neighbor to the south.

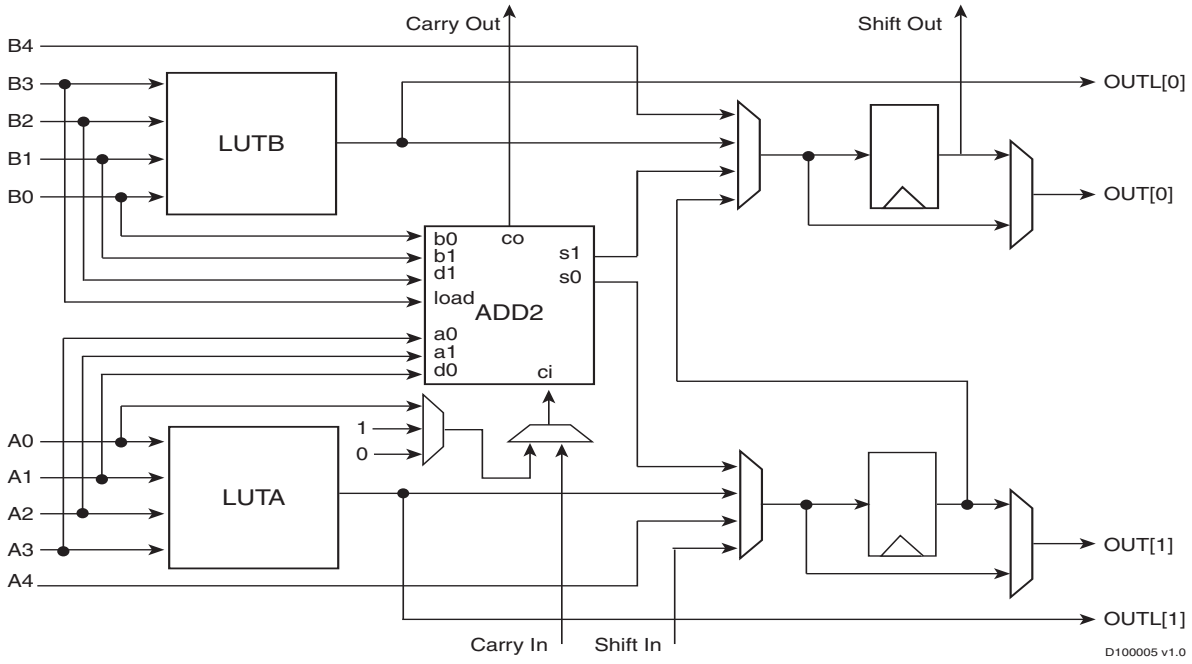


Figure 7: *Heavy Logic Cluster*

Memory Resources

Block RAMs (BRAM)

The block RAM (BRAM) contained within the Speedster22i is an 80Kbit, true dual port memory (2 independent read/write ports). The BRAM provides support for write-through and no-change modes (no support for read-first mode).

The key features (per block RAM) are summarized in **Table 4**, and illustrated in **Figure 8**:

Table 4: *Block RAM Key Features*

Feature	Value
Block RAM Size	80 Kb
Organization	2k × 40, 2k × 36, 2k × 32, 4k × 20, 4k × 18, 4k × 16, 8k × 10, 8k × 9, 8k × 8, 16k × 5, 16k × 4, 32k × 2, 64k × 1
Performance	750 MHz
Physical implementation	Columns throughout device
Number of ports	Dual port (independent read and write)
Port access	Synchronous

Organization

The organization of each block RAM port can be independently configured (the available organizations are listed in **Table 4**).

Note: *Access from opposite ports are not required to have the same organization; however, the number of total memory bits accessible by each port must be the same.*

Operation

The read and write operations are both synchronous. For higher performance operation, an addition output register can be enabled. Enabling the output register will add an additional cycle of read latency.

Write enable (wea/web) controls provide 10-bit enable control for port widths of 20 or 40 bit.

The initial value of the memory contents may be specified by the user from either parameters or a memory initialization file. The initial/reset values of the output registers may also be specified by the user. The reset values are independent of the initial (power-up) values. (They don't need to match.)

The porta_write_mode/portb_write_mode parameters define the behavior of the output data port during a write operation. When porta_write_mode/portb_write_mode is set to *write_first*, the douta/doutb is set to the value that is written on the dina/dinb port during a write operation. Setting porta_write_mode/ portb_write_mode to *no_change* keeps the douta/ doutb port unchanged during a write operation to porta/portb.

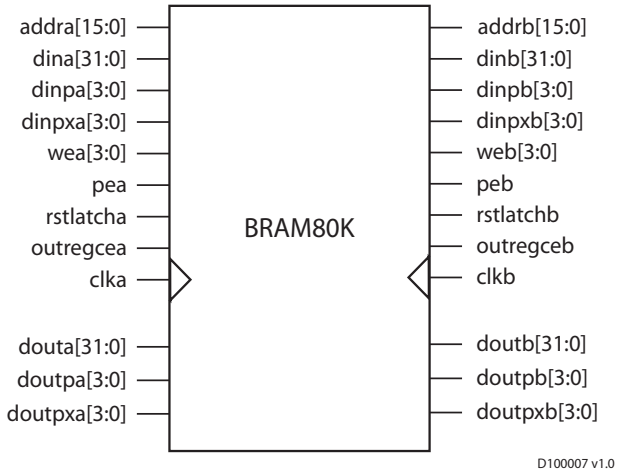


Figure 8: *Block RAM I/O*

Built-in FIFO Controller

Each BRAM has a FIFO controlled built into it. Each FIFO is capable of operating with two independent ports which have clocks that may either be the same or asynchronous with respect to each other.

Error Correction

The BRAM supports single-bit error correction (ECC) and two-bit error detection on a 32-bit wide data bus. The internal error correction circuitry is available when the BRAM is used in either the RAM or the FIFO mode.

The user implementation of parity or ECC on a 32-bit wide data bus requires an 8-bit overhead, resulting in a 40-bit wide bus. These overhead bits can be used for other purposes as well such as tagging, control functions, etc.

Initialization and Reset

Initial content of the block RAMs is loaded during device configuration. On reset, the RAM contents are unchanged.

The initial state of the RAM read outputs is also loaded during device configuration. Unlike the RAM content, this default output state is restored on reset.

Logic RAM (LRAM)

The local RAM (LRAM640) implements a 640-bit memory block with one write port and one read port. LRAMs are included in dedicated columns spread throughout the device. Each LRAM is a single 640-bit block of dedicated memory. A summary of LRAM features is shown in **Table 5**.

The LRAM ports are shown in **Figure 9**.

Organization

The LRAM640 can be configured as either a 64 × 10 simple dual-port (1 write port, 1 read port) RAM or a 64 × 10 single port (1 read/write port) RAM.

Table 5: *Logic RAM Key Features*

Feature	Value
Logic RAM size	640 bits
Organization	64 × 10
Performance	750 MHz
Physical Implementation	Dedicated Columns
Number of Ports	Simple dual port (one read, one write), or Single port (one read/write port)
Port Access	Synchronous writes, Asynchronous or Synchronous Reads

Initialization and Reset

By default, the contents of the LRAM640 memory are undefined. If the user wants the initial contents to be defined, he may assign them from either a file pointed to by the mem_init_file parameter or assign them from the value of the mem_init parameter.

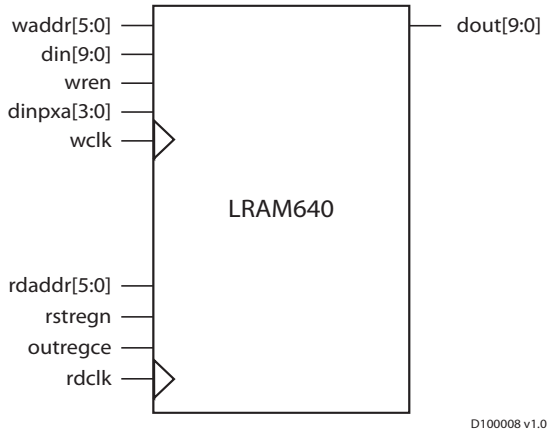


Figure 9: *Logic RAM I/O*

Operation

The LRAM640 has a synchronous write port. The read port can be configured for either asynchronous or synchronous read operations. The read port output has a register that can be bypassed.

The memory is organized as little-endian with bit 0 mapped to bit zero of parameter mem_init and bit 639 mapped to bit 639 of parameter mem_init.

Multipliers/BMAC56

The multiplier/accumulator (BMACC56) block implements a signed 28 × 28 multiplier followed by an optional accumulator block. The multiplier produces a 56-bit result which is fed into (or bypasses) the 56-bit accumulator. The key features are summarized in **Table 6**.

Table 6: *Multiplier Features*

Feature	Value
Arithmetic type	Two's complement (signed)
Performance	750 MHz
Multiplier size	28 × 28
Accumulate size	56 bits
Cascade size	56 bits

Multiplication and accumulation is an important part of real-time digital signal processing (DSP) applications ranging from digital filtering to image processing. Speedster 22i HD devices have numerous BMACC56 instances arranged in columns. Each BMACC56 block has a 56-bit cascaded path interconnecting adjacent (north/south) BMACC56 blocks.

The BMAC I/O are illustrated in **Figure 10**.

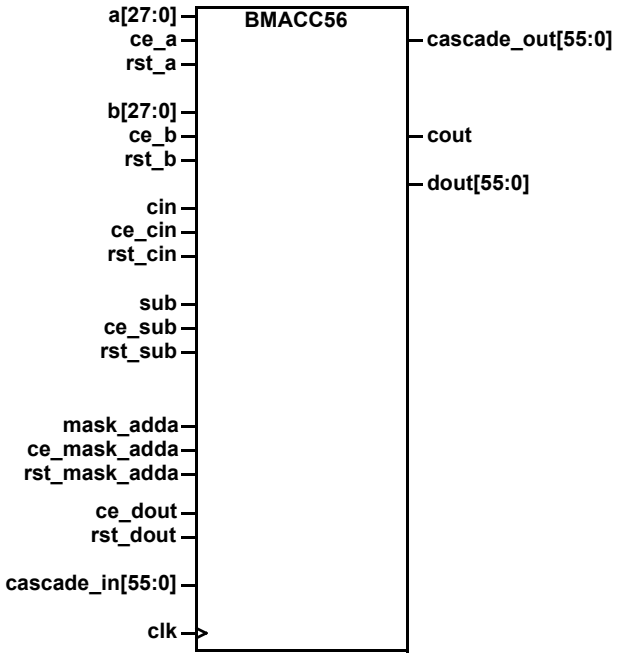


Figure 10: *BMAC I/O*

The internal block diagram is shown in **Figure 11**:

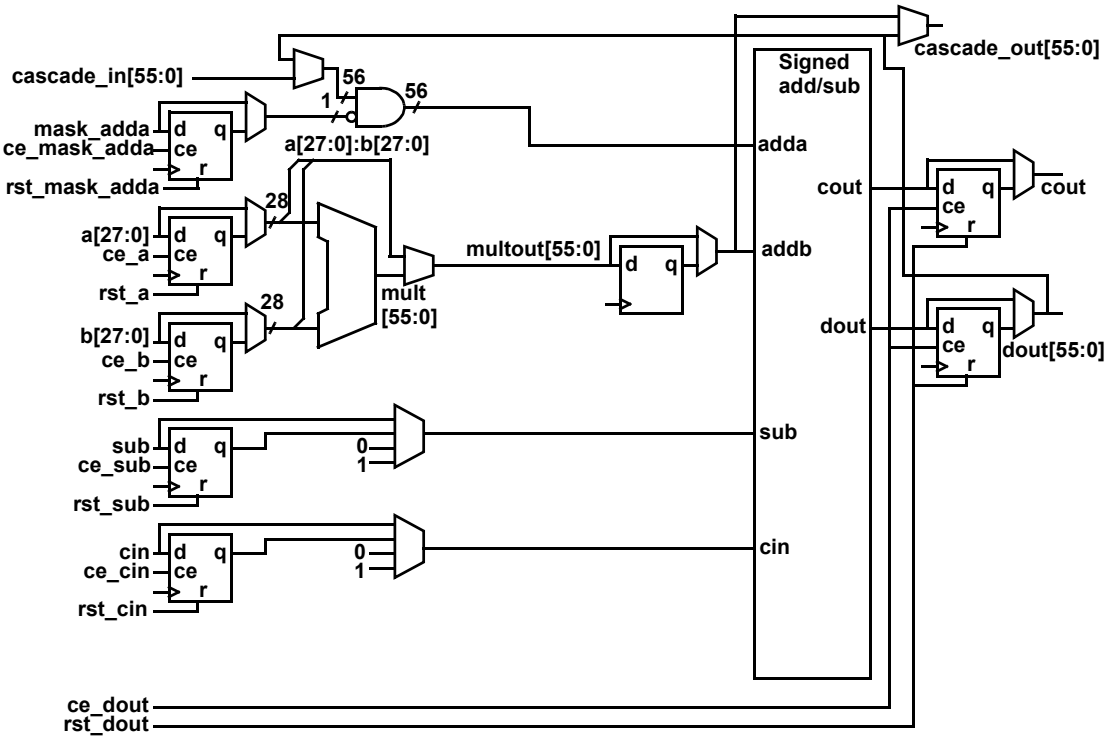


Figure 11: *BMAC Block Diagram*

Clocking and Reset Resources

Global Clock Network

Speedster22iHD FPGAs have two hierarchical clock networks: a global clock network and a direct clock network. These two clock networks have common input sources: Direct clock input pins and PLL outputs which make up the Global Clock Generator (GCG), recovered SerDes input clocks and byte lane clocks from the GPIO. These input sources get channeled in from both the north and south sides of the device, and are then fed into the FPGA core.

The global clock network is a balanced clock tree that enables clock distribution to all parts of the Speedster 22iHD FPGA. Clock signals are routed through the clock hub to the center of the device, and are then provided to all clock regions on both the west and east sides of the device. **Figure 12** below provides a high level illustration of the routing and connection paths for the global clock network.

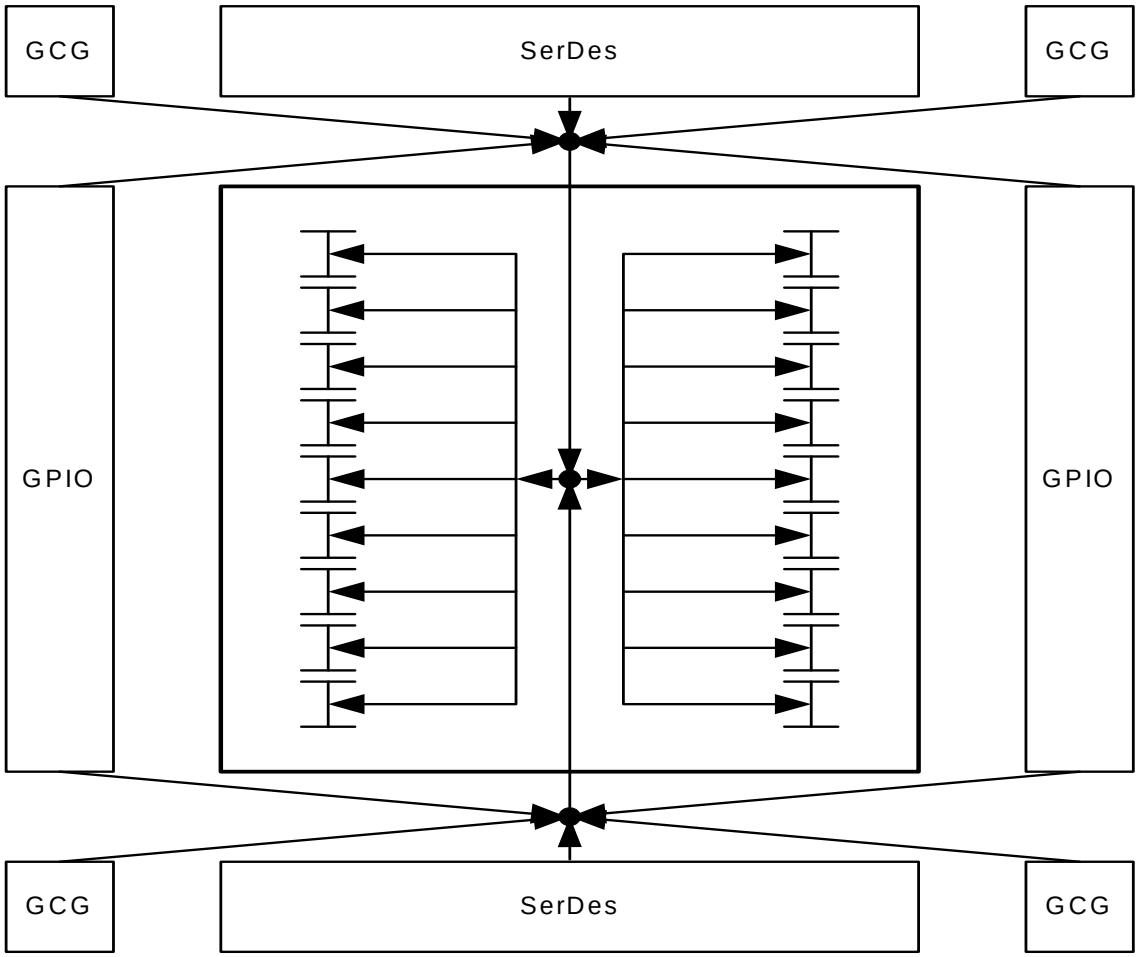


Figure 12: *Global Clock Network*

The direct clock network is a distribution system that provides for much lower clock insertion delay, which is particularly useful for more complex designs that utilize multiple clocks and require clocks to be internally generated and re-distributed to certain parts of the FPGA. Each branch of the direct clock network is restricted to the clock region it reaches as shown in **Figure 13** below.

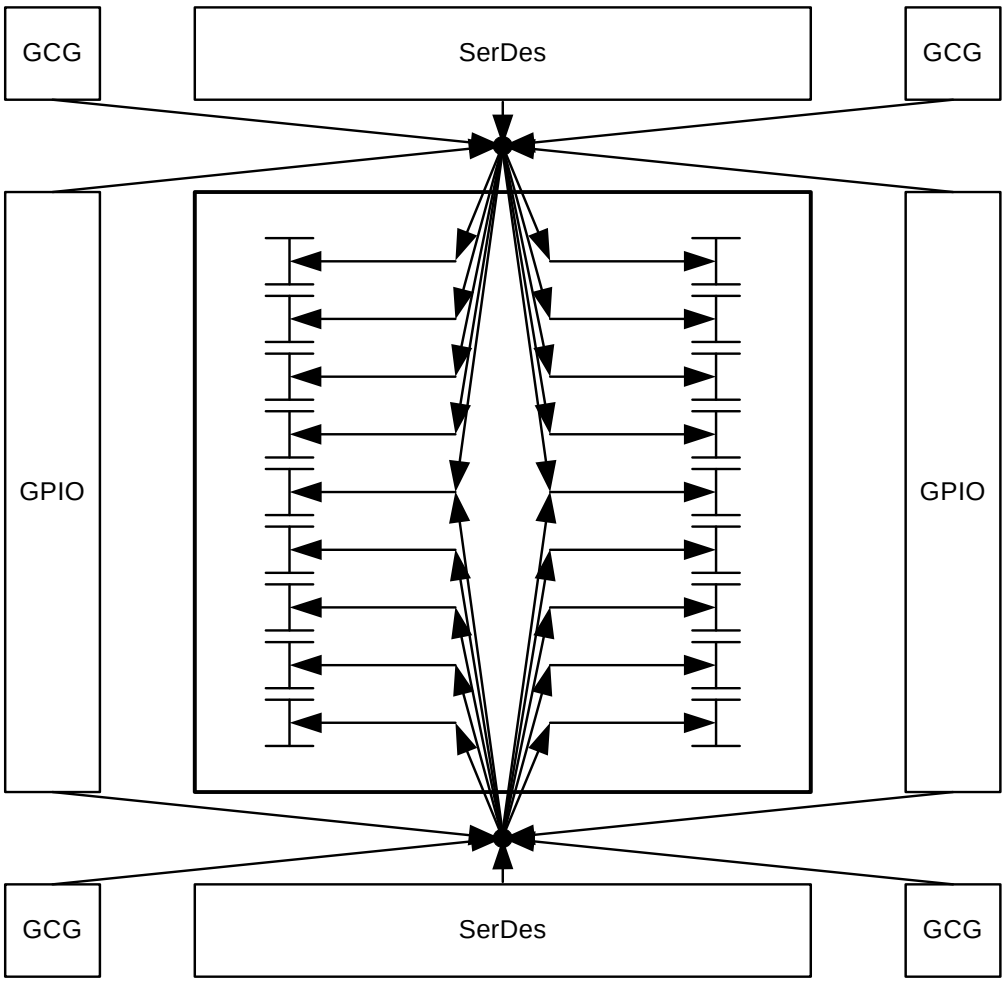


Figure 13: *Direct Clock Network*

Clock Network Components

The clock network in the core of the Speedster 22iHD FPGA is made up of the following building blocks:

- Top and bottom clock muxes to select a set of clock signals coming from that part of the FPGAs clock input sources as mentioned earlier.
- Clock hub in the center of the device to mux between the clock signals coming from the bottom of the chip, the top of the chip and those generated in the data interconnect network. The clock hub then distributes the muxed clock signals to clock regions in all parts of the chip.
- Distinct clock regions: half on the east side of the device and half on the west side. Every clock region contains a Regional Clock Manager (RCM), which selects from the incoming direct and global clocks and distributes these to the columns within that clock region in the FPGA fabric.

A high-level block diagram of the clock network components, and a more detailed view of a clock region are shown in **Figure 14** and **Figure 15** below.

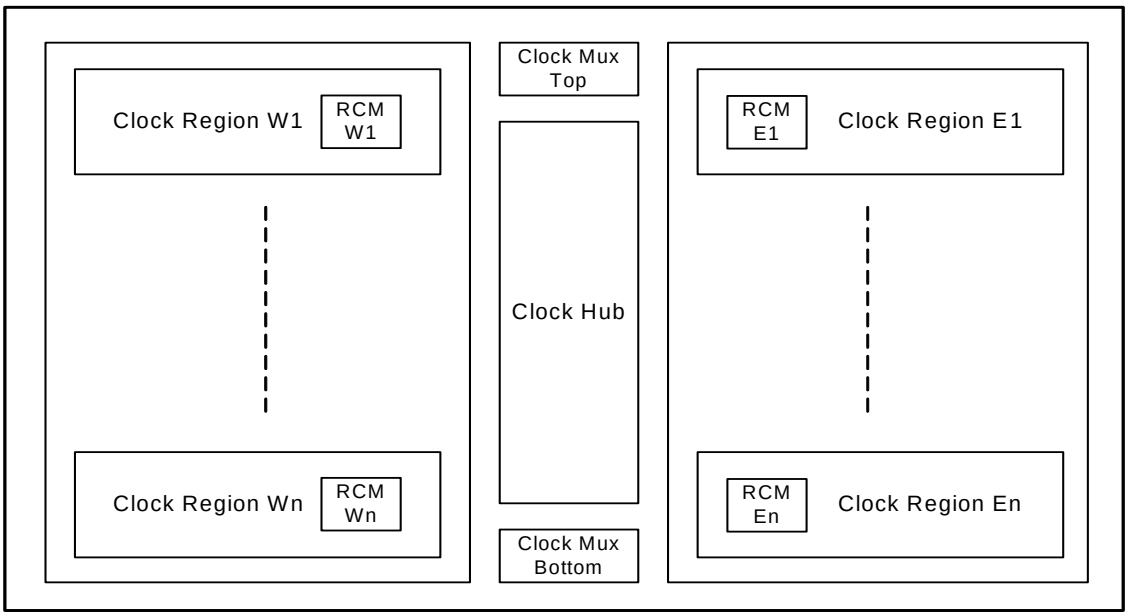


Figure 14: *Core Clock Network Block Diagram*

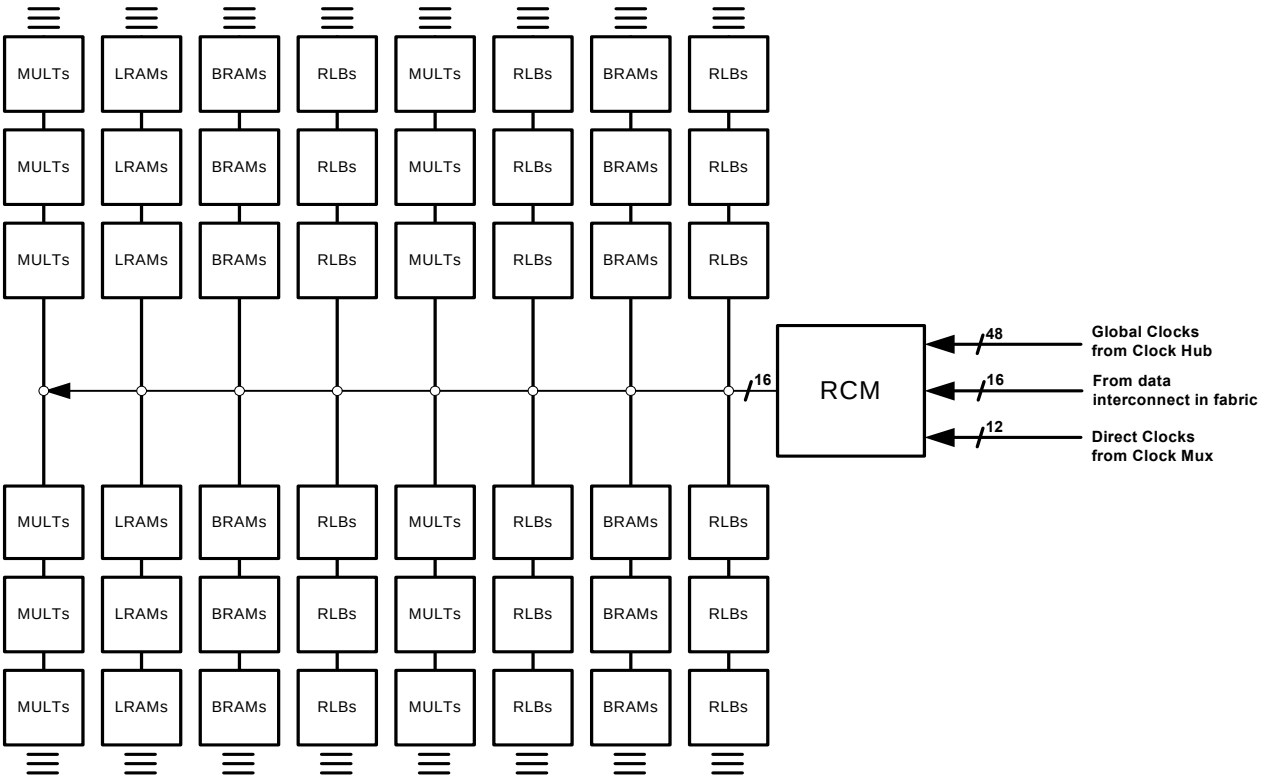


Figure 15: *Detailed View of a Clock Region*

Clock Sources

The three clock network sources are Global Clock Generators (GCGs), recovered SerDes input clocks and byte lane clocks from the GPIO. The following provides some additional detail on these.

There are four GCGs on a Speedster 22iHD FPGA, one in each corner of the device. Each GCG contains six Global Clock I/O buffers (CBs) and four Phase Locked Loops (PLLs). The clock buffers can be used either as three differential I/O or six single-ended I/O. If these

I/O are not used as clock buffers, they can be used as generic inputs or outputs.

The PLLs are low jitter, wide range, independent multi-phase outputs with glitch-free phase rotators that can be used for PLL outputs of up to 1066 MHz for core circuit applications. The block diagram below shows a high-level view of the PLL architecture.

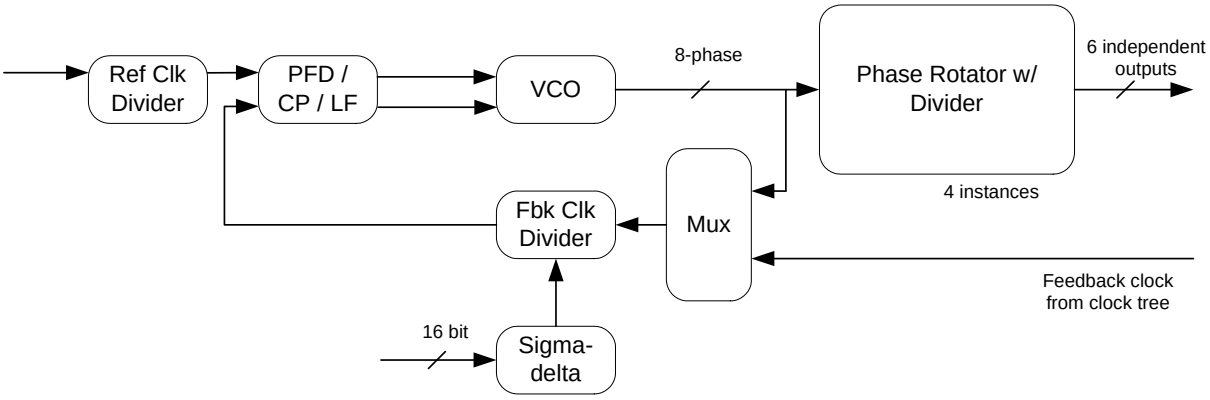


Figure 16: *PLL Architecture*

The reference clock, which can come from a global clock I/O buffer or other GCG, is divided by the reference clock divider (6-bit: 1 to 63) before being sent to the PFD. The supported reference clock range is from 66MHz to 400MHz (after reference clock divider).

The VCO operating range is between 1.25 GHz and 2.5GHz. The VCO generates 8 equally separated phases, one of which is sent to the feedback divider through a mux to allow the PLL running in short loop operation without de-skew. All 8-phases are sent to 4 phase rotators which can independently select one of the 8 phases. This then goes through an output divider (6-bit: 1 to 63) before being sent out of the PLL block. One of the 4 output clocks, after going through the clock distribution tree, has an option to be sent to the feedback divider for de-skew functionality. The feedback divider has two modes of operations: one is integer mode which offers from 2 to 255; the other is fractional mode with resolution of 16-bit. In the fractional mode, the integer divider range is limited to 8 to 254.

The PLL performance specifications are listed in **Table 7** below.

Table 7: *Clock Generator PLL Performance Specifications*

Performance Specifications		Units
Reference Clock Frequency Range (Post Input Clock Divider)	66 – 400	MHz
VCO Operating Range	1.25 – 2.5	GHz
Bandwidth	Tracking between 1/10th and 1/8th of ref clk frequency	
Feedback Divider	8 (2 to 255) In fractional mode, only supports 8 to 254	Bits
Post Divider	6	Bits
Reference Clock Divider	6	Bits
Number of Post Dividers	4	
Fractional Synthesizer Support	PLL includes a 16-bit accurate fractional synthesizer	
Spread Spectrum	No support	
Feedback signal delay(max)	Half of divided reference clock period	ns
Operation Mode	Normal, Bypass, Pwrdn, Reset	
Internal phase separation	12.5% output cycle	%
Internal phase accuracy	±3.5% output cycle at 2GHz	%

Table 7: *Clock Generator PLL Performance Specifications (Continued)*

Performance Specifications		Units
Output phase accuracy	±5% output cycle at 2GHz	%
Number of selectable Phases	Eight. Each PLL output clock can select and change to one of the 8 phases dynamically in a glitch-free manner.	
Maximum Duty cycle variation	50% ± 2%	%
Static Phase Error	± 80ps	ps
Jitter – Period	± 4% p2p of output clock period	%
Jitter – Cycle to Cycle	5ps (integer divider mode, typical); 8ps (fractional divider mode, typical)	ps
Jitter – Long Term	worst case 100ps/sigma	ps
Lock Time	500 ref clk periods (integer mode); 1000 ref clk periods (fractional mode)	
Reset divide-by-1 output frequency range	30MHz – 50 MHz	MHz

For the clocks generated in the SerDes, each SerDes lane has its own reference clock, and its own pair of PLLs:

- A transmit PLL, which synthesizes the transmit clock directly from the reference clock, and then a slower transmit word-clock for data-input from the fabric,
- A receive PLL, which synthesizes a receive bit-clock (and corresponding word-clock) from the incoming data-stream.

Thus each SerDes lane provides two word-clocks (transmit and receive) to the fabric.

Clocks can also be sourced from the byte lanes in the GPIO. For clocks that need to be routed to a FPGA fabric in a balanced fashion, going through the clock mux and hub and using either the global or direct clock networks is recommended. However, there does exist an alternative whereby the clock from the GPIO can be routed directly into the fabric along with the data. For source synchronous transfers, especially when the logic needing to be clocked this way is not significant,

this approach can be extremely useful. These are referred to as the byte-lane clock networks.

A byte lane in a Speedster22i FPGA consists of 12 I/O buffers. Two of these I/O buffers are clock capable and can be used to receive or send a clock. These buffers can be used as one differential pair for a clock or as two single-ended buffers for two clocks. Each of these received clocks can optionally be delayed using a DLL. The byte-lane clock network supports four byte lanes in a repeating fashion. For example, there are three byte-lane clock networks in an I/O Bank with 12

bytes: byte 0 to 3, byte 4 to 7, and byte 8 to 11. Each byte-lane clock network can function as follows:

- Eight by-9 clock networks
- Four by-18 clock network
- One by-36 clock network

Reset Resources

Reset Sources and the Reset Input Block

Each corner of a Speedster22i FPGA has an individual Reset Input Block. This block receives external reset inputs as well as inputs generated internally within the device. External reset inputs can be driven by dedicated clock pads as well as a number of GPIOs located in the East-North (EN), East- South (ES), West-North (WN) or West-South (WS) sides of the device. Internal reset inputs are driven through data and clock paths in the logic fabric.

The inputs to the Reset Input Block generated either externally or internally are required to be active-low and glitch free. The input resets can be either asynchronous or synchronous. An asynchronous reset is synchronized for de-assertion to each and every clock domain where it is utilized. A synchronous reset does not need to be synchronized to the same clock domain but is synchronized when used in any other clock domain not synchronous with the current clock domain.

Reset Distribution

Reset signals need to be distributed to both the FPGA core fabric as well as the I/O ring, which includes the GPIOs, SerDes and hard IP. There is no dedicated reset network in the FPGA programmable fabric, so distribution to the core is generally recommended to be done using the clock network resources described above, to take advantage of timing and load balancing.

For the I/O ring, there actually is a dedicated 16-bit reset bus that ensures a balanced reset assertion and de-assertion latency across the entire device. This is made possible by pipelining the reset distribution using the clock to which the reset is synchronized.

Each side of the device has two groups of reset signals running in opposite directions. Each group consists of eight reset signals each, spanning the entire edge of the device in a pipelined manner. The two groups of reset signals are tapped at each I/O bank or logic block (eg DDR controller, SerDes), using a configurable pipeline multiplexer with configurable pipelined latency. The configuration is set for each multiplexer individually to balance the latency for each reset signal across the entire device. The outputs of the pipeline multiplexer are subsequently distributed to the reset network inside the I/O banks and logic blocks. This is shown in **Figure 17** below.

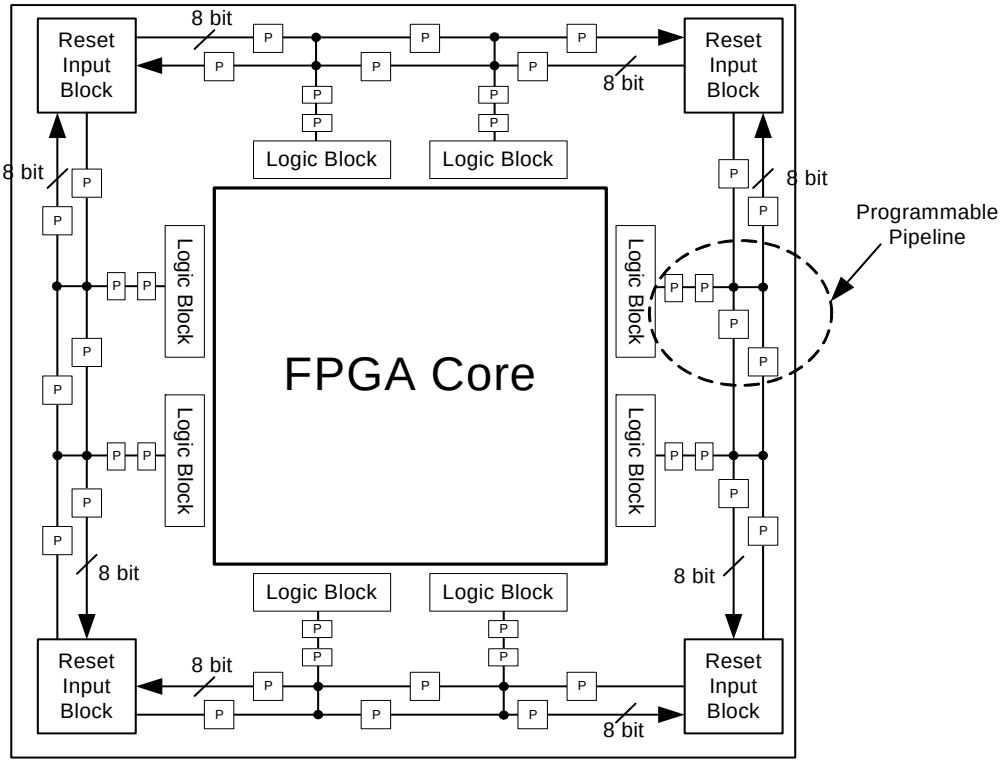


Figure 17: *I/O Ring Reset Network*

Embedded (Hard) IP

Speedster22i HD devices include several embedded (Hard) IP blocks. These implement the following protocols:

- DDR 2/3
- PCI Express Gen 1/2/3, ×1, ×4, ×8
- 10/40/100Gbit Ethernet
- Interlaken – up to 12 lanes (two cores can be combined to make a single 24 lane interface)
- DMA engine for PCIe

This section provides an overview of the capabilities of these IP blocks and their connectivity to the fabric for the user to interface to.

The following diagram shows the quantity and location of the IP blocks on the HD1000.

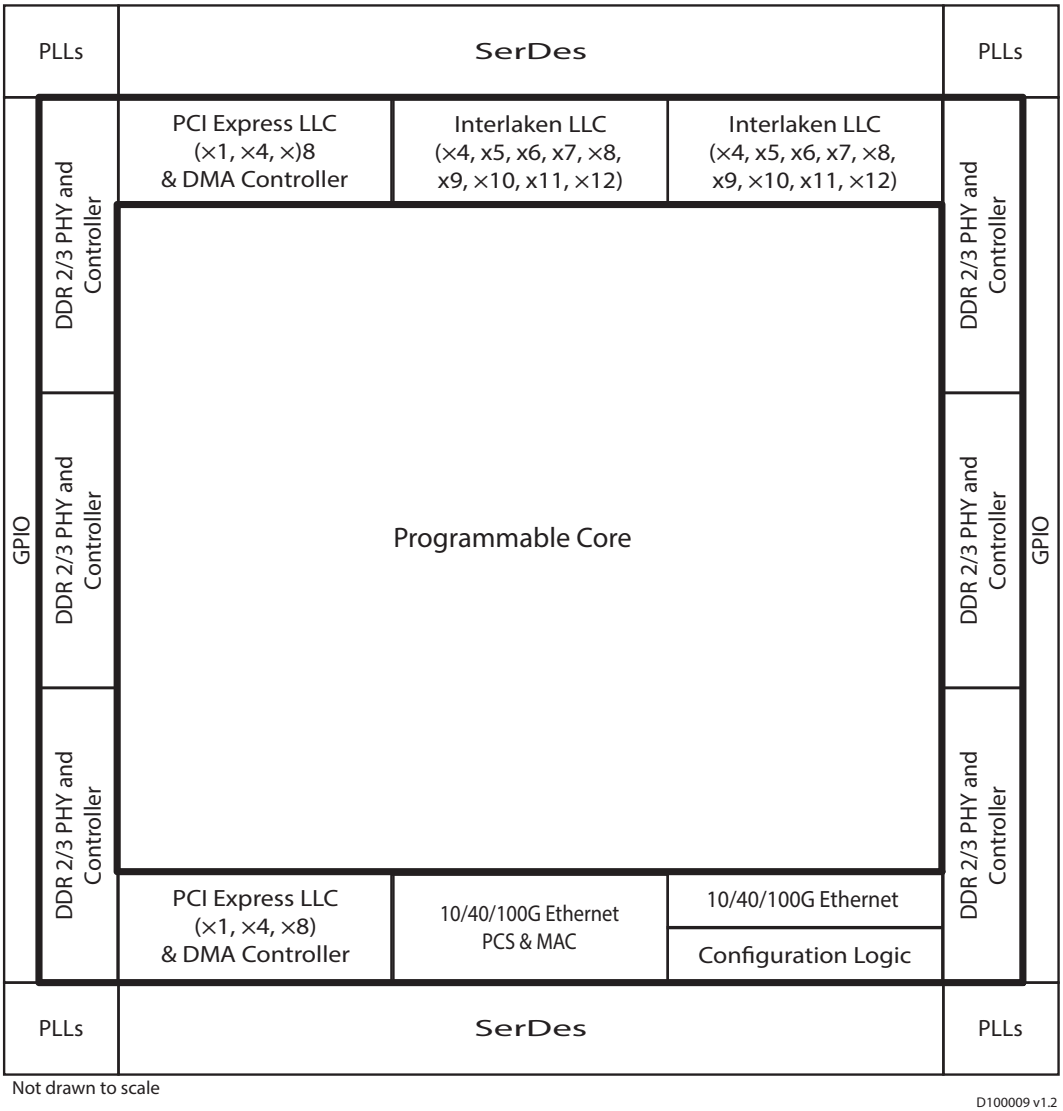


Figure 18: *Speedster Device Overview*

DDR Controller

Speedster22i HD devices contain up to 6 embedded DDR 2/3 controllers which can be used to interface with and control off-chip DDR3 memory devices or DIMMs. Each of the DDR3 controllers supports up to 72-bit wide data up to 1600 Mbps (800 MHz DDR).

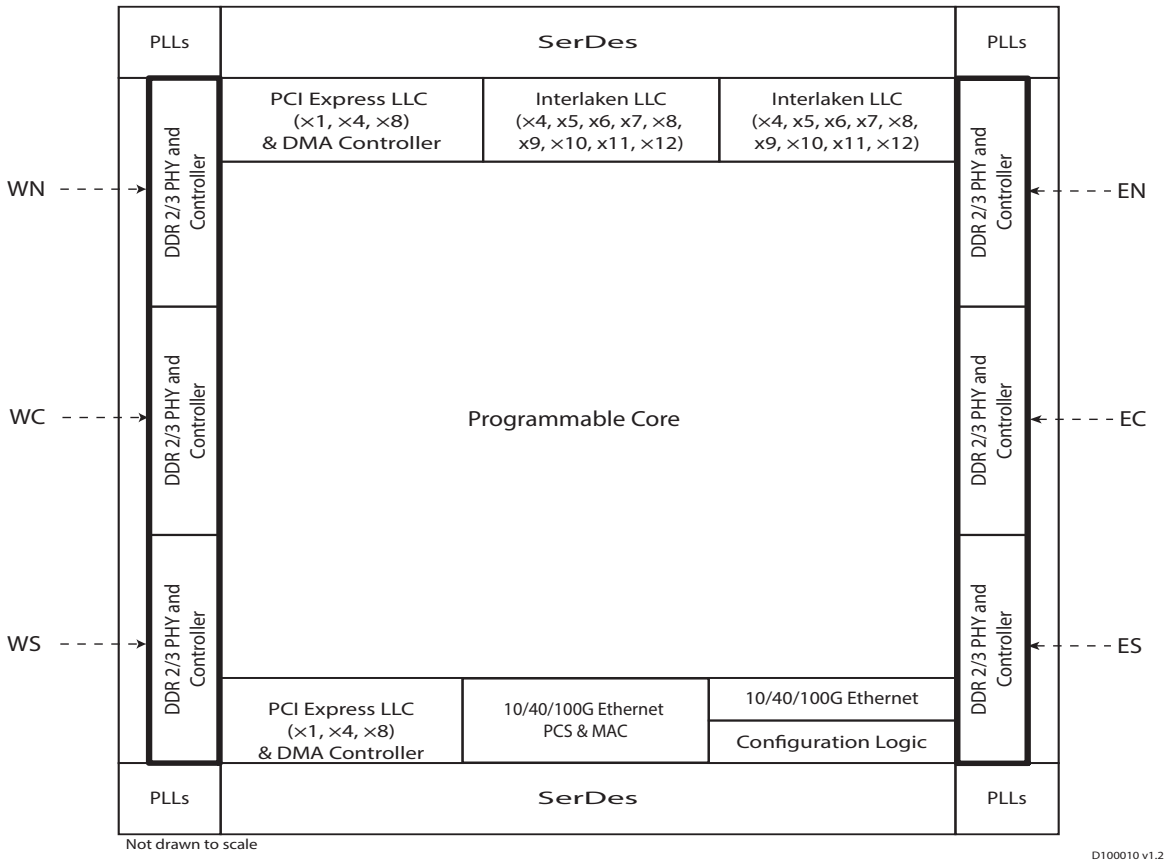


Figure 19: *Embedded DDR 2/3 Controller*

The DDR3 controller supports both "auto" and "custom" modes. When in the "auto" mode, functions such as (but not limited to) activating/precharging banks/rows, running calibration algorithms, and initialization sequences are handled transparently to the user (by the DDR core logic) in the embedded DDR controller. The mapping of byte lanes to pins is handled transparently by the embedded DDR PHY.

When in "custom" mode the user has the option to manually override functions such as automated refresh and initialization engines/sequences

Features:

- Up to 1600 Mbps data rate
  - The controller and PHY runs up to 800 MHz. To achieve 1600 Mbps data rate a 2× clock setting must be enabled, allowing the logic fabric to operate at half rate (400 MHz). The 2× clock setting can be enabled regardless of the data rate, allowing the interface to the fabric to run at half the rate of the hard IP controller
- 8:1 DQ:DQS ratio
  - The controller support 8 DQ signals for every DQS.
  - A 4:1 ratio can be used at the cost of half the available memory space
- 4 Chip selects (ranks) per controller
  - The external memory connected to each controller can comprise of up to 4 ranks (either 4 single-rank DIMMs or 2 dual-rank DIMM)
- Registered DIMM and unbuffered DIMM support
  - Each controller can independently support either rDIMMs or µDIMMs
- Address mirroring is supported. This feature is typically required for dual-rank µDIMMs

- Multi-burst mode
  - Each controller supports multi-burst mode, up to a burst length of 252 (DDR2) / 254 (DDR) / 248 (DDR3). This allows the embedded controller to automatically issue up to 252 cascaded read or write commands and automatically increment addresses based on a single command from the core fabric.
- Backwards-compatible
  - The embedded DDR controllers can support DDR3 (up to 1600 Mbps), DDR2 (up to 800 Mbps) and DDR protocols.
- Bypassable
  - If the user does not require all 6 DDR controllers, any (or all) can be bypassed to leverage use of the designated I/O for other purposes.
  - If the user does not require all 72 bits of the data bus, unused bits are available for general purpose I/O.
- Minimal LUT use
  - The DDR controllers are embedded, and as such do not use any of the LUTs in the Core Fabric
  - LUTs are required to drive the DDR Controllers; this driving logic is user-defined, and minimal in size.

DDR Control logic

Speedster22i HD devices contain six embedded (Hard IP) DDR Controller instances. Each instance is comprised of a DDR2/3 Controller and a DDR2/3 PHY, and is controlled using dedicated DDR Core logic.

The embedded DDR2/3 controller performs the following:

- All required initialization sequences such as the programming of AL and CL values based on user-defined parameters
- All required calibration algorithms including write levelization
- DQS enable (to control read-write turnaround of DQ/DQS bidirectional buses)
- DQS Delay (to skew the DQS by 90° relative to the corresponding DQ, such that the latter can be sampled in the middle of the bit transition)
- Translation of READ and WRITE requests received from the DDR driver into DDR protocol, i.e., RAS, CAS and WE.
- Translation between SDR and DDR
- Maintains integrity of memory contents by issuing periodic auto-refresh and zqcal commands
- Manages the activating and pre-charging of memory banks and rows, as required.
- Manages the driving of the memory address pins (with column or row information, as well as A10 function (precharge-all, auto-precharge, etc).
- Provides a data request signal to the DDR driver logic, some number of cycles after a corresponding write transaction request is received. This ensures that CAS latency, additive latency and burst length are all managed internally to the ACX DDR controller. It also provides early data request signal which can be used if more time is required to generate data.
- Provides data request signal and early data request signal if 2× mode is selected for 1600 Mbps.
- Providing a read data valid signal to accompany read data in response to a read request. This ensures that the round-trip latency to (and through) the memory is managed internally to the ACX DDR controller. It also provides early data valid signal which can be used to latch read data.
- Provides data request signal and early data valid signal if 2× clock mode is selected for 1600 Mbps.
- Provides signal to DDR driver logic to indicate that the DDR3 controller is busy and is not accepting new requests.

PCI Express

The Speedster22i PCI Express hard IP core implements all three layers (physical, data link, and transaction) defined by the PCI Express standard.

Key Features

- PCI Express Base Specification Revision 3.0 version 0.9 compliant; backward compatible with PCI Express 2.1/2.0/1.1/1.0a
- ×1, ×4 or ×8 PCI Express Lanes
- 8.0GT/s, 5.0 GT/s, and 2.5 GT/s line rate support
- Operates as Endpoint only
- PIPE-compatible PHY interface for easy connection to PIPE PHY
- Support for autonomous and software-controlled equalization
- Flexible equalization methods (algorithm, preset, user-table)
- Transaction layer bypass option
- Selectable data widths
  - 128-bit (all lane widths)
  - 256-bit (all lane widths – requires shim built using soft logic in fabric)
- Complete error-handling support
- Flexible core options allow for design complexity/feature trade-offs:
  - Advanced error reporting (AER) capability
  - End-to-end CRC (ECRC)
  - MSI-X and multi-vector MSI
- Supports Lane reversal, upconfigure, downconfigure, autonomous link width and speed
- Supports user expansion of configuration space
- Implements Type 0 configuration registers in Endpoint mode

Note: *When The PCIE interface's lane reversal mode is used, the reference clock to it's SerDes lane 0 must be provided even in ×1 and ×4 modes.*

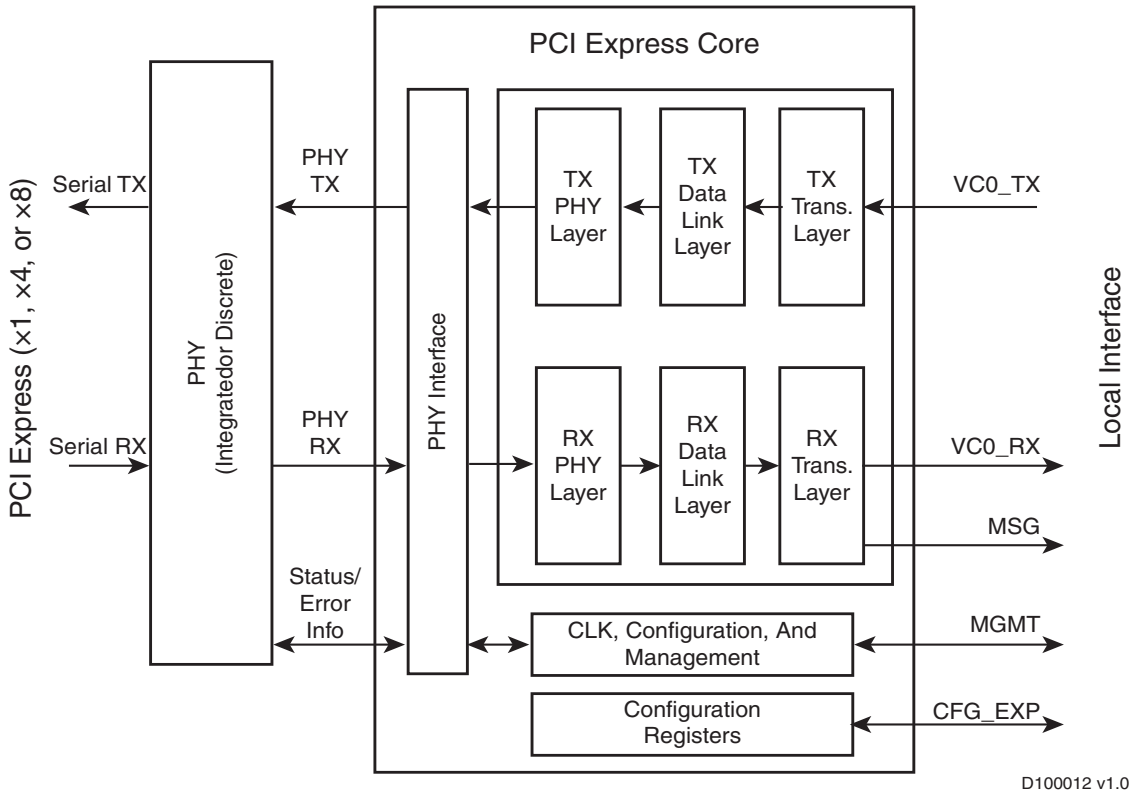


Figure 20: *PCI Express Block Diagram*

The PCI Express 3.0 Core implements all three layers defined by the PCI Express Specification: transaction, data link, and physical.

User-side interfaces include a Transmit Interface (VC0_TX), a receive interface (VC0_RX), a management interface (MGMT), a message interface

(MSG), and a Configuration register expansion interface (CFG_EXP).

The transmit and receive interfaces are intuitive packet-based interfaces that are used to transfer data between the PCI Express 3.0 Core and the user's application logic.

The table below shows the internal interface width/speed options available with the embedded PCIe controllers.

In addition, there is an Achronix provided shim for use to widen the bus for a lower bus speed.

Table 8: *PCI Express core interface width and speed*

Standard	PCIE Width	Internal Data Width and Speed
PCIe 3.x (8GT/s per lane)	×8	256 bits(1) at 250 MHz
		128 bits at 500 MHz
	×4	256 bits(1) at 125 MHz
		128 bits at 250 MHz
	×1	256 bits(1) at 31.25 MHz
		128 bits at 62.5 MHz
PCIe 2.x (5GT/s per lane)	×8	256 bits(1) at 125 MHz
		128 bits at 250 MHz
	×4	256 bits(1) at 62.5 MHz
		128 bits at 125 MHz
	×1	256 bits(1) at 15.625 MHz
		128 bits at 31.25 MHz
PCIe 1.x (2.5GT/s per lane)	×8	256 bits(1) at 62.5 MHz
		128 bits at 125 MHz
	×4	256 bits(1) at 31.25 MHz
		128 bits at 62.5 MHz
	×1	256 bit interface not required
		128 bits at 15.625 MHz

(1) *Requires shim built using soft logic in the fabric.*

Note: *PCI Express is configured as part of the entire FPGA configuration sequence. The FPGA configuration time is likely to exceed the maximum enumeration time allowed in the PCI Express specification.*

10/40/100G Ethernet MAC

The 10 / 40 / 100 Gigabit Ethernet MAC and PCS Core is designed to comply with the IEEE P802.3ba Specification Draft 2.2. The core can be used in either network interface card (NIC) or Ethernet Switching applications. A set of configuration registers is available to dynamically set the core to terminate and form MAC frames (NIC application) or to pass MAC frames without modification to the User application or to the Ethernet line (Switching application). When used in NIC or switching applications, the core provides support for IEEE managed objects, IETF MIB-II and RMON for management applications (e.g. SNMP).

The channelized MAC and PCS core can be configured to support either one of the following 5 configurations:

- 1-12 × 10 Gigabit Ethernet channels
- 1 × 100 Gigabit, 1-2 × 10 Gigabit Ethernet channels
- 1-3 × 40 Gigabit Ethernet channels
- 1-4 × 10 Gigabit, 1-2 × 40 Gigabit Ethernet channels
- 1-8 × 10 Gigabit, 1 × 40 Gigabit Ethernet channels

Figure 21 shows a high-level block diagram of the 10/40/100G Ethernet core and its main interfaces.

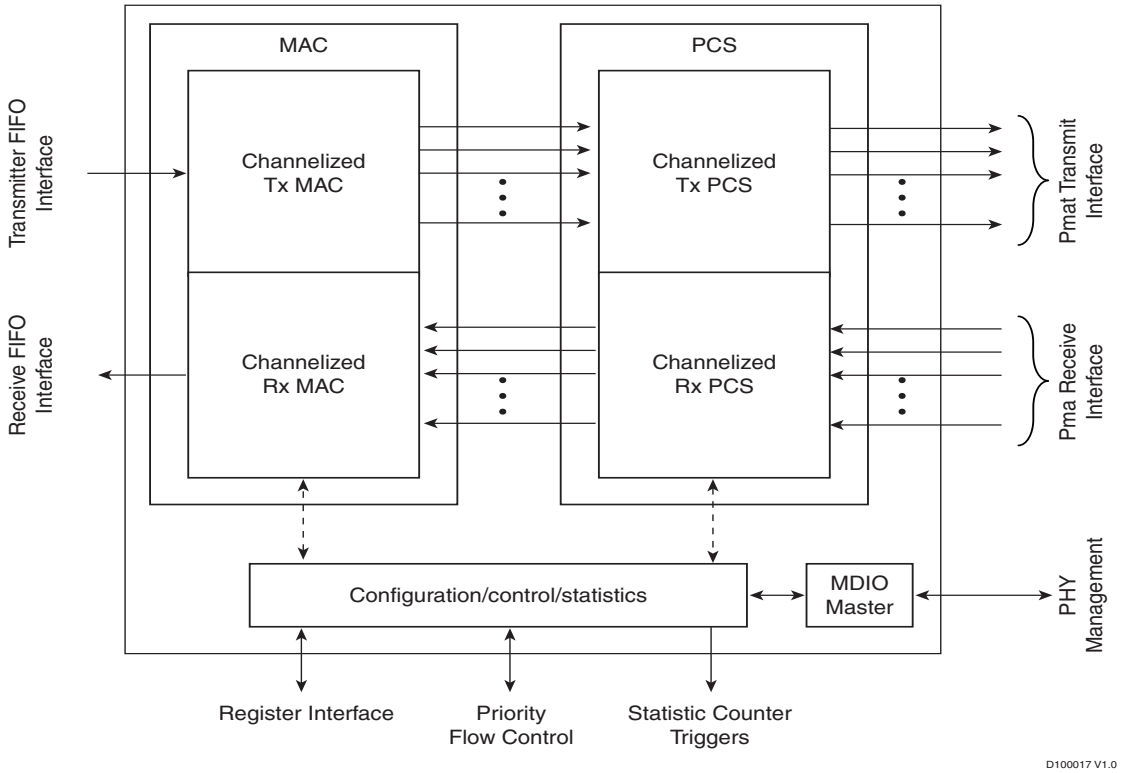


Figure 21: *The Embedded10/40/100G Ethernet Controller*

FIFO Interface

The 10/40/100 Gigabit Ethernet MAC and PCS Core implements a flexible FIFO interface that is connected to the internal FPGA fabric. This interface is decoupled, and therefore asynchronous to the Ethernet core. The transmit and receive FIFOs are also decoupled and therefore can operate at completely unrelated frequencies.

In order to allow for the start of frame to always be aligned on lane 0, the transmit and receive interface clocks have to run faster than the nominal required clock frequency (100 Gbps/384b = 260.42 MHz). In 100G mode of operation, worst case are 97-byte packets, which consume only 117 bytes (97-byte packet + 8-byte preamble + 12-byte IPG) on the Ethernet line, but require 3 × 48-byte words in the FIFO. Therefore, the minimum required transmit and receive interface clock speed is 144/117 × 260.42 MHz = 320.51 MHz.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a credit scheme. **Figure 22** shows the 10/40/100G Ethernet core FIFO interface.

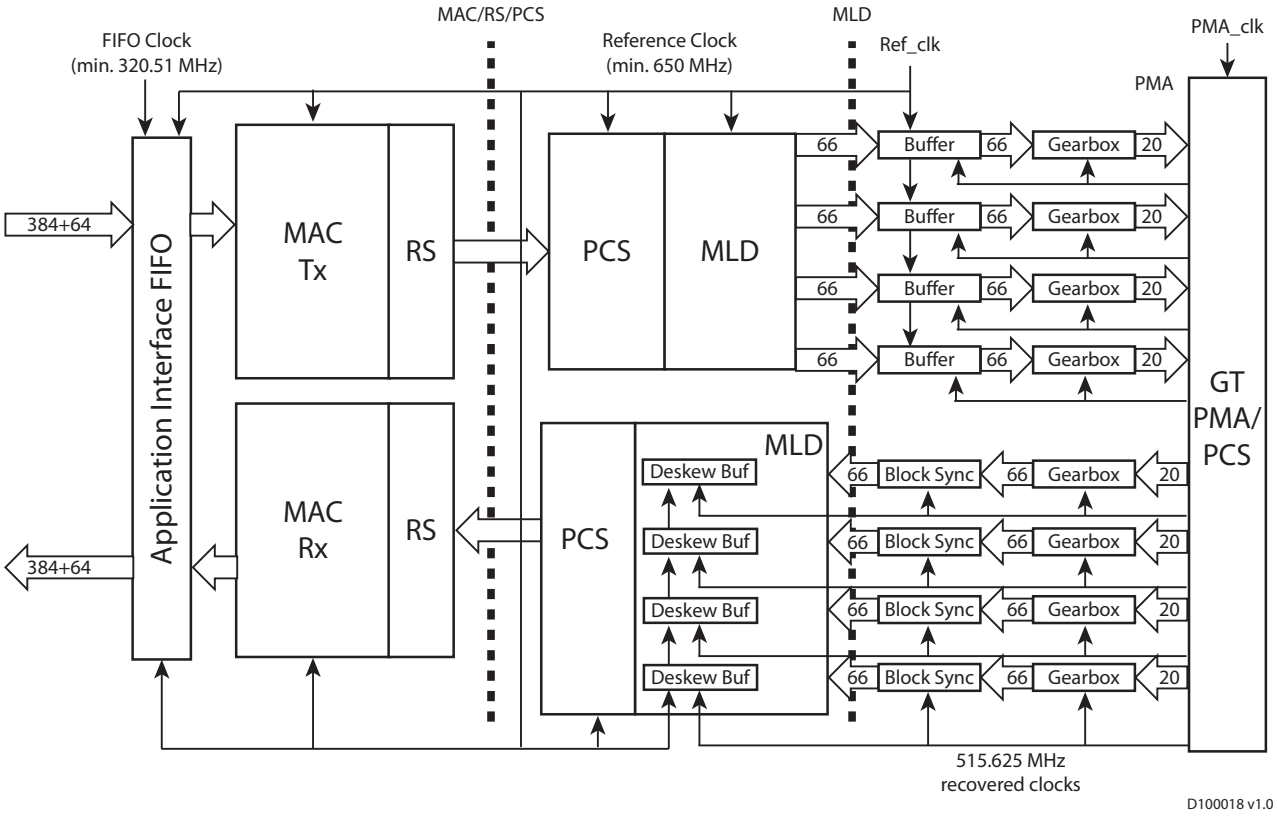


Figure 22: *The Embedded10/40/100G Ethernet FIFO Interface*

Interlaken

Interlaken is a scalable chip-to-chip interconnect protocol designed to enable transmission speeds from 10 Gbps to 100 Gbps and beyond. Using the latest SerDes technology and a flexible protocol layer, Interlaken minimizes the pin and power overhead of chip-to-chip interconnect and provides a scalable solution that can be used throughout an entire system.

In addition, Interlaken uses two levels of CRC checking and a self-synchronizing data scrambler to ensure data integrity and link robustness.

Speedster22i devices include a high-performance, low-power and flexible implementation of the Interlaken Protocol. The core is compliant with the Interlaken Protocol Definition, Revision 1.2, and offers a fast, turnkey Interlaken interface for chip-to-chip interconnect.

One of the benefits of Interlaken is its scalability and flexibility to accommodate different system designs.

Features

The Interlaken core has the following features:

- Support for data rates of 3.125 Gbps, 6.25 Gbps and 10.3125 Gbps (C2 only)
- Configurable lane width from ×4 up to ×12.
- Data striping and de-striping across the lanes
- Programmable BurstMax, BurstShort and Meta-FrameSize parameters
- 64/67 encoding and decoding
- Automatic word and lane alignment
- Self-synchronizing data scrambler
- Data bus width of 512 bits
- CRC24 generation and checking for burst data integrity
- CRC32 generation and checking for lane data integrity
- Data scrambling and disparity tracking to minimize baseline wander and maintain DC balance
- Support for all synchronization, scrambler state, diagnostic, and skip word block types
- Programmable rate limiting circuitry
- Robust error condition detection and recovery
- Channel-level and link-level flow control mechanism
- Support for 256 different logical channels
- Segment-mode and packet-mode transmission format
- Segment-mode and packet-mode receive format
- BurstMax size can be programmed up to 256 bytes
- Support for BurstShort requirement of 32 or 64 bytes
- In-band flow control
- Support for link-level flow control
- Flow control mechanism supports stopping packets in mid-stream – head of line blocking
- Rate matching with granularity of 1 Gbps
- Meta frame length programmable between 128 to 8K words
- Support for status messaging
- Lane decommissioning and resiliency

The block diagram of the Interlaken core is shown in **Figure 23**:

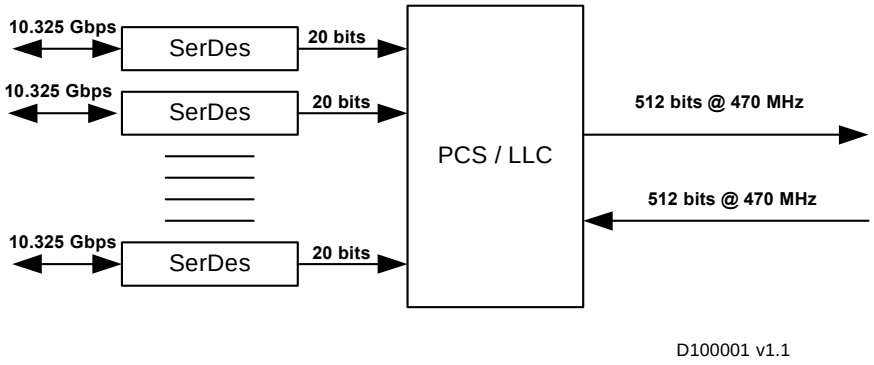


Figure 23: *Interlaken Block Diagram*

DMA Engine

Direct memory access (DMA) is the process of copying large amounts of data efficiently between two devices (typically the host 'system' memory and a bus device, or 'card') with minimal host processor involvement.

DMA requires a dedicated hardware resource — a DMA engine — to do the memory copy. The DMA engine's job is to do the copy operation specified by software. When using a DMA engine, the software only needs to implement the control function of the copy (tell the DMA engine where to copy from and to, etc.) rather than having to actually move the data itself to perform the copy.

There are two primary advantages of using DMA

- A DMA engine is much better at copying memory than software; DMA engines can issue large burst transactions (as large as the underlying hardware protocol allows) while the processor typically is only capable of very small burst transactions; the protocol efficiency, which is the ratio of payload_transferred / (payload_transferred + hardware_protocol_overhead), is typically extremely poor with the small payload size used by the processor and very good with the larger payload size used by a DMA Engine; a DMA Engine may produce 10 to 100 times greater throughput than a non-DMA software copy
- Software offloads the time-consuming copy task to the DMA engine and thus frees the processor for other tasks for which software is better suited. The copy operation is a repetitive task requiring only simple decisions and is well suited for hardware acceleration via DMA. Processor resources are better utilized on tasks which software is better suited for such as running the user's applications and processing (converting, parsing, displaying, etc.) the DMA data.

Features

The hard IP DMA Engine included in Speedster22i devices is directly attached to the PCI Express controller and thus to a host system — the PCIe core is typically a slave, not a master. The DMA engine uses the ARM AXI bus standard for connecting with the local card memory (DDR2/3) or other FPGA resources that will be part of a DMA transaction (Ethernet, Interlaken etc).

The DMA Engine block diagram is shown below in **Figure 24**.

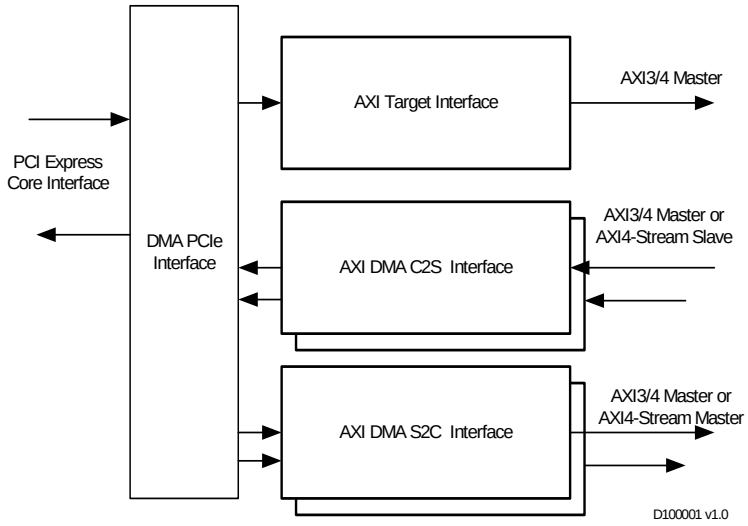


Figure 24: *DMA Engine Block Diagram*

The DMA engine consists of three main blocks

- AXI target interface. Used when the system host/master is outside of the Speedster22i FPGA and the host connects to the FPGA using PCIe. In this case the PCIe is acting as a slave and the DMA engine translates accesses received via the PCIe to an AXI master, which can in turn access other AXI slaves within the FPGA.
- AXI DMA C2S. This is the engine that controls card-to-system (C2S) transfers, i.e., data flow from the user logic in the FPGA fabric, to the PCIe Root

Complex via the FPGAs embedded PCIe endpoint controller.

- AXI DMA S2C. This is the engine that controls system-to-card (S2C) transfers, i.e., the data flow from the PCIe Root complex to the user logic in the FPGA.

The internal interfaces of the DMA block are fully compliant with the ARM AXI 3 and AXI 4 specifications.

Bus bridges to translate AXI transactions to the native format of the other Speedster22i hard IP are available from Achronix. This allows a system designer to connect all hard IP using the AXI bus protocol, simplifying the IP interconnection process.

Configuration Interface

The embedded programming and configuration logic is designed to support a variety of programming options. **Figure 25** outlines the basic block diagram of the programming and configuration logic, including additional logic to implement security features. The configuration management unit controls the startup and shutdown sequence from configuration mode to the user mode and back. The configuration management unit includes the provisions for configuring the device with a secure bitstream using a 256-bit advanced encryption standard (AES) algorithm in cipher block chaining (CBC) mode. The device contains a small non-volatile memory for the storage of the required AES key.

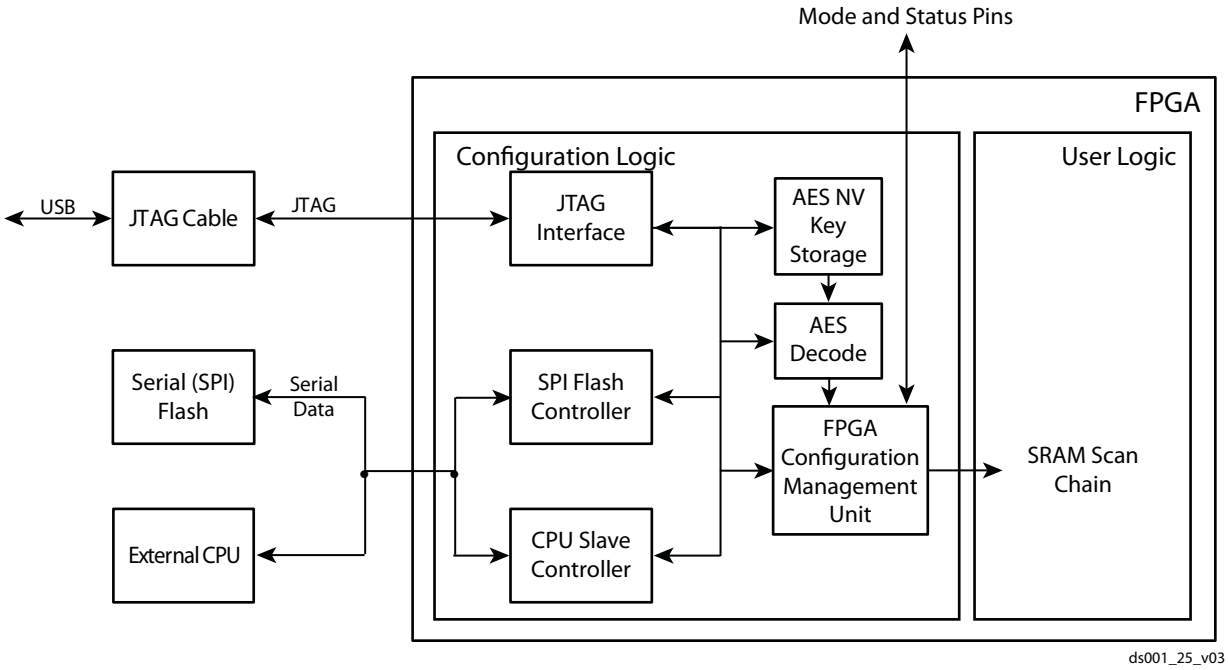


Figure 25: *Configuration Logic Overview*

Supported Programming and Configuration Modes

Several programming and configuration modes are used to support FPGA development and deployment phases. To avoid confusion, the term programming refers to the action of writing a bitstream to flash, so that on the next power-on cycle, the newly written bitstream can be used to configure the FPGA. The term configuration refers to the process of configuring the FPGA to implement the required user functionality.

Note: *The recommended memory space to store the configuration data for the Speedster22iHD is 128 MB.*

The supported programming modes are:

- Serial flash (SPI) programming (SFP)

The supported configuration modes are:

- JTAG FPGA configuration (JFC)
- Serial flash (SPI) FPGA configuration (SFC)
- External CPU FPGA configuration (EFC)
- Multiple serial flash (SPI) interfaces (MSF)

Note: *All flash modes listed are master (where an external clock is routed to the flash memory from the FPGA, controlling the configuration timing).*

A simplified diagram showing the supported configuration modes is shown in **Figure 26**, page 29.

JTAG FPGA Configuration (JFC)

The JFC mode allows the FPGA to be configured directly via a JTAG download cable. This mode is used during user-logic development and testing cycles.

Serial Flash Configuration (SFC)

The SFC mode allows serial flash PROMs to be used to configure the FPGA. In this mode the FPGA is the master, and therefore, supplies the clock to the PROM.

External CPU FPGA Configuration (EFC)

The EFC mode configures the FPGA from an external CPU after a system power-up cycle. This mode can be used both during user-logic development as well as in a production environment.

Multiple Serial Flash (SPI) Interfaces (MSF)

To reduce programming timing without adding the complexity of a parallel flash controller, a parallel array of four SPI flash devices can be used.

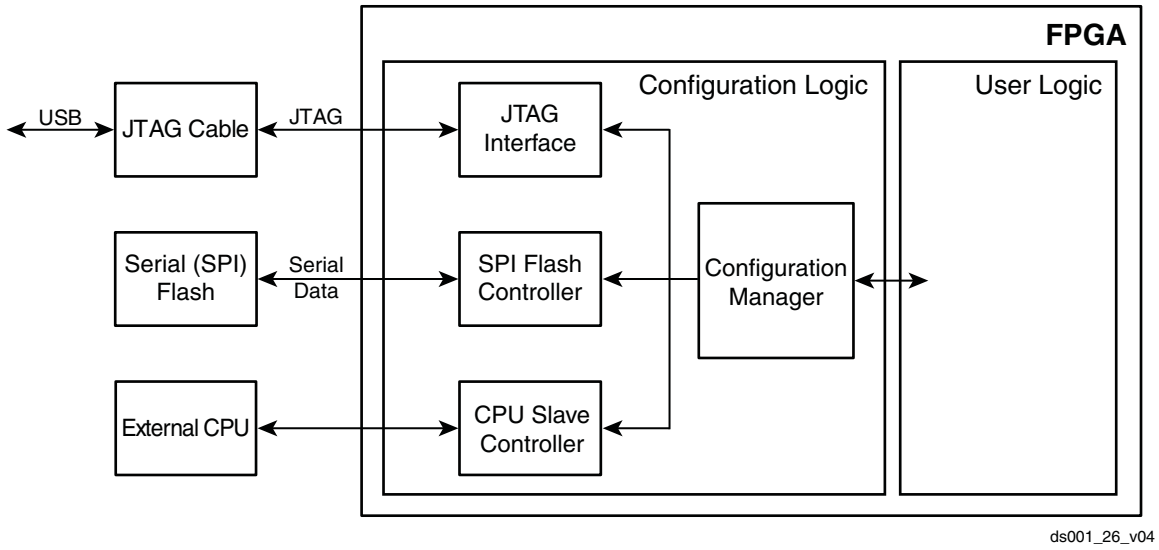


Figure 26: *Simplified Configuration Diagram*

Configuration Pin Descriptions

Table 9 *below* describes the FPGA pins used for the various supported configuration modes. Dedicated pins between modes can be shared (Serial Flash by 1, Serial Flash by 4, CPU modes) since the function is determined by the static state selected by the CONFIG_MODESEL[2:0] inputs. **Figure 27** through **Figure 30**, page 31 illustrate each of the supported configuration interfaces.

Table 9: *Pins Used for Support Configuration Modes*

External Pin Name	EFC	SFC	MSF	JFC
SDI	DQ[0]	Serial data output to flash memory		–
SDO[3]	DQ[1]	–	Input of configuration data from flash	–
SDO[2]	DQ[2]	–		–
SDO[1]	DQ[3]	–		–
SDO[0]	DQ[4]	Input of configuration data from flash		–
HOLDN	DQ[5]	Hold output to flash		–
CSN[3]	DQ[6]	–	Active-low chip select	–
CSN[2]	DQ[7]	–		–
CSN[1]	Unused	–		–
CSN[0]	Active-low chip select			–
CPU_CLK	CPU clock	–	–	–
CONFIG_RSTN	Active-low configuration reset			
CONFIG_DONE	Open-drain configuration done output			
CONFIG_STATUS	Open-drain SRAM initialization complete output			
CONFIG_MODESEL[2:0]	Configuration mode select; must be '100'	Configuration mode select; must be '001'	Configuration mode select; must be '010'	Configuration mode select; Not used in JFC but these pins should be '100', '001', '010' or '000'.
CONFIG_SYSCLK_BYPASS	Bypass configuration system clock. Tie to '0' or '1'	Bypass configuration system clock. Set to '0'		Bypass configuration system clock; Tie to '0' or '1'
CONFIG_CLKSEL	Selects configuration clock. set to '0'			Tie to '0' or '1'

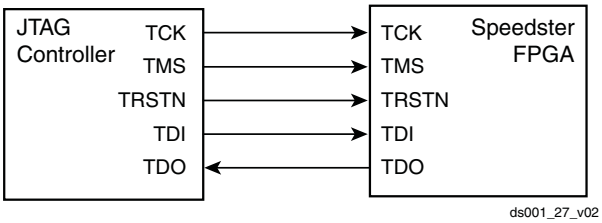


Figure 27: *JTAG Configuration Interface*

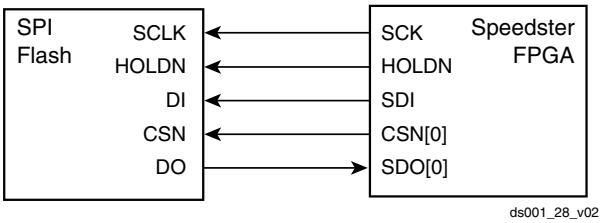


Figure 28: *SPI Flash PROM Configuration Interface*

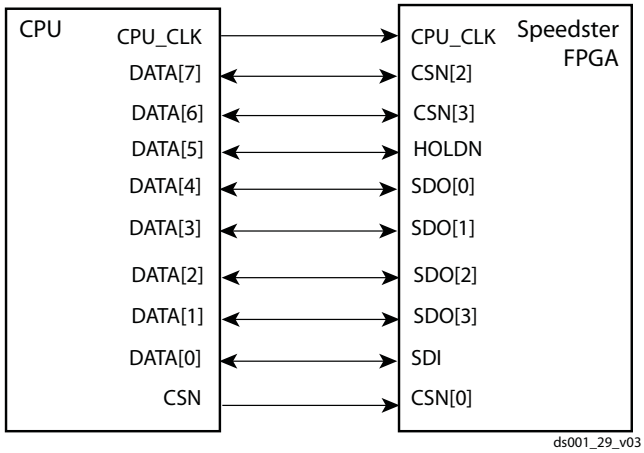


Figure 29: *Slave CPU Configuration Interface*

Figure 31, page 31 shows a simplified block diagram of FPGA configuration control logic. Regardless of the configuration control settings, if the programming hardware is connected to JTAG pins driving the FCU, the configuration clock (CFG_CLK) is automatically driven by TCK. In single or multiple flash configuration mode, if not overridden by JTAG circuitry, users can select between internally generated (SYSCLK) or externally driven configuration clock (CPU_CLK) using CONFIG_SYSCLK_BYPASS pin.

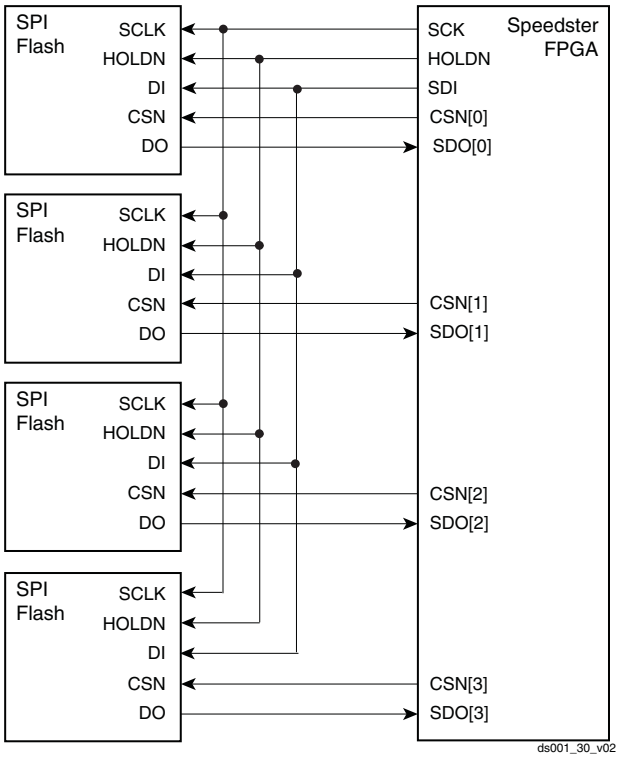


Figure 30: *By-Four SPI Flash Programming*

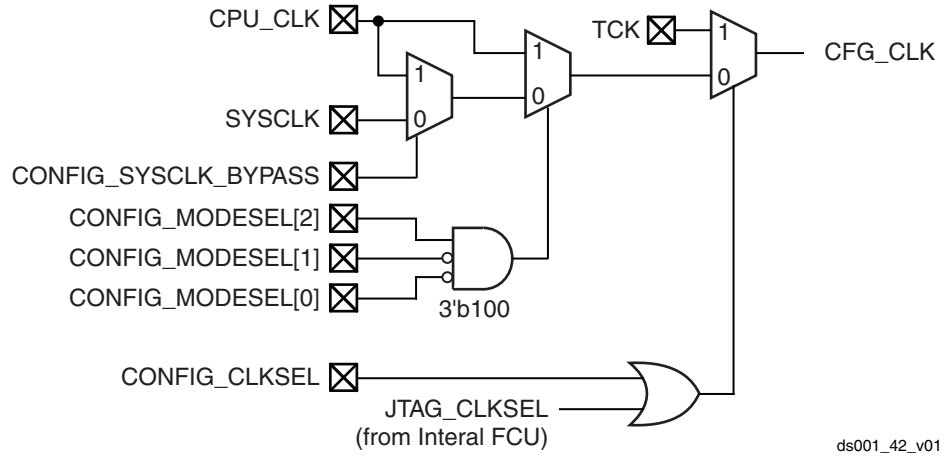


Figure 31: *FPGA Configuration Clock Selection Mux (Simplified View)*

10.3125 Gbps SerDes

Overview

All members of the Speedster22i family include embedded SerDes, which can be used to implement protocols running at between 1.0625 Gbps and 10.3125 Gbps per lane. Each SerDes can be used for chip-to-chip or backplane signaling.

Block diagrams of the transmit and receive sections respectively are shown in **Figure 32** and **Figure 33**.

As shown in **Figure 32** below, the transmit datapath consists of both the physical media access (PMA) and the physical coding sublayer (PCS). The PMA handles the low-level data signaling, functions, while the PCS handles the interface to protocol controllers as well as any data encoding which may be required.

Note: *Whenever a SerDes lane is used, it's reference clock MUST be running prior to the FPGA being configured. Before, during and after configuration, the reference clock must never be stopped or varied in any way. Whenever an SBUS interface of any SerDes lane is used then the clock of that interface MUST be running prior to the FPGA being configured. Before, during and after configuration, the SBUS clock must never be stopped or varied in any way. The SBUS clock must be less than 100 MHz and must be less than the SerDes reference clock frequency.*

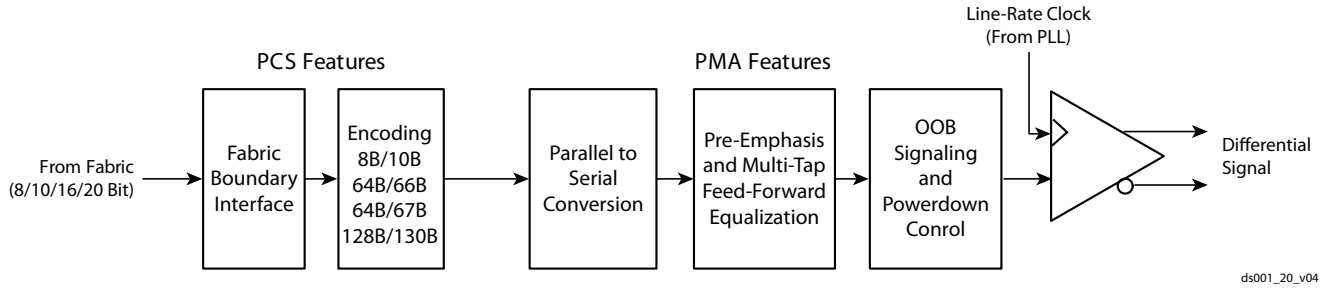


Figure 32: *10.3125G SerDes – Transmit Section*

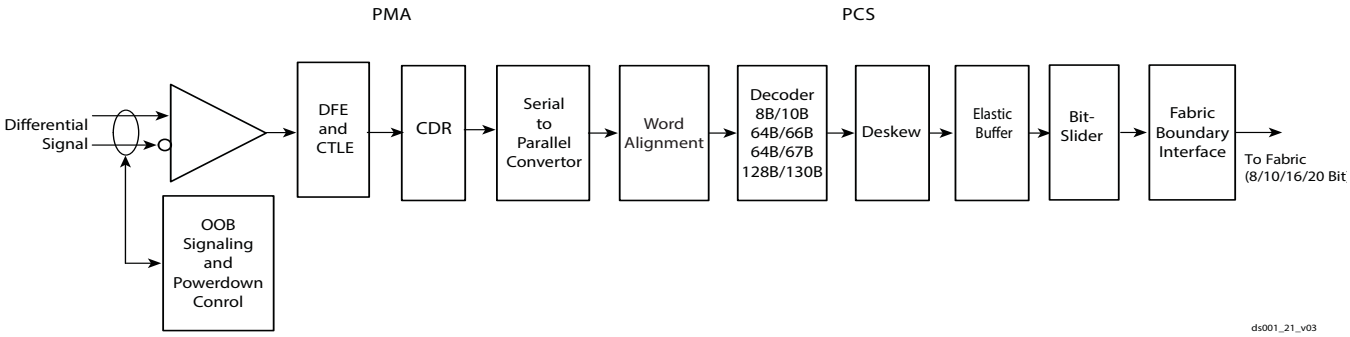


Figure 33: *10.3125G SerDes – Receive Section*

PMA Features

The PMA has the following feature set:

- Operating bitrate of 1.0625 Gbps, 10.3125 Gbps per lane
- DC coupling or external AC coupling
- Lock to reference clock or lock to data
- Data oversampling for capture of un-encoded as well as slower speed traffic
- CTLE with up to 20 dB of gain at 4 key data rates
- 5-tap adaptive DFE for high-speed operation over lossy channels
- On-chip scope in the receive channel for measuring eye (width and height) and BER
- 4-tap transmit de-emphasis
- Dedicated transmit PLL for every SerDes lane providing maximum bitrate flexibility
- Supports loop timing for re-timer applications
- Built-in self-test (BIST) withnear/far end loopback, PRBS 7, 15, 23 and 31 generation/checking and 40-bit user-defined pattern

PCS Features

The PCS supports the following functions:

- Support for 8B/10B, 64B/66B, 64B/67B and 128B/130B encoding and decoding
- Symbol alignment
- Clock compensation
- Lane-to-lane de-skew
- Manual bit-wise de-skew
- Near and far end loopback
- Clocking and reset for interface to programmable fabric

Control Plane Features

Programmable interface for status monitoring and dynamic configuring of SerDes operation

Transmit Datapath

On the transmit data path the PCS block receives data from the fabric and delivers it to the PMA layer. The encoder can be in the path or can be bypassed.

The transmit data path consists of the following functions:

- Polarity and bit reversal
- 8b/10b encoder
- 64b/66b encoder
- 64b/67b encoder
- 128b/130b encoder

Polarity and bit reversal

This function can be used to optionally change the polarity or bit-ordering of the transmit data. There are two instances of the polarity and bit reversal function that can be independently controlled.

Encoders

The encoder generates encoded words from un-encoded data at the PCS input. The encoders can support a single 8-bit or a double 16-bit word, with the encoders cascaded for the 16-bit case. The encoder output is passed to the PMA for serialization.

8b/10b encoder

The 8b/10b encoder function includes extra logic to fully support Gigabit Ethernet, XAUI and PCIe (gen1, gen2) protocols.

64b/66b encoder

The 64b/66b encoder function includes extra logic to fully support 10/40/100 GbE and other ≥ 10Gbps protocols.

64b/67b encoder

The 64b/67b encoder function supports Interlaken.

128b/130b encoder

The 128b/130b encoder is specifically to support the PCIe gen3 protocol. The function is compliant to the PIPE specification and interoperates with the PCIe embedded IP to provide gen3 support.

Other protocols that require a generic 8b/10b or 128b/130b encoder can also make use of this functional block.

BIST pattern generation

The transmit datapath will include pattern generation logic to be used in test mode for built-in self test. The patterns supported are:

- PRBS7
- PRBS15
- PRBS23
- PRBS31
- User defined pattern (up to 40-bits that will be repeated continuously).

Receive Datapath

On the receive datapath the PCS receives data from the PMA and transfers it to the fabric interface.

A byte level de-serializer can be implemented at the fabric interface to relax clock timing requirements.

The receive datapath consists of the following functions:

- Phase picking logic
- Polarity and bit reversal
- Symbol alignment
- 128b/130b block synchronization and decode
- Lane to lane de-skew (up to 12 channels)
- Clock compensation
- 8b/10b decoder
- Transition density checker
- Bit slider
- Symbol-slip mode allows data to be advanced or delayed by 1 cycle in addition to general increment of 1 cycle

Phase Picking Logic (PPL)

The phase picking logic (PPL) function is meant to be used in conjunction with the CDR over-sampling mode of operation in the PMA. The PPL block accepts the oversampled data from the PMA and implements an algorithm to extract the receive data bits. The PPL block also provides feedback on the phase change rate to the PMA. This can be used to account for any frequency offset between the receive data on the line and the over-sampling clock. The following configurations are supported for over-sampling.

- 4× over-sampling, up to 6.25 Gbps
- 8× over-sampling, up to 3.125 Gbps

Polarity and Bit Reversal

There are two independently configurable polarity and bit reversal blocks in the receive datapath. There is also an option to invert the receive data from the PMA dynamically.

Symbol Alignment

The parallel data output from the SerDes PMA needs to be aligned to byte boundaries before it can be used by downstream logic. The transmit side typically sends unique symbols that can be used for alignment. The symbol alignment block looks for these symbols and sets the byte boundary. The symbol alignment block supports alignment to 2 different symbols and can also detect a 4-symbol sequence match.

- Manual mode
  - Aligns to a pre-defined symbol for each request from the fabric.
- Bit slip mode
  - Slips 1-bit of data for each request from the fabric.
- Automatic alignment
  - Configurable state machine to automatically align to a pre-defined symbol, including optional hysteresis to determine loss of alignment.

128b/130b Decode

The 128b/130b decode block is to support PCIe Gen3 protocol. It is compliant to the PIPE specification.

Lane to Lane De-skew

The PCS includes support for lane to lane de-skew across the SerDes lanes on one side of the chip (up to 12 lanes). De-skew across lanes on different sides of the chip is not supported. The de-skew function supports up to 2 different de-skew patterns (up to 5-symbols long each). The de-skew function supports the following modes:

- Automatic de-skew
- Manual de-skew
- Symbol slip. Data in a given lane is advanced or delayed by 1-2 cycles.

Clock Compensation

The receive data from the PMA is timed to the recovered clock that is output from the CDR (except in CDR over-sampling mode). Some protocols need this data to be synchronized to an internal system clock (typically same as the transmit clock). The clock compensation function uses an elastic buffer to compensate for any frequency offset between the recovered clock and the internal system clock. This function uses a configurable skip or pad symbol that can be dropped or added at specific instances in the receive data stream to compensate for any frequency off-set.

8b/10b Decoder

The 8b/10b decoder generates 8-bit code groups and 1-bit control from 10-bit encoded data. It uses the code group mapping specified in IEEE 802.3 clause 36. If the fabric interface is 16-bit data path, then 2 8b/10b

decoders are cascaded to produce a 16-bit width to the fabric.

The 8b/10b decoder includes error indication logic to provide status of code word or running disparity errors to downstream logic.

Bit Slider

The bit slider is an 80-bit barrel shifter that can be used to control bit-wise alignment from the fabric. This feature can be used to implement any user specific algorithm for bit-level alignment or de-skew across multiple lanes. It can be used in conjunction with the symbol slip mode of the de-skew function to achieve a wide range of de-skew. If used stand alone this function can provide up to 30 UI of de-skew.

Transition Density Checker

This function monitors the receive parallel data from the PMA and keeps track of consecutive 0s or 1s. If the number of consecutive 0s or 1s exceeds a pre-configured threshold it sends a status indication to the downstream logic.

Control Plane Interface

The PCS and PMA blocks have a parallel bus control plane interface to allow user logic to configure the data path and also to monitor the status of key blocks. The control plane interface is based on a simple request, acknowledge handshake protocol with an 8-bit data bus read, write data bus and 16-bit address bus. To minimize pin usage on the fabric side, a serialized version of the control plane interface is supported for access from the fabric. A serial to parallel converter, parallel to serial converter is implemented in the logic core to interface the fabric to the PMA and PCS control plane. Each SerDes lane has its own control plane interface.

Support for Standard Protocols

Table 10 shows which standard protocols are supported. Any listed PCS feature can be bypassed if not needed.

Table 10: *10.3125Gbps SerDes Supported Standards*

Standard	General		PCS						PMA	
	Number of Lanes	Gbps (Per Lane)	Internal Bus Width (Per Lane)	Coding	Word Alignment	Lane Alignment	Elastic Buffer	Spread Spectrum Clocking	Power-Down States	OOB Signaling
PCIe 1.1	1 / 4 / 8	2.5	8 / 16	8B / 10B	K28.5	✓	✓	✓	✓	✓
PCIe 2.0	1 / 4 / 8	2.5 / 5.0	16	8B / 10B	K28.5	✓	✓	✓	✓	✓
PCIe 3.0	1 / 4 / 8	2.5 / 5.0 /8.0	16	128B / 130B	K28.5	✓	✓	✓	✓	✓
SRIO	1 / 4 / 8	1.25/2.5/ 3.125/6.25	16	8B / 10B	K28.5	✓	✓	✓	✓	✓
Gigabit Ethernet	1	1.25	8	8B / 10B	K28.5		✓			
10 Gigabit Ethernet (XAUI)	4	3.125	8	8B / 10B	K28.5	✓	✓			
10 Gigabit Ethernet (XFI)	1	10.3125	8	64B /66B	sync header					
40 Gigabit Ethernet (XLAUI)	4	10.3125	8	64B /66B	sync header					
100 Gigabit Ethernet (CAUI)	10	10.3125	8	64B /66B	sync header					
SGMII	1	1.25	8	8B / 10B	K28.5		✓			

Table 10: *10.3125Gbps SerDes Supported Standards (Continued)*

Standard	General		PCS						PMA	
	Number of Lanes	Gbps (Per Lane)	Internal Bus Width (Per Lane)	Coding	Word Alignment	Lane Alignment	Elastic Buffer	Spread Spectrum Clocking	Power-Down States	OOB Signaling
Fibre Channel – 1	1	1.0625	8	8B / 10B	K28.5		✓			
Fibre Channel – 2	1	2.125	8	8B / 10B	K28.5		✓			
Fibre Channel – 4	1	4.25	16	8B / 10B	K28.5		✓			
Fibre Channel – 8	1	8.5	16	8B / 10B	K28.5		✓			
Fibre Channel – 10	1	10.52	16	8B / 10B	K28.5		✓			
SATA Gen 1	1	1.5	10	8B / 10B	K28.5			✓	✓	✓
SATA Gen 2	1	3	10	8B / 10B	K28.5			✓	✓	✓
SATA Gen 3	1	6	10	8B / 10B	K28.5			✓	✓	✓
SAS	1	1.5 / 3.0 / 6.0	10	8B / 10B	K28.5			✓	✓	
Interlaken	6, 8, 10, 12, 24	4.6 – 10.3125	20	64B / 67B	sync header	(2)				
CEI6 – SR	1	4.976 – 6.375	16							
CEI6 – LR	1	4.976 – 6.375	16							
CEI11 – SR	1	9.95 – 11.1	16							
SPI–5	19	3.125	8		(2)	(2)				
SFI–5.1 (3)	19	2.488 – 3.125	8		(2)	(2)				
SFI–5.2 (3)	19	9.95 – 11.1	8		(2)	(2)				
SFI–S (3)	19	9.95 – 11.1	8		(2)	(2)				
Infiniband	1	2.5 / 5.0 / 10.0	8		(2)	(2)				
OC48[(1)]	1	2.488	8		(2)					
GPON	1	1.25 / 5.0 / 10.0	8		(2)					
EPON	1	1.25 / 5.0 / 10.0	8		(2)					
CPRI	1	1.228 / 2,456 . 3.072 / 6.144	8		(2)					
OBSAI	1	1.536 / 2.456 / 6.144	8		(2)					

Table 10: *10.3125Gbps SerDes Supported Standards (Continued)*

Standard	General		PCS						PMA	
	Number of Lanes	Gbps (Per Lane)	Internal Bus Width (Per Lane)	Coding	Word Alignment	Lane Alignment	Elastic Buffer	Spread Spectrum Clocking	Power-Down States	OOB Signaling
Backplane Inter-connect (with DFE)	1–20	1.25 – 10.3125	8/16							

Notes:

1. Not supported by integrated PCS block (must be implemented in fabric).
2. Optional.
3. Bit slider is available for SFI Protocols

Programmable I/O

I/O Types

Speedster22i device I/O come in five categories, as shown in **Table 11**.

Table 11: *I/O Types*

Full Name	Description	Count
Programmable I/O – Extended Feature	• User programmable • Complies with wide range of I/O standards • Low to medium bit rates • DLLs for input and output delay adjustments as required for advanced memory and datapath interfaces	12 × N where N = byte-lane count (device/package dependent)
SerDes I/O	• User programmable • Complies with wide range of SerDes-based standards • High bit rates	For a pair of lanes: • 2 × 4 pins for data (receive/transmit) • 1 × 2 pins for ref clocks (shared between lanes) Effectively 5× no of lanes.
Clock I/O	These I/O have connectivity to the **"Global Clock Network"** in the corresponding corner.	24
Dedicated I/O	Reserved for device configuration and test	39
Power / Ground	Core power; I/O power; ground	Device/package dependent

Supported Standards

Each programmable I/O can be configured to conform to any of a large number of I/O standards, both single-ended and differential, as summarized in **Table 12**, page 39. Each I/O can operate as an input, an output, or a bidirectional I/O (**Figure 34**). Of course, a differential signal consumes two I/O, whereas a single-ended signal consumes only one.

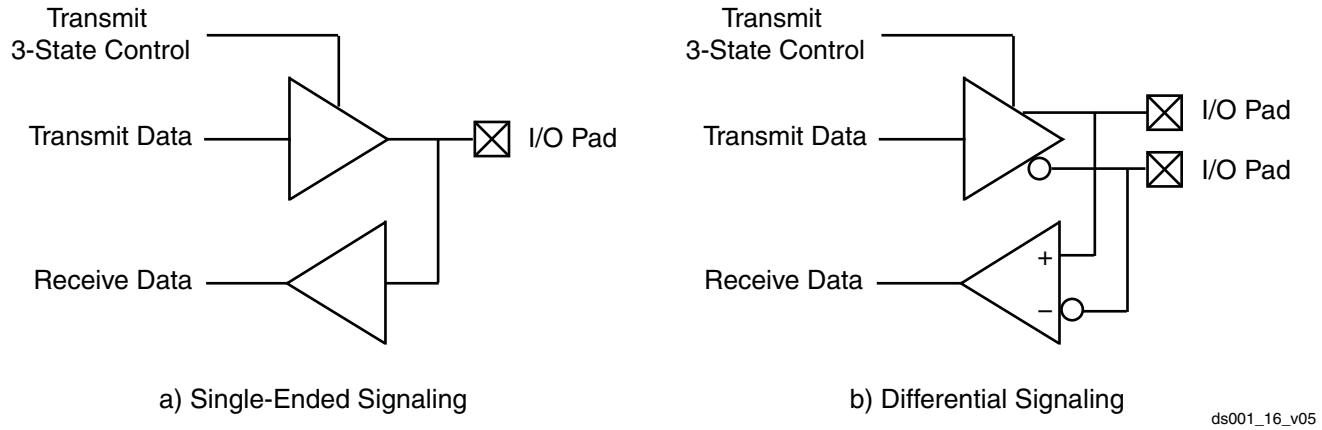


Figure 34: *Programmable I/O*

Table 12: *Programmable I/O: Supported Standards*

Type		Volts	Class	Interface Applications and Standards		
				Standard	Max Clock Rate (MHz)	Max Data Rate (Mbps)
Single-Ended	LVCMOS	1.8V	–	General Purpose	300	300
	LVCMOS	1.5V	–		300	300
	LVCMOS	1.2V	–		300	300
	HSTL	1.8V	Class I	QDR II SRAM / RLDRAM II	533	1066
	HSTL	1.8V	Class II		533	1066
	HSTL	1.5V	Class I	Memory and Switch Fabric	533	1066
	HSTL	1.5V	Class II		533	1066
	SSTL	1.8V	Class I	DDR2 SDRAM	533	1066
	SSTL	1.8V	Class II		533	1066
	SSTL	1.5V	Class I	DDR3 SDRAM	800	1600(1)
	SSTL	1.5V	Class II		800	1600(1)
	POD	1.8V	–		400	800
	POD	1.5V	–		400	800
	LVDS	1.8V	–	SPI4.2, SFI4.1	800	1600
Differential	Differential HSTL	1.8V	Class I	QDRII SRAM / RLDRAM II	533	1066
	Differential HSTL	1.8V	Class II		533	1066
	Differential HSTL	1.5V	Class I	Memory and Switch Fabric	533	1066
	Differential HSTL	1.5V	Class II		533	1066
	Differential SSTL	1.8V	Class I	DDR2 SDRAM	533	1066
	Differential SSTL	1.8V	Class II		533	1066
	Differential SSTL	1.5V	Class I	DDR3 SDRAM	800	1600(1)
	Differential SSTL	1.5V	Class II		800	1600(1)

Notes:

1. Maximum data rate is speed-grade dependent.

Grouping: Banks and Byte Lanes

Programmable I/O are deployed in banks. All I/O within a bank must share:

- The same V_{DDO}
- The same V_{REF}
- The same R_{REF}s (used for controlled-impedance I/O)

The following sections describe byte-lane and I/O banks.

Byte-Lane

I/O banks consist of byte-lanes. Each byte-lane consists of following:

- Twelve I/O buffers consisting of twelve bit-modules. Each pair of two I/O buffers can be configured as two single-ended I/O buffers or as one differential I/O buffer.
- One master DLL and twelve slave delay elements – There is one slave delay element for each of the twelve bit-modules.
- Appropriate logical/physical structures for DQS preamble enable and postamble shut-off for DDR2/DDR3 SDRAM application.
- Write-pointer and read-pointer logic for a mesochronous synchronizer.
- Muxes and buffers for source synchronous clock networks
- Two I/O buffers in each byte-lane are capable of receiving clocks from external devices. They can be

used as one differential pair (e.g., to receive differential DQS for DDR2) or two single-ended buffers (e.g. to receive clock CQ and CQn for QDR2). The other 10 I/O buffers can be used as Data pins to be latched by the received clocks.

- Logic in each bit-module to support SDR, DDR and QDR support for the input and output directions, plus output enable logic.
- V_{REF} is integrated in the byte-lane.

I/O Banks

I/O buffers in Speedster22i have been grouped into multiple banks. There are three type of I/O banks — EF bank, clock bank and configuration bank.

Enhanced Function (EF) Bank

An EF bank consists of the following:

- Four or five byte-lanes (each byte-lane is 12 I/O, which would give a total of 48 or 60 I/O depending on the bank).
- Common PVT calibration control and calibration resistors for driver impedance.
- Common PVT calibration control and calibration resistors for on-die parallel termination and on-die differential termination impedance.
- All I/O buffers use a common I/O voltage V_{DDO}.
- All I/O buffers use a common reference voltage V_{REF}.

There are three EF banks corresponding to each DDR2/3 controller: two sets with four byte-lanes and one set with five byte-lanes, giving a total of 13 byte-lanes or 156 I/O. Half of the total EF banks are on the right side of the device, and the other half are on the left.

Clock Bank

A clock bank consists of following:

- Six I/O buffers for six single-ended clocks or three differential clocks or any combination of single-ended and differential clocks.
- All six I/O buffers use a common I/O voltage V_{DDO}.
- All six I/O buffers use a common reference voltage V_{REF}.
- Unlike the EF banks, the clock bank does not have PVT calibration control.

There are a total of 4 clock banks, one in each corner for a total of 24 clock input pins. As the name suggests clock banks are for clock interfaces.

Configuration Bank

The configuration bank is a special dedicated I/O bank which sits on the bottom edge of Speedster22i. I/O buffers in this bank are used for JTAG and configuration interfaces. The total number of I/O buffers in this bank is 36.

Figure 35 shows a conceptual floorplan of a device with ten banks. Clearly, the number of banks varies from device to device.

Differential operation

When used as differential pair such as LVDS, even-numbered pins are positive and odd-numbered (+1 from the positive) pins the negative, e.g., PAD_EN_-BYTEIO0DQ2 is positive and PAD_EN_BYTEIO0DQ3 is the corresponding negative. For pins labeled *DQS and *DQSn, DQS is positive and DQSn is negative.

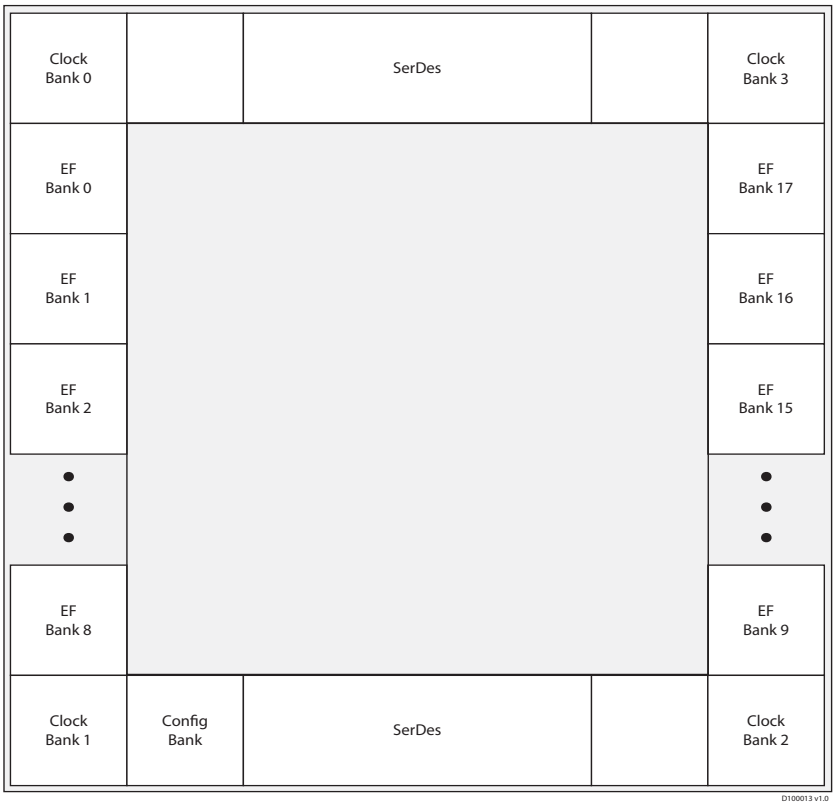


Figure 35: *I/O Banks in an HD1000 Device*

I/O Bank Resources

Each I/O buffer has a bit-module associated with it. the bit-module consists of appropriate logic blocks to facilitate I/O buffer interface with the FPGA core. The logic in a bit-module consist of following blocks.

Receive Data Path

The receive data-path (**Figure 36**) has structures to interface incoming data from the I/O buffer to the fabric. The incoming data can optionally pass through a delay element before reaching any logic in the data-path. The bit-module in the receive direction supports three modes:

- Combinatorial to the core – In this mode the data coming from the input buffer is directly forwarded to the FPGA core.
- SDR – In this mode the data coming from the input buffer gets registered in the bit-module before getting forwarded to the FPGA core.
- DDR – In this mode DDR data is received, and for each bit, two-bit data is forwarded to the fabric. This mode has two versions:
  - The two bits are forwarded to the fabric core on different clock-edges
  - Both bits are forwarded to the fabric core at the same clock edge.

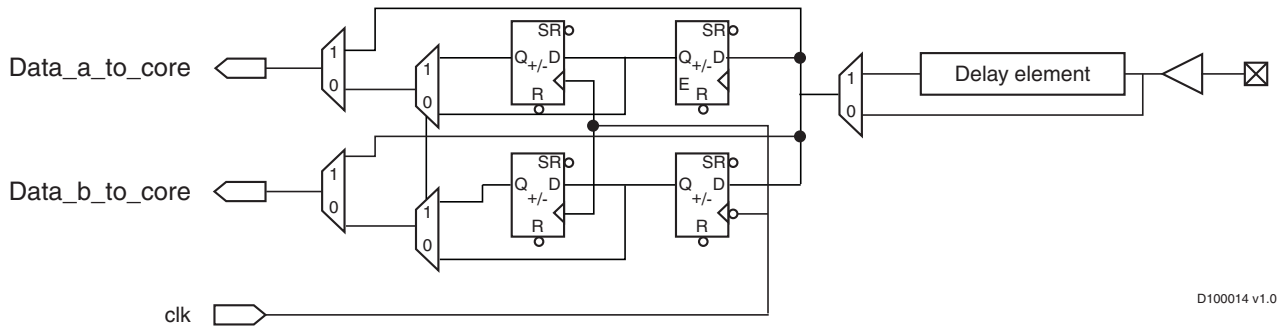


Figure 36: *Recieve Data path*

Transmit Data Path

The transmit data-path has structures to interface data from the fabric to the I/O buffer. The bit-module in the transmit direction supports three modes:

- Combinatorial – In this mode the data coming from FPGA core is directly forwarded to the output buffer.
- SDR – In this mode the data coming from FPGA core gets registered in the Bit-Module before getting forwarded to the output buffer.
- DDR – In this mode two bits of data coming from the fabric gets converted to DDR data before being forwarded to output buffer.

In each of the above three modes, the transmit data-path can optionally include a delay element.

Figure 37 below shows the bit-module's transmit data-path.

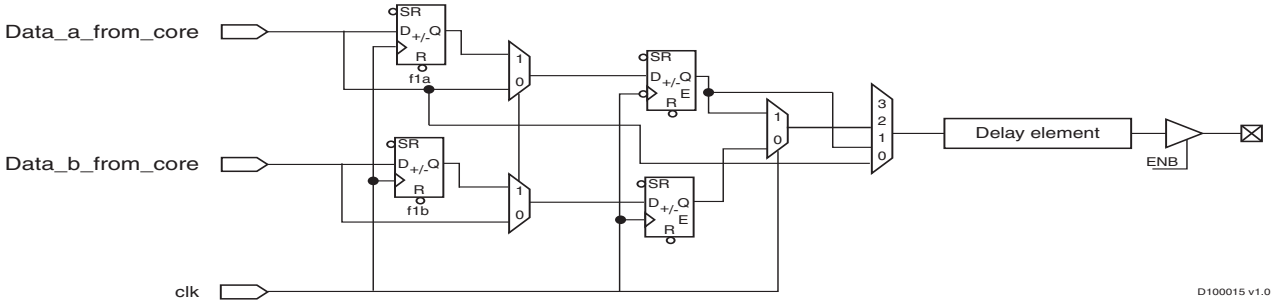


Figure 37: *Transmit Data path*

All the flops shown above can be configured to be either positive or negative edge triggered. All the flops support an asynchronous or a synchronous reset. They also have a common enable. When coming out of reset all the flops can be either set to "1" or reset to "0".

Output Enable Path

The output enable path has the same choices as the transmit data-path:

- Combinatorial – In this mode the enable coming from the FPGA fabric goes directly to the output buffer.
- SDR – In this mode the enable coming from the FPGA fabric gets registered in the bit-module before being forwarded to the output buffer.
- DDR – In this mode two bits of enable coming from FPGA fabric get converted to DDR data before being forwarded to the output buffer.

Figure 38 below shows the bit-module's output enable data-path.

All the flops shown below can be configured to be either positive or negative edge triggered. All the flops have an asynchronous reset. When coming out of reset all the flops can be either set to "1" or reset to "0".

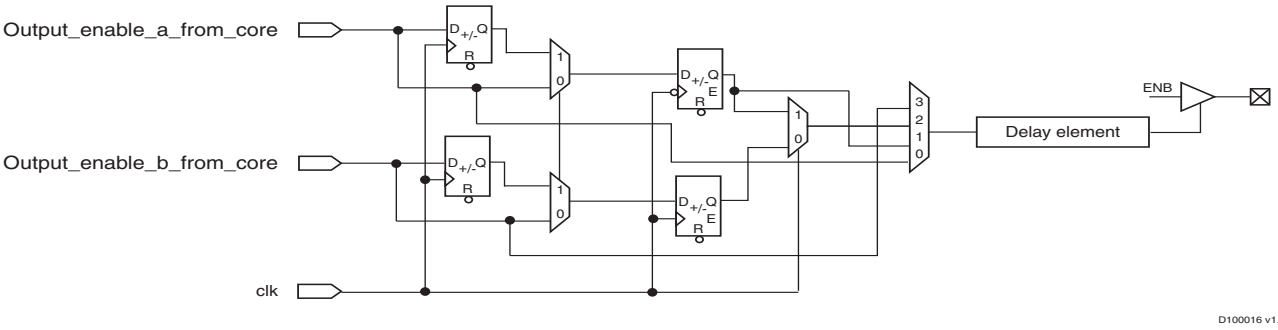


Figure 38: *Output Enable path*

Delay Element

Each bit-module has a delay element which can be used in either the receive data-path or in the transmit data-path. The delay of the delay element is set at configuration or dynamically during operation. When using a DLL as a master over the delay element, the delay element will maintain a set delay during PVT variations. In addition, when using the DLL, the delay can be adjusted in increments of 1/64 the period of the DLL's reference clock.

Table 13 below shows the full specifications of the delay element (DLL).

Table 13: *DLL Operating Conditions*

Parameter	Value
Clock frequency range	311MHz – 1066MHz
Reference input duty range	40% – 60%
Number of outputs per lane	1
Number of lanes per master	12
Slave delay adjustment	0% – 100% of reference clock cycle
Output phase resolution	6-bits
Output phase accuracy	± 4% of reference clock cycle
Maximum period jitter with noise frequency of 200 MHz and +/115mV sinusoidal noise	± 2% peak-to-peak of reference clock cycle
Minimum high/low slave pulse width	25% of reference clock cycle
Maximum lock time	<500 reference clock cycles

Advanced I/O Logic Resources

The bit-module has additional logic blocks to support the phy function of various memory interfaces (e.g., DDR2/DDR3 SDRAM, QDR2+ SRAM), network interfaces (e.g., SPI4.2) and other source synchronous interfaces.

Mesochronous Synchronizer

The receive data-path has an 8-bit FIFO synchronizer for mesochronous applications utilizing the Window Method for moving data from the "capture clock domain" over to the core clock domain. Window method is used for DDR data application; this allows both positive-edged and negative-edged received data to be valid for 4T cycles after capture. This provides the core a window of 4T to transfer captured data to its clock-domain. The synchronizer has a write-pointer for data written into the synchronizer, and a read-pointer for data to be read out of the synchronizer. The write-pointer is clocked by the received clock (e.g., DQS for DDR2, CQ/CQn for QDR2), whereas read-pointer is clocked by core Clock. In mesochronous mode, the wpb_wr_ptr counter will be incremented with each positive edge of DQS, whereas the wpb_rd_ptr will only be incremented when enabled by the core. The core enables this counter such that it samples the captured data in the middle of the window.

Phase Aligner

In many source-synchronous interfaces, bus de-skew is essential at the I/O for proper recovery of received data. Data-bits received across a bus may arrive at the FPGA at different times (called channel-to-channel skew). To de-skew the channels and align the bus on proper word boundary, protocols, such as SPI4.2, typically require the transmitter to send a training pattern to the receiver during initialization and/or after some interval. Per-bit skew is a way to effectively remove channel-to-channel skew and position the forwarded clock at the center of the data eye for each channel. The receive data-path has a phase-alignment circuit which, when used in conjunction with a training pattern and a delay element, can be used to achieve bit-alignment using per-bit skew.

Temperature Sensor

There is an on-chip temperature sensing diode. The anode and cathode for the internal temperature sensing diode are connected to two dedicated pins (TEMP_DIODE_P and TEMP_DIODE_N).

In a typical application, the user's circuitry can monitor the temperature and use the result as a decision criterion, for example:

- Selectively disabling circuits to reduce power consumption.
- Delay enabling selected circuits until a specified condition is reached.

DC and Switching Characteristics

Recommended Operating Conditions

Table 14: *Recommended Operating Conditions*[1]

Symbol	Description	Commercial			Units
		Min	Typical	Max	
T_J	Junction temperature	0		85	°C
V_{DDL}	Supply voltage for Internal logic fabric	0.97	1.00	1.03	V
V_{DD_CFG}	Supply voltage for configuration bank	0.97	1.00	1.03	V
V_{DD_CFGWL} (normal)	Supply voltage for configuration word line -No read back	0.97	1.00	1.03	V
V_{DD_CFGWL} (read back)	Supply voltage for configuration word line – to allow configuration read back	0.87	0.90	0.93	V
V_{CC}	Digital Power	0.97	1.00	1.03	V
V_{DD_BRAM}	Supply voltage for BRAMs	0.97	1.00	1.03	V
V_{DDO_Bxx*} (1.2V)	Supply voltage for I/O bank – 1.2V standards	1.10	1.20	1.30	V
V_{DDO_Bxx*} (1.5V)	Supply voltage for I/O bank – 1.5V standards	1.40	1.50	1.60	V
V_{DDO_Bxx*} (1.8V)	Supply voltage for I/O bank – 1.8V standards	1.70	1.80	1.89	V
V_{DDO_JCFG}	Supply voltage for configuration I/O	1.70	1.80	1.89	V
V_{DDO_CBxx*}(1.2V)	Supply voltage for clock I/O bank – 1.2V standards	1.10	1.20	1.30	V
V_{DDO_CBxx*} (1.5V)	Supply voltage for clock I/O bank – 5V standards	1.40	1.50	1.60	V
V_{DDO_CBxx*} (1.8V)	Supply voltage for clock I/O bank – 8V standards	1.70	1.80	1.89	V
V_{PA_VDD1}	SerDes low voltage	0.90	0.95	1.05	V
V_{PA_VDD2}	SerDes high voltage	1.71	1.80	1.98	V
$V_{DDA\_NOM\_E}$	Nominal analog voltage	0.95	1.00	1.15	V
$V_{DDA\_NOM\_W}$	Nominal analog voltage	0.95	1.00	1.15	V
V_{CCFHV_EFUSE} (normal)	Efuse power supply for fuse program/erase operations. 1V nominal for non efuse-programing mode	0.95	1.00	1.15	V
V_{CCFHV_EFUSE} (efuse programming)	Efuse Power supply for fuse program/erase operations. 2.2V Nominal for efuse program/erase	2.15	2.20	2.30	V
V_{CCRAM_EFUSE}	Efuse power supply normal/read operations	0.95	1.00	1.15	V
$AV_{DD\_PLL\_xxx**}$	Analog supply voltage for PLL	1.60	1.70	1.80	V
V_{REF_Bxx*}	Reference voltage for I/O receivers		V_{DDO} / 2		V

Notes:

* Substitute bank number for xx

** Substitute location identifier for xxx

The following table provides data on noise margins for the analog supplies used for high speed SerDes interfaces and memory interfaces at high data rates. For noise margins on digital rails and core fabric power supplies, please use **Table 14** above to ensure that IR drop and ripples while your system is in operation are within the specified tolerances.

Table 15: *AC Noise Margins on Analog Interface Supplies*

Parameter	Notes	Max	Units
$PA\_VDD2_{AC\text{-}LOFREQ}$	0.95V SerDes analog supply voltage maximum AC power supply noise Total Integrated Peak-Peak noise for frequencies from 1KHz to 10MHz	0.03	V_{pkpk}
$PA\_VDD1_{AC\text{-}HIFREQ}$	1.8V SerDes analog supply voltage maximum AC power supply noise Total Integrated Peak-Peak noise for frequencies from 1KHz to 10MHz	0.03	V_{pkpk}
$PA\_VDD2_{AC\text{-}HIFREQ}$	0.95V SerDes analog supply voltage maximum AC power supply noise Total Integrated Peak-Peak noise for frequencies of 10MHz and higher	0.05	V_{pkpk}
$PA\_VDD2_{AC\text{-}HIFREQ}$	1.8V SerDes analog supply voltage maximum AC power supply noise Total Integrated Peak-Peak noise for frequencies of 10MHz and higher	0.05	V_{pkpk}
$VDDO\_Bxx_{AC}$ (1.2V/1.5V/1.8V)	GPIO bank buffer analog supply voltage maximum AC power supply noise	0.05	V_{pkpk}
$VDDO\_CBxx_{AC}$ (1.2V/1.5V/1.8V)	Clock bank buffer analog supply voltage maximum AC power supply noise	0.05	V_{pkpk}
$VDDA\_NOM\_E/W_{AC}$	Nominal I/O circuitry analog supply voltage maximum AC power supply noise	0.03	V_{pkpk}
$AVDD\_PLL\_xxx_{AC}$	PLL analog supply voltage maximum AC power supply noise	0.03	V_{pkpk}

I/O Electrical Specifications

LVCMOS18

Table 16: *LVCMOS18 Supply Voltage*

Symbol	Parameter	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.7	1.8	1.89	V

Table 17: *LMCMOS18 DC Specifications*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{IH}	Input High Voltage	$V_{OUT} \geq V_{OH}$	0.65 V_{DDO}	V_{DDO} + 0.3	V
V_{IL}	Input Low Voltage	$V_{OUT} \leq V_{OL}$	–0.3	0.35 V_{DDO}	V
V_{OH}	Output High Voltage	I_{OH} = –4mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –6mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –8mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –12mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –16mA	V_{DDO}(min) – 0.4		V
V_{OL}	Output Low Voltage	I_{OL} = 4mA		0.4	V
		I_{OL} = 6mA		0.4	V
		I_{OL} = 8mA		0.4	V
		I_{OL} = 12mA		0.4	V
		I_{OL} = 16mA		0.4	V

LVCMOS15

Table 18: *LVCMOS15 Supply Voltages*

Symbol	Parameter	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.4	1.5	1.6	V

Table 19: *LVCMOS15 DC Specifications*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{IH}	Input High Voltage	$V_{OUT} \geq V_{OH}$	0.65 V_{DDO}	V_{DDO} + 0.3	V
V_{IL}	Input Low Voltage	$V_{OUT} \leq V_{OL}$	–0.3	0.35 V_{DDO}	V
V_{OH}	Output High Voltage	I_{OH} = –4mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –6mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –8mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –12mA	V_{DDO}(min) – 0.4		V
		I_{OH} = –16mA	V_{DDO}(min) – 0.4		V

Table 19: *LVCMOS15 DC Specifications (Continued)*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{OL}	Output Low Voltage	I_{OL} = 4mA		0.4	V
		I_{OL} = 6mA		0.4	V
		I_{OL} = 8mA		0.4	V
		I_{OL} = 12mA		0.4	V
		I_{OL} = 16mA		0.4	V

LVCMOS12

Table 20: *LVCMOS12 Supply Voltages*

Symbol	Parameter	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.1	1.2	1.3	V

Table 21: *LVCMOS15 DC Specifications*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{IH}	Input High Voltage	$V_{OUT} \geq V_{OH}$	0.65 V_{DDO}	V_{DDO} + 0.3	V
V_{IL}	Input Low Voltage	$V_{OUT} \leq V_{OL}$	–0.3	0.35 V_{DDO}	V
V_{OH}	Output High Voltage	I_{OH} = –4mA	0.75* V_{DDO}		V
		I_{OH} = –6mA	0.75* V_{DDO}		V
		I_{OH} = –8mA	0.75* V_{DDO}		V
		I_{OH} = –12mA	0.75* V_{DDO}		V
		I_{OH} = –16mA	0.75* V_{DDO}		V
V_{OL}	Output Low Voltage	I_{OL} = 4mA		0.25* V_{DDO}	V
		I_{OL} = 6mA		0.25* V_{DDO}	V
		I_{OL} = 8mA		0.25* V_{DDO}	V
		I_{OL} = 12mA		0.25* V_{DDO}	V
		I_{OL} = 16mA		0.25* V_{DDO}	V

HSTL18

Table 22: *HSTL18 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.7	1.8	1.89	V
V_{REF}	Output reference voltage		V_{DDO} / 2		V
V_{TT} (1)	Termination voltage for Class I and II outputs		V_{DDO} / 2		V

Notes:

1. V_{TT} must track V_{REF} of receiving device.

Table 23: *HSTL18 DC Specifications*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{IH}	High input voltage(1)	Single-ended	V_{REF} + 0.1	V_{DDO} + 0.3	V
V_{IL}	Low input voltage(1)	Single-ended	–0.3	V_{REF} – 0.1	V
V_{OH}	High output voltage				
	Class I buffer	I_{OH} = 8 mA	V_{DDO} – 0.4		V
	Class II buffer	I_{OH} = 16 mA	V_{DDO} – 0.4		V
V_{OL}	Low output voltage				
	Class I buffer	I_{OL} = –8 mA		0.4	V
	Class II buffer	I_{OL} = –16 mA		0.4	V

Notes:

1. Input buffer is single-ended only.

HSTL15

Table 24: *HSTL15 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.4	1.5	1.6	V
V_{REF}	Output reference voltage		V_{DDO}/2		V
V_{TT}(1)	Termination voltage for Class I and II outputs		V_{DDO}/2		V

Notes:

1. V_{TT} must track V_{REF} of receiving device.

Table 25: *HSTL15 DC Specifications*

Symbol	Description	Condition	Specification		Units
			Min	Max	
V_{IH}	High input voltage(1)	Single-ended	V_{REF} + 0.1	V_{DDO} + 0.3	V
V_{IL}	Low input voltage(1)	Single- ended	–0.3	V_{REF} – 0.1	V
V_{OH}	High output voltage				
	Class I buffer	I_{OH} = 8 mA	V_{DDO} – 0.4		V
	Class II buffer	I_{OH} = 16 mA	V_{DDO} – 0.4		V
V_{OL}	Low output voltage				
	Class I buffer	I_{OL} = –8 mA		0.4	V
	Class II buffer	I_{OL} = –16 mA		0.4	V

Notes:

1. Input buffer is single-ended only.

SSTL18

Table 26: *SSTL18 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.7	1.8	1.89	V
V_{REF}	Output reference voltage		V_{DDO} / 2		V
V_{TT}	Termination voltage	VREF-40mV	VREF	VREF+40mV	V

Table 27: *SSTL18 DC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(DC)}$	High DC input voltage	V_{REF} + 0.125	V_{DDO} + 0.3	V
$V_{IL(DC)}$	Low DC input voltage	–0.3	V_{REF} – 0.125	V
Output Buffer R_T = 25Ω, R_S = 20Ω				
I_{OH}	Output minimum source current(1)	–13.4		mA
I_{OL}	Output minimum sink current(1)	13.4		mA
Notes:				
1. V_{DDO} = 1.7V, $V_{OUT(MIN)}$ = 833 mV, V_{TT} = 40 mV				

Table 28: *SSTL18 AC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(AC)}$	High AC input voltage	V_{REF} + 0.25		V
$V_{IL(AC)}$	Low AC input voltage		V_{REF} – 0.25	V
Output Buffer R_{on} = 20Ω ODT = 25Ω				
$V_{OH(AC)}$	High AC output voltage(1)	V_{TT} + 0.57		V
$V_{OL(AC)}$	Low AC output voltage(1)		V_{TT} – 0.57	V
Output Buffer R_{on} = 20Ω ODT = 25Ω				
$V_{OH(AC)}$	High AC output voltage(2)	V_{TT} + 0.76		V
$V_{OL(AC)}$	Low AC output voltage(2)		V_{TT} – 0.76	V

SSTL15

Table 29: *SSTL15 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.425	1.5	1.575	V
V_{REF}	Input reference voltage		V_{DDO} / 2		V
V_{TT}	Termination voltage	V_{REF} -0.04	V_{REF}	V_{REF} +0.04	V

Table 30: *SSTL15 DC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(DC)}$	High DC input voltage	V_{REF} + 0.100		V
$V_{IL(DC)}$	Low DC input voltage		V_{REF} – 0.100	V
$V_{OH(DC)}$	High DC output voltage	$V_{TT} + 0.1 \times V_{DDO}$		V
$V_{OL(DC)}$	Low DC output voltage		V_{TT} - 0.1 * V_{DDO}	V
I_{OZK}	Off-state leakage current	47.1	135.6	uA
I_{DD}	Static Supply current	35	78	mA

Table 31: *SSTL15 AC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(AC)}$	High AC input voltage	V_{REF} + 0.175		V
$V_{IL(AC)}$	Low AC input voltage		V_{REF} – 0.175	V
$V_{OH(AC)}$	High AC output voltage	$V_{TT} + 0.1 \times V_{DDQ}$		V
$V_{OL(AC)}$	Low AC output voltage		$V_{TT} - 0.1 \times V_{DDQ}$	V

POD18

Table 32: *POD18 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.7	1.8	1.89	V
V_{REF}	Output reference voltage	0.69*V_{DDO}	0.70*V_{DDO}	0.71*V_{DDO}	V

Table 33: *POD18 DC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(DC)}$	High DC input voltage	V_{REF} + 0.15		V
$V_{IL(DC)}$	Low DC input voltage		V_{REF} - 0.15	V
I_L	Input Leakage	-5	5	µA
$V_{OL(DC)}$	Output Logic Low		0.76	V

Table 34: *POD18 AC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(AC)}$	High AC input voltage	V_{REF} + 0.25		V
$V_{IL(AC)}$	Low AC input voltage		V_{REF} -0.25	V

POD15

Table 35: *POD15 General Specifications*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	Output supply voltage relative to GND	1.455	1.5	1.545	V
V_{REF}	Output reference voltage	0.69*V_{DDO}	0.70*V_{DDO}	0.71*V_{DDO}	V

Table 36: *POD15 DC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(DC)}$	High DC input voltage	V_{REF} + 0.12		V
$V_{IL(DC)}$	Low DC input voltage		V_{REF} - 0.12	V
$V_{OL(DC)}$	Output Logic Low		0.62	V

Table 37: *POD15 AC Specifications*

Symbol	Description	Specification		Units
		Min	Max	
$V_{IH(AC)}$	High AC input voltage	V_{REF} + 0.20		V
$V_{IL(AC)}$	Low AC input voltage		V_{REF} -0.20	V

LVDS18

The low voltage differential signaling (LVDS) implementation in Achronix I/O rely on the differential mode being obtained by using single ended drivers. It does *not* provide a common mode voltage on the transmitter, and requires a 100Ω termination across the two differential drivers on the receive channel to obtain the common mode. The common mode voltage is expected to be proportional to the supply voltage.

Table 38: *LVDS18 Supply Voltages*

Symbol	Description	Min	Nom	Max	Units
V_{DDO}	I/O supply voltage	1.7	1.8	1.89	V

Table 39: *LVDS18 DC Specifications*

Symbol	Description	Conditions	Min	Max	Units
Receiver					
V_I	Input voltage range, V_{IA}/V_{IB}		0	1.8	V
V_{IDTH}	Input differential threshold		-100	+100	mV
V_{ID}	Input differential voltage range		100	900	mV
R_{IN}	Receiver diff input impedance	Worst case: SS, 0°C	90	110	Ω
Driver					
V_{OH}	Output voltage High, V_{OA}/V_{OB}	R_{LOAD} = 100Ω ± 1%		1.512	V
V_{OL}	Output voltage Low, V_{OA}/V_{OB}		395		mV
$\|V_{OD}\|$	Output differential voltage		240	1050	mV
V_{OS}	Output offset voltage		0.85	0.945	V
R_O	Output impedance, single ended	V_{CM} = 1.0V and 1.4V	36	44	Ω
ΔR_O	R_O mismatch between A and B			10	%
I_{SA}/I_{SB}	Output Current	Driver shorted to ground	38.63	52.5	mA
I_{SAB}	Output Current	Drivers shorted together	19.31	26.25	mA

Table 40: *LVDS18 AC Specifications*

Symbol	Description	Conditions	Min	Max	Units
Receiver					
T_{PD}	Propagation delay		1.285	0.6	ns
Driver					
T_{FALL}	V_{OD} fall time, 20–80%	R_{LOAD} = 100Ω ± 1%	250	200	ps
T_{RISE}	V_{OD} rise time, 20–80%		250	200	ps

I/O Leakage

Table 41: *I/O Tri-state Leakage Currents*

Symbol	Description	Condition	Min	Max	Units
I_{OZ}	Off-state Leakage Current(1)(2)	$1.62 \leq V_{DDO} \leq 1.986$	90	183	µA
		$1.4 \leq V_{DDO} \leq 1.6$	47	105	µA

Notes:

1. Includes input leakage current
2. At 85 degrees C

Compensation and Termination

The EF banks in Speedster devices have compensation controller blocks to provide accurate process, voltage, and temperature compensated driver output and receiver termination impedances. Some I/O standards require termination or compensation resistors for proper operation.

The external resistance value needs to be the same as the desired driver RON value. Similarly for termination, the external resistor needs to be the same as the desired termination resistance.

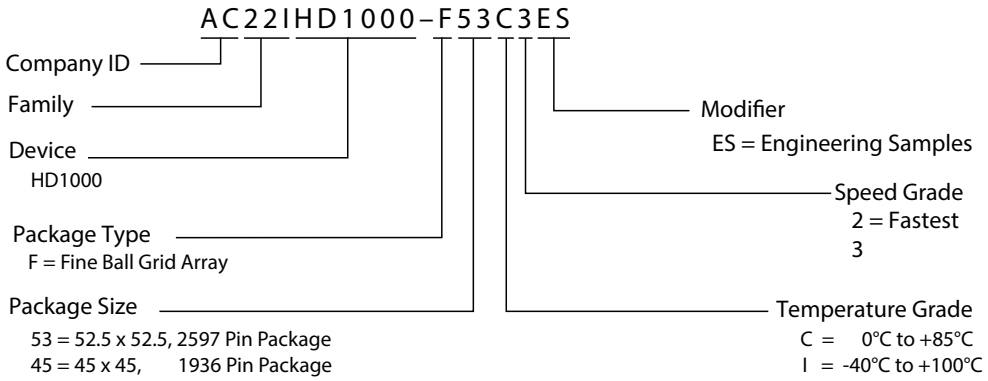
Table 42: Range of Drive Resistance (R_{drv})

VCCA_1PHV	Min	Max	Units
1.2V	18	74	Ω
1.5V	16.8	72	Ω
1.8V	16.3	70.7	Ω

Table 43: Range of ODT Resistance (R_{odt})

Mode	Min	Max	Units
1.2V	27	222	Ω
1.5V	25.2	216	Ω
1.8V	24.45	212.1	Ω

Ordering Information



ds001_33_v08

Revision History

The following table lists the revision history of this document.

Version	Revision
1.0	Initial released version
1.1	• Minor updates, corrections and format fixes. • Update the description **"Enhanced Function (EF) Bank,"** on page 40.
1.2	• Added section **"Speed Grade Options,"** on page 2. • Updated maximum SerDes speed to 10.3125 Gbps. • Clarified maximum data rates.

Achronix Semiconductor Corporation

2953 Bunker Hill Lane, Suite 101
Santa Clara, CA, 95045
USA

Phone : 877.GHZ.FPGA (877.449.3742)
Fax : 408.286.3645
E-mail : info@achronix.com