Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	4992
Total RAM Bits	75776
Number of I/O	278
Number of Gates	397000
Voltage - Supply	1.425V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	680-BBGA
Supplier Device Package	680-FPBGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/ort8850l-2bm680c

Table of Contents

- New 200 MHz embedded quad-port RAM blocks, two read ports, two write ports, and two sets of byte lane enables. Each embedded RAM block can be configured as:
  - One—512 x 18 (quad-port, two read/two write) with optional built-in arbitration.
  - One—256 x 36 (dual-port, one read/one write).
  - One—1K x 9 (dual-port, one read/one write).
  - Two—512 x 9 (dual-port, one read/one write for each).
  - Two RAM with arbitrary number of words whose sum is 512 or less by 18 (dual-port, one read/one write).
  - Supports joining of RAM blocks.
  - Two 16 x 8-bit Content Addressable Memory (CAM) support.
  - FIFO 512 x 18, 256 x 36, 1K x 9, or dual 512 x 9.
  - Constant multiply (8 x 16 or 16 x 8).
  - Dual variable multiply (8 x 8).
- Embedded 32-bit internal system bus plus 4-bit parity interconnects FPGA logic, MicroProcessor Interface (MPI), embedded RAM blocks, and embedded backplane transceiver blocks with 100 MHz bus performance. Included are built-in system registers that act as the control and status center for the device.
- Built-in testability:
  - Full boundary scan (*IEEE* 1149.1 and Draft 1149.2 JTAG).
  - Programming and readback through boundary scan port compliant to *IEEE* Draft 1532:D1.7.
  - TS_ALL testability function to 3-state all I/O pins.
  - New temperature-sensing diode.
- Cycle stealing capability allows a typical 15% to 40% internal speed improvement after final place and route. This feature also supports compliance with many setup/hold and clock to out I/O specifications and may provide reduced ground bounce for output buses by allowing flexible delays of switching output buffers.

Programmable Logic System Features

- PCI local bus compliant for FPGA I/Os.
- Improved *PowerPC/Power QUICC MPC860* and *PowerPC* II *MPC8260* high-speed synchronous MicroProcessor Interface can be used for configuration, readback, device control, and device status, as well as for a general-purpose interface to the FPGA logic, RAMs, and embedded backplane transceiver blocks. Glueless interface to synchronous *PowerPC* processors with user-configurable address space provided.
- New embedded *AMBA*™ specification 2.0 AHB system bus (*ARM*® processor) facilitates communication among the MicroProcessor Interface, configuration logic, embedded block RAM, FPGA logic, and backplane transceiver logic.
- New network PLLs meet ITU-T G.811 specifications and provide clock conditioning for DS-1/E-1 and STS-3/STM-1 applications.
- Variable size bused readback of configuration data capability with the built-in MicroProcessor Interface and system bus.
- Internal, 3-state, and bidirectional buses with simple control provided by the SLIC.
- New clock routing structures for global and local clocking significantly increases speed and reduces skew (<200 ps for OR4E04).
- New local clock routing structures allow creation of localized clock trees.
- Two new edge clock routing structures allow up to six high-speed clocks on each edge of the device for improved setup/hold and clock-to-out performance.
- New Double-Data Rate (DDR) and Zero-Bus Turn-around (ZBT) memory interfaces support the latest high-speed memory interfaces.
- New 2x/4x uplink and downlink I/O capabilities interface high-speed external I/Os to reduced speed internal logic.

SONET Framing

Each 850 Mbits/s serial link uses a pseudo-SONET protocol. SONET A1/A2 framing is used on the link to detect the 8 kHz frame location. The link is also scrambled using the standard SONET scrambler definition to ensure proper transitions on the link for improved CDR performance. The ORT8850 can do SONET framing and scrambling in both STS-12 and STS-3 formats.

Elastic buffers (FIFOs) are used to align each incoming STS-12 link to the local 77.76 MHz clock and 8 kHz frame. These FIFOs will absorb delay variations between the eight channels due to timing skews between cards and along backplane traces. For greater variations, a streamlined pointer processor (pointer mover) within the STM macro will align the 8 kHz frames regardless of their incoming frame position.

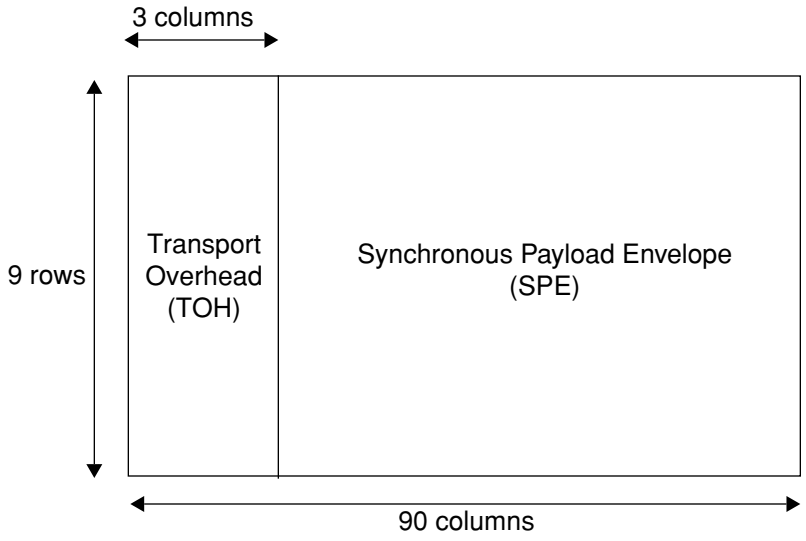
The data rates for SONET are covered in the following table. Values that fall in between those shown in the table for each mode are supported (126.00 Mbits/s - 212.50 Mbits/s, 504.00 Mbits/s - 850.00 Mbits/s). 63.00 MHz is the slowest reference clock while 106.25 MHz is the fastest reference clock frequency supported.

Table 2. Supported SONET Data Rates

Reference Clock	STS-12 Mode	STS-3 Mode
63 MHz	504.00 Mbits/s	126.00 Mbits/s
77.76 MHz	622.08 Mbits/s	155.52 Mbits/s
106.25 MHz	850.00 Mbits/s	212.50 Mbits/s

An STS-N frame can be broadly divided into the Transport Overhead (TOH) and the Synchronous Payload Envelope (SPE) areas. The TOH comprises of bytes that are used for framing, error detection and various other functions. The start of the SPE can begin at any point in a SONET frame. The start of the SPE is determined using the pointer bytes located in the TOH. The basic STS-1 frame is shown in Figure 4. Higher rate STS_N signals are created by byte interleaving N STS-1 signals. Some TOH bytes have slightly different functions in STS-N frames than in the basic STS-1 frame. The ORT8850 offers both a transparent option and a serial insertion option for processing the TOH bytes.

Figure 4. STS-1 Frame Format



LVDS Reference Clock

The reference clock for the ORT8850 SERDES is an LVDS input (SYS_CLK_[P:N]). This reference clock can run in the range from 63.00 MHz to 106.25 MHz and is used to clock the entire Embedded Core. This clock is also available in the FPGA interface as the output signal FPGA_SYSCLK at the Embedded Core/FPGA Logic interface.

The supported range of reference clock frequencies will drive the internal and link serial rates from 504 MHz to 850 MHz. For standard SONET applications a reference clock rate of 77.76 MHz will allow the ORT8850 to communicate with standard SONET devices. If the ORT8850 is communicating with another ORT8850, the reference clock can run anywhere in the defined range. When using a non 77.76 MHz reference clock, the frame pulse will now need to be derived from the non standard rate thus making the frame pulse rate not 8 kHz, but rather a single clock pulse every 9720 clock cycles.

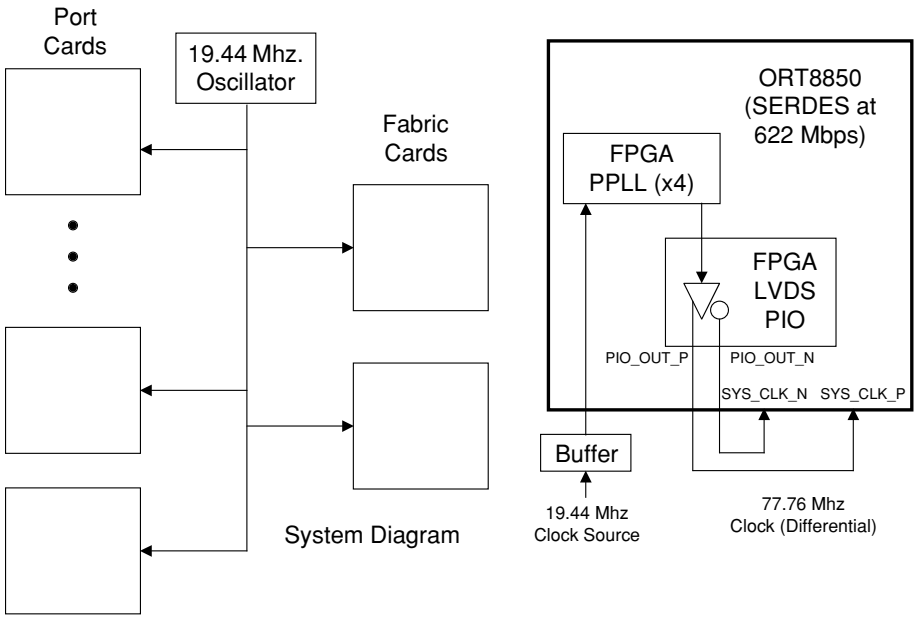
System Considerations for Reference Clock Distribution

There are two main system clocking architectures that can be used with the ORT8850 at the system level to provide the LVDS reference clocks. The recommended approach is to distribute a single reference clock to all boards. However, independent clocks can be used on each board provided that they are matched with sufficient accuracy and the alignment is not used. These two approaches are summarized in the following paragraphs

Distributed Clocking

A distributed clock architecture, shown in Figure 5, uses a single source for the system reference clock. This single source drives all devices on both the line and switch sides of the backplane. Typically this is a lower speed clock such as a 19.44 MHz signal. An external PLL on each board or and internal ORT8850 FPGA PLL is then used to multiply the clock to the desired reference clock rate (i.e. by 4x to 77.76 MHz if the distributed clock is at 19.44 MHz). Using this type of clock architecture the ORT8850 data channels are fully synchronous and no domain transfer is required from the transmitter to the receiver.

Figure 5. Distributed Clock Architecture



Independent Clocking

An independent clock architecture uses independent clock sources on each ORT8850 board. With this architecture, for the SERDES to sample correctly the independent oscillators must be within reference clock tolerance requirements for the Clock and Data Recovery (CDR) to correctly sample the incoming data and recover data and clock. The local reference clock and the recovered clock will not be synchronous since they are created from a different source. The alignment FIFO uses the recovered clock for write and the local reference clock for read. Due to

performed, one extra payload byte is inserted into the SONET frame. The H3 byte is used to hold this extra byte and is hence called the pointer action byte. When justification is not being performed, this byte contains a default value of 0x00.

- B2 - Line Bit-Interleaved Parity code (BIP-8) byte - This byte carries the parity information which is used to check for transmission errors in a line. This is a even parity computed over all the bytes of the frame, except section overhead bytes, before scrambling. The computed parity value is transmitted in the next frame in the B2 position. This byte is defined for all the STS-1signals in an STS-N signal.
- K1, K2 - Automatic Protection Switching (APS channel) bytes - These bytes carry the APS information. They are used for implementing automatic protection switching and for transmitting the line Alarm Indication Signal (AIS-L) and the Remote Defect Indication (RDI-L) signal.
- D4 to D12 - Line Data Communications Channel (DCC) bytes - These bytes provide a 576 Kbits/s channel for transmission of information.
- S1- Synchronization Status - This byte carries the synchronization status of the network element. It is located in the first STS-1 of an STS-N. Bits 5 through 8 (as defined in GR-253) of this byte carry the synchronization status.
- Z1 - Growth - This byte is located in the second through Nth STS-1s of an STS-N and are allocated for future growth. An STS-1 signal does not contain a Z1 byte.
- M0 - STS-1 REI-L - This byte is defined only for STS-1 signals and is used to convey the Line Remote Error Indication (REI-L). The REI-L is the count of the number of B2 parity errors detected by an LTE and is transmitted to its peer LTE as feedback information. Bits 5 through 8 of this byte are used for this function.
- E2 - Orderwire byte - This byte carries for line orderwire information.

In the ORT8850 transmit path, the TOH processing is confined to the framing and byte interleaved parity bytes. The remaining bytes are either passed through transparently or inserted from data sent from the serial TOH interface.

In the receive direction, the TOH bytes are stripped and optionally sent to the FPGA logic through the serial TOH interface. Regenerated framing bytes are sent to the FPGA on the parallel data bus. The APS bytes K1 and K2 can be optionally passed through the pointer mover under software control, or can be set to zero. The header bytes, J0 and C1 are also detected and used by the receive path, as will be discussed in a later section.

The serial TOH processing block is clocked by the TOH_CLK. If using the TOH bytes in the serial insertion mode to support a communication channel, this TOH_CLK should be driven from the FPGA interface. The TOH processor operates from 25 MHz to 106 MHz. A domain clock transfer takes place inside the TOH block and the TOH processor does not need to run as fast as the data.

Transparent Insert Mode

In the transmit direction the SPE and TOH data received on parallel input bus is transferred unaltered to the serial LVDS output. However, B1 byte of STS-1 is always replaced with a new calculated value (the 11 bytes following B1 are replaced with all zeros). Also, A1 and A2 bytes of all STS-1s are always regenerated. The source for the TOH bytes in the transparent mode is summarized in Table 9. (The order of transmission is row by row, left to right, and then from top to bottom [most significant bit first]. SPE bytes are not shown.)

Table 9. Transmitter TOH on LVDS Output (Transparent Mode)

A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2												
B1	0	0	0	0	0	0	0	0	0	0	0																								

Regenerated bytes.

Transparent bytes from parallel input port.

Table 10. Transmitter TOH on LVDS Output (TOH Insert Mode)

A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	J0	J0	J0	J0	J0	J0	J0	J0	J0	J0	J0
B1	0	0	0	0	0	0	0	0	0	0	0	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3
H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3
B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K2	K2	K2	K2	K2	K2	K2	K2	K2	K2	K2
D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9
D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12
S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2

Regenerated bytes.

Bytes optionally inserted from TOH serial port data or transparently forwarded from parallel input port

Bytes transparently forwarded from parallel input port

Bytes transparently forwarded from parallel input port

Repeater

This block is essentially the inverse of the sampler block discussed in the receive path description section. It receives byte-wide STS-12 rate data from the TOH insert block. In order to support the STS-3 mode of operation, the HSI (622 Mbits/s) can be connected to a slower speed device (e.g., 155 Mbits/s). The purpose of this block is to rearrange the data being fed to the HSI so that each bit is transmitted four times, thus simulating 155 Mbits/s serial data. In STS-3 mode, the incoming STS-12 stream is composed of four identical STS-3s so only every fourth byte is used. The bit expansion process takes a single byte and stretches it to take up 4 bytes each consisting of 4 copies of the 8 bits from the original byte.

A1/A2 Processing

The A1 and A2 bytes provide a special framing pattern that indicates where a STS-1 begins in a bit stream. All 12 A1 bytes of each STS-12 are set to 0xF6, and all 12 A2 bytes are set to 0x28 automatically by the SONET framer. The latency from the transmit of the first bit of the A1 byte at the device output pins from the system frame pulse on the FPGA interface is between five to seven clock cycles of the reference clock (FPGA_SYSCLK).

The A1 and A2 bytes can also be intentionally corrupted for testing by the A1/A2 error insert control register (0x3000D, 0x3000E). Only the last A1 and first A2 are corrupted. When A1/A2 corruption detection is set for a particular channel, the A1/A2 values in the corrupted A1/A2 value registers are sent for the number of frames defined in the corrupted A1/A2 frame count register (0x3000C). When the corrupted A1/A2 frame count register is set to 0x00, A1/A2 corruption will continue until the A1/A2 error insert register is cleared.

The ORT8850 device only has one control register to set the A1/A2 bytes as well as the number of frames of corruption. To insert the corrupted A1/A2 each channel has an enable A1/A2 insert register. When the per channel error insert register bit is set, the A1/A2 values are corrupted for the number specified in the number of frames to corrupt. To insert errors again, the per channel error insert register bit must be cleared, and set again.

It is also possible to not insert the A1/A2 framing bytes using the per channel register bit “disable A1/A2 insert.”

B1 Processing

In the transmit direction a bit interleaved parity (BIP-8) error check set for even parity over all the bits of an STS-1 frame B1 is defined for the first STS-1 in an STS-12 only, the B1 calculation block computes a BIP-8 code, using even parity over all bits of the previous STS-12 frame after scrambling and is inserted in the B1 byte of the current STS-12 frame before scrambling.

Per-bit B1 corruption is controlled by the force BIP-8 corruption register (0x3000F). For any bit set in this register, the corresponding bit in the calculated BIP-8 is inverted before insertion into the B1 byte position. Each stream has an independent fault insert register that enables the inversion of the B1 bytes. B1 bytes in all other STS- 1s in the stream are filled with zeros.

It is also possible to not insert the B1 byte using the per channel register bit "disable B1 insert."

violated a per channel alarm bit will be set indicating that this channel has exceeded the threshold, as well as a FIFO out-of-sync alarm bit to indicate the channel is not longer in sync with the reset of the alignment group.

The incoming data can be considered as 4 STS-12 channels (A, B, C, and D) per quad. Thus we have STS-12 channels AA to AD from quad A of the STM and STS-12 channels BA to BD of quad B. The 8 channels of parallel SONET data can be grouped into an alignment group by 2, by 4 or all 8 channels. As the serial data is run through the backplane and SERDES the parallel data can be slightly varied. The alignment FIFO can absorb this difference in the channels and create a byte aligned grouping.

These streams can be frame aligned in the following patterns. Streams can be aligned on a twin STS-12 basis as shown in Figure 18. In STS-48 mode, all four STS-12s of each STM quad are aligned with each other (i.e. AA, AB, AC, AD) as shown in Figure 19. Optionally in STS-48 mode all eight STS-12s (STMs A and B) can be aligned which allows hitless switching since all streams will be byte aligned (Figure 20). Multiple ORT8850 devices can be aligned with each other using a common system frame pulse to enable STS-192 or higher modes.

Figure 18. Twin Channel Alignment

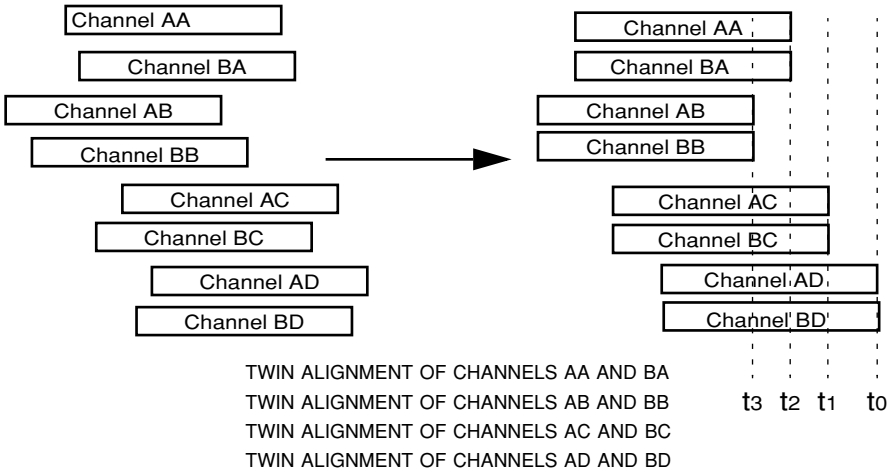


Figure 19. Alignment of SERDES Quads A and B

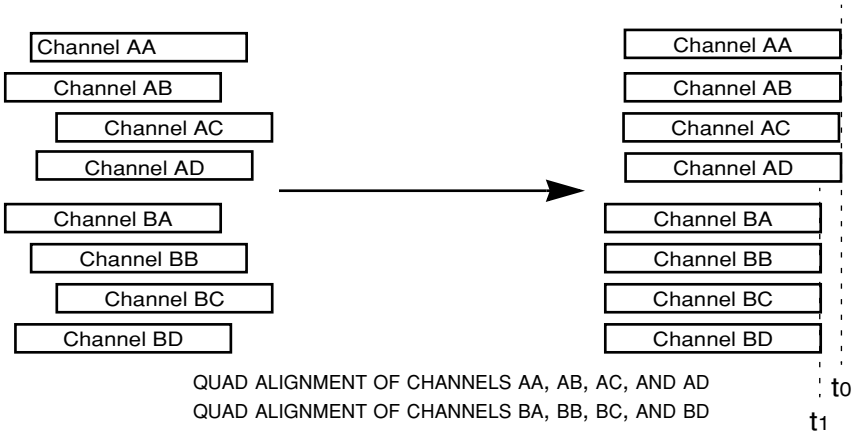


Table 13. Valid Starting Positions for and STS MC (Continued)

STS-1 Number	STS-3cSPE	STS-6cSPE	STS-9cSPE	STS-12cSPE	STS-15cSPE	STS-18c to STS-48c SPEs
46	Yes	No	No	No	No	No

Note:
Yes = STS-Mc SPE can start in that STS-1.
No = STS-Mc SPE cannot start in that STS-1.
- = Yes or no, depending on the particular value of M.

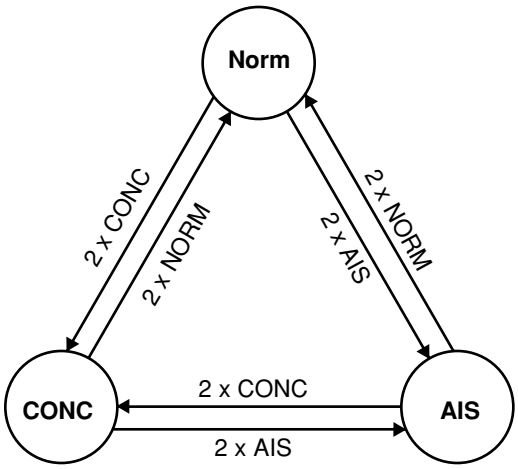
A pointer action byte (H3) is allocated for SPE frequency justification purposes. Frequency justification is discussed in a later section. The H3 byte is used in all STS-1s within an STS-N to carry the extra SPE byte in the event of a negative pointer adjustment. The value contained in this byte when it's not used to carry the SPE byte is undefined.

Pointer Interpreter State Machine.
The pointer interpreter's highest priority is to maintain accurate data flow (i.e., valid SPE only) into the elastic store. This will ensure that any errors in the pointer value will be corrected by a standard, fully SONET compliant, pointer interpreter without any data hits. This means that error checking for increment, decrement, and new data flag (NDF) (i.e., 8 of 10) is maintained in order to ensure accurate data flow. A single valid pointer (i.e., 0-782) that differs from the current pointer will be ignored. Two consecutive incoming valid pointers that differ from the current pointer will cause a reset of the J1 location to the latest pointer value (the generator will then produce an NDF). This block is designed to handle single bit errors without affecting data flow or changing state.

The pointer interpreter has only three states (NORM, AIS, and CONC). NORM state will begin whenever two consecutive NORM pointers are received. If two consecutive NORM pointers that both differ from the current offset are received, then the current offset will be reset to the last received NORM pointer. When the pointer interpreter changes its offset, it causes the pointer generator to receive a J1 value in a new position. When the pointer generator gets an unexpected J1, it resets its offset value to the new location and declares an NDF. The interpreter is only looking for two consecutive pointers that are different from the current value. These two consecutive NORM pointers do not have to have the same value. For example, if the current pointer is ten and a NORM pointer with offset of 15 and a second NORM pointer with offset of 25 are received, then the interpreter will change the current pointer to 25.

If the data is concatenated, the receipt of two consecutive CONC pointers causes CONC state to be entered. Once in this state, offset values from the head of the concatenation chain are used to determine the location of the STS SPE for each STS in the chain. Finally, if two consecutive AIS pointers cause the AIS state to occur. Any two consecutive normal or concatenation pointers will end this AIS state. This state will cause the data leaving the pointer generator to be overwritten with 0xFF.

Figure 22. Pointer Mover State Machine



phases (i.e., received and system) are determined. This latch point is then stable unless the relative framing changes and the received H byte times collide with the system F1 or E2 times, in which case the latch point would be switched to the collision-free byte time.

There is no restriction on how many or how often increments and decrements are processed. Any received increment or decrement is immediately passed to the generator for implementation regardless of when the last pointer adjustment was made. The responsibility for meeting the SONET criteria for maximum frequency of pointer adjustments is left to an upstream pointer processor.

Receive Bypass Options

Not all of the blocks in the receive direction are required to be used. The following bypass options are valid in the receive (backplane → FPGA) direction:

- STM Pointer Mover bypass:
  - In this mode, data from the alignment FIFOs is transferred to the FPGA logic. All channels are synchronous to the FPGA_SYSCLK signals driven to the FPGA logic, as is also the case when the pointer mover is not bypassed. During bypass SPE, C1J1, and data parity signals are not valid. When the pointer mover is bypassed, eight frame pulses (DOUTxx_FP) from aligned channels are provided by the embedded core to the FPGA.
  - When the pointer mover is used, the FPGA logic provides the frame pulse on the LINE_FP (recall: there is only one LINE_FP just like there is only one SYS_FP) signal essential for the Pointer Mover to move the data. The FPGA gets eight channels of SONET data with the A1 byte position of each channel of the TOH arbitrarily offset from the LINE_FP. The DOUTxx_FP signals are not valid when the pointer mover is used.
- STM Pointer Mover and Alignment FIFO bypass:
  - In this mode, data from the framer block is transferred to the FPGA logic. All channels supply data and frame pulses synchronous with their individual recovered clock (CDR_CLK_xx) per channel. During bypass, SPE, C1J1, and data parity signals are not valid. Additionally, no serial TOH_OUT_xx data and frame pulse signals will be available. The DOUTxx_FP signals are aligned with the A1 byte position of each channel, as shown in Figure 26.

Figure 26. Pointer Mover and Alignment FIFO Bypass Timing

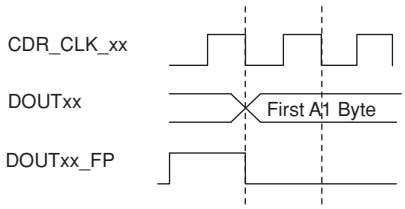


Table 14 shows the register settings to enable the bypass modes.

Table 14. Register Settings for Bypass Mode

Register Address	Value	Description
0x3000C	0x04	Turn off the SONET scrambler/descrambler
0x30020	0x07	Channel AA in functional mode
0x30038	0x07	Channel AB in functional mode
0x30050	0x07	Channel AC in functional mode
0x30068	0x07	Channel AD in functional mode
0x30080	0x07	Channel BA in functional mode
0x30098	0x07	Channel BB in functional mode
0x300B0	0x07	Channel BC in functional mode
0x300C8	0x07	Channel BD in functional mode

Table 14. Register Settings for Bypass Mode (Continued)

Register Address	Value	Description
0x30021	0x01	Channel AA in transparent mode
0x30039	0x01	Channel AB in transparent mode
0x30051	0x01	Channel AC in transparent mode
0x30069	0x01	Channel AD in transparent mode
0x30081	0x01	Channel BA in transparent mode
0x30099	0x01	Channel BB in transparent mode
0x300B1	0x01	Channel BC in transparent mode
0x300C8	0x01	Channel BD in transparent mode
0x30023	0x30	Channel AA - Do not insert A1/A2 or B1
0x3003B	0x30	Channel AB - Do not insert A1/A2 or B1
0x30053	0x30	Channel AC - Do not insert A1/A2 or B1
0x3006B	0x30	Channel AD - Do not insert A1/A2 or B1
0x30083	0x30	Channel BA - Do not insert A1/A2 or B1
0x3009B	0x30	Channel BB - Do not insert A1/A2 or B1
0x300B3	0x30	Channel BC - Do not insert A1/A2 or B1
0x300CB	0x30	Channel BD - Do not insert A1/A2 or B1
0x30037	0x44	Channel AA - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x3004F	0x44	Channel AB - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x30067	0x44	Channel AC - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x3007F	0x44	Channel AD - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x30097	0x44	Channel BA - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300AF	0x44	Channel BB - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300C7	0x44	Channel BC - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300DF	0x44	Channel BD - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer

Note: To select between full, half and quad rate modes, registers 0x300E1 and 0x300E2 are used. See the memory map for details on these registers.

FPGA/Embedded Core Interface Signals

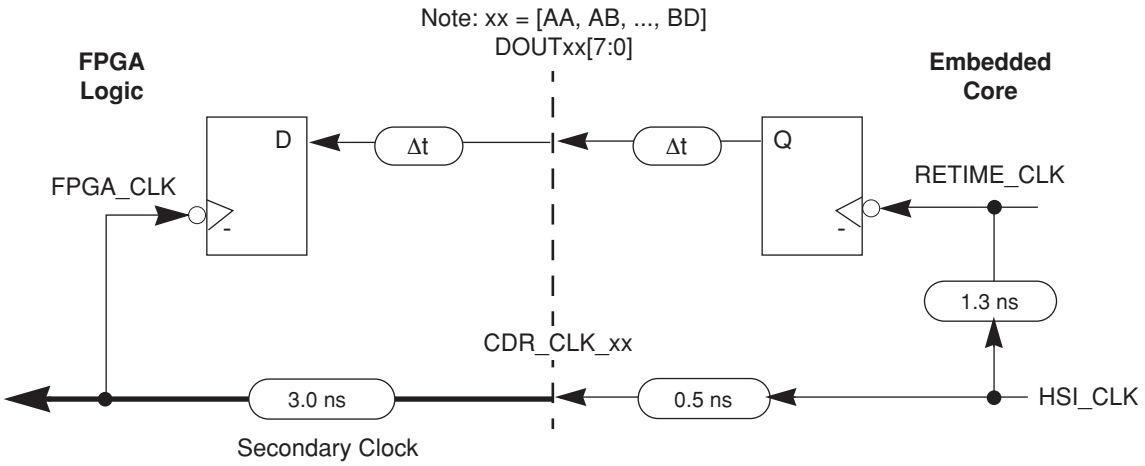
Table 15. FPGA/Embedded Core Interface Signals

ORT8850 FPGA/Embedded Core Interface Signals - SONET Blocks		
FPGA/Embedded Core Interface Signal Name xx=[AA,…,BD]	**Input (I) to or Output (O) from Core**	**Signal Description**
Common Interface Signals		
FPGA_SYSCLK	O	Local reference clock from the core to the FPGA. All of the transmit data is captured on this clock edge inside the ORT8850 core. If using the alignment FIFO all of the parallel data from the ort8850 core will also be clocked from this clock. This signal uses an ORCA Series4 primary clock route.

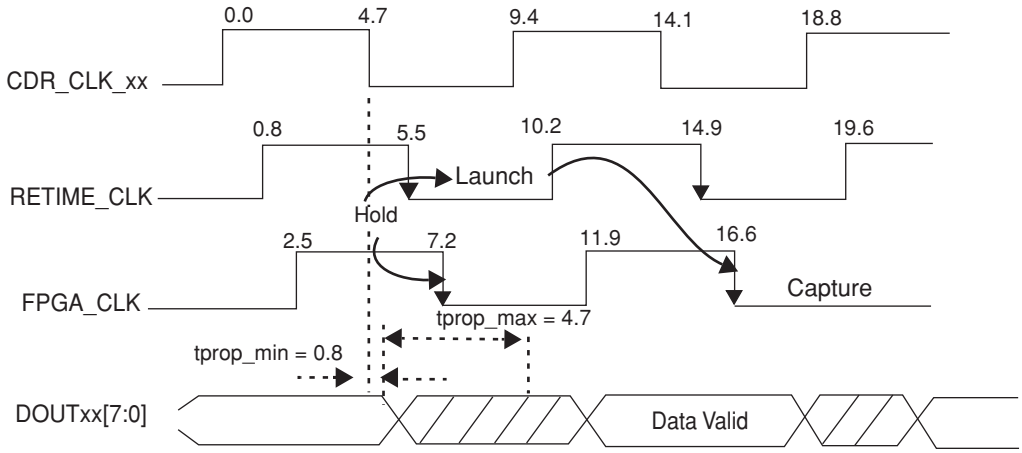
tions. (The clock edge on which data is latched in the core is hard wired to be the falling edge.) Since the falling edge of the clock (FPGA_CLK) at the FPGA latch occurs after the next data byte is launched, the delay from the interface to the FPGA latch must be large enough that an acceptable hold time margin is obtained. However the maximum propagation delay is fairly large, so a half cycle approach might lead to setup time problems.

Figure 27. Full Cycle, Alignment FIFO Bypass Mode Output Configuration and Timing (-1 Speed Grade)

a. Configuration



b. Timing (ns)



In the case shown in Figure 28 the alignment FIFO is used and all timing is with respect to the single reference clock, which is routed through the FPGA as a primary clock. The capturing clock edge occurs after the launch of the next data byte, so hold time margin is of concern and an acceptably margin should be verified. Launched data has nearly a full clock period to become stable at the capture latch, so setup margin should not be a problem. Moving the capture to the rising clock edge might give a setup time margin problem.

Figure 28. Half Cycle, Alignment Mode Output Configuration and Timing (-1 Speed Grade)

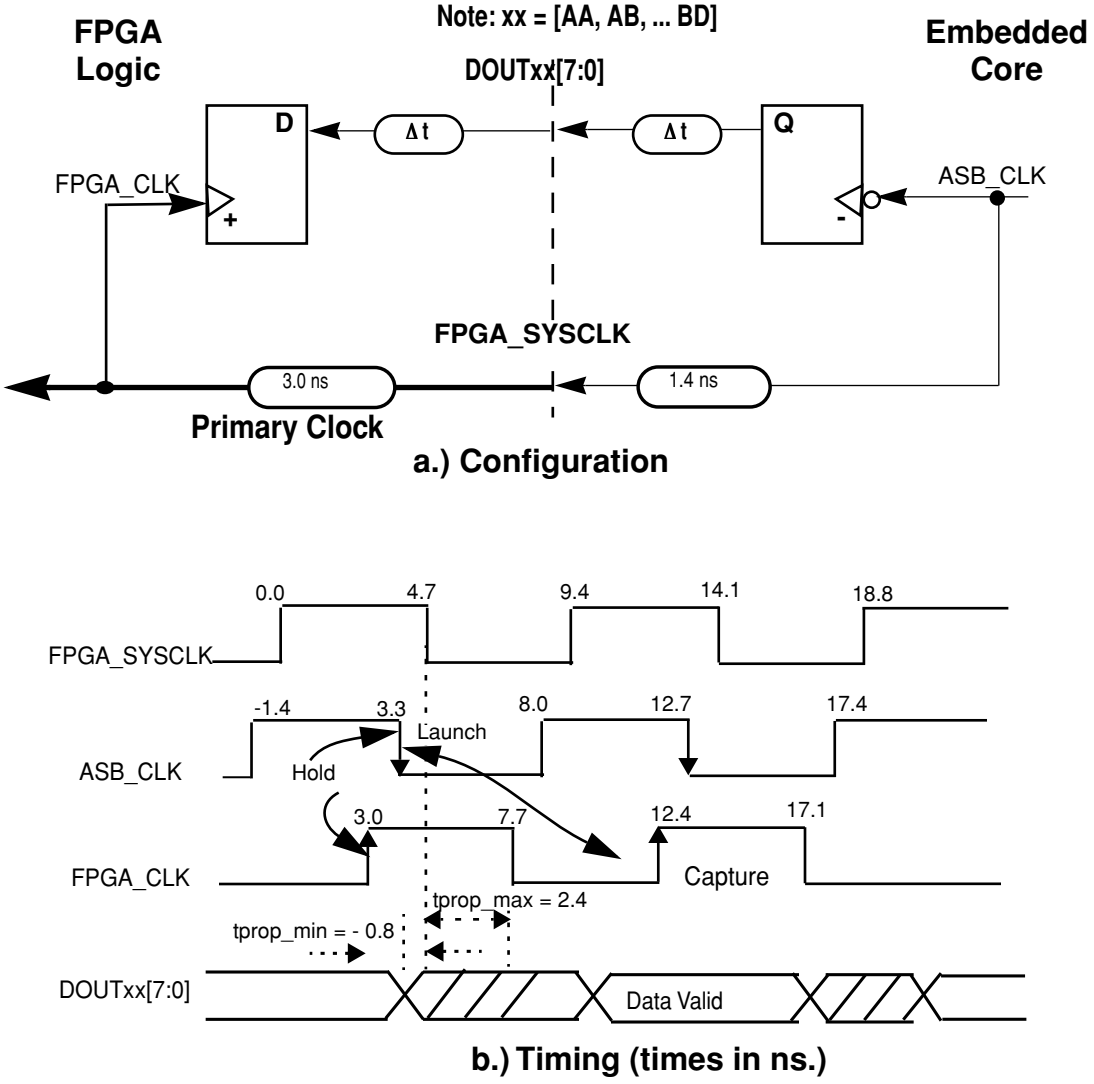
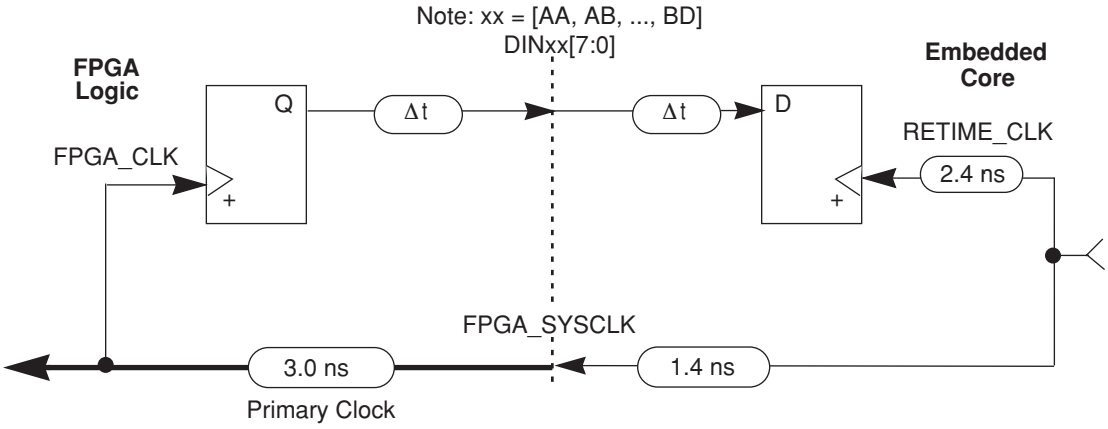


Figure 29 shows the timing for sending data from the FPGA logic to the Core. In the input case, the constraints on the data are specified in terms of setup and hold times on the data at the interface relative to the clock at the interface. For correct operation these constraints must be met. In the case shown, launch and capture occur on the same (rising) clock edge. Data is captured before the next data is launched, so there will be no hold margin problem. Launched data also has nearly a full clock period to become stable at the capture latch, so setup margin should not be a problem.

Figure 29. Full Cycle, Align and Bypass Mode Input Configuration and Timing (-1 Speed Grade)

a.) Configuration



a.) Timing (ns)

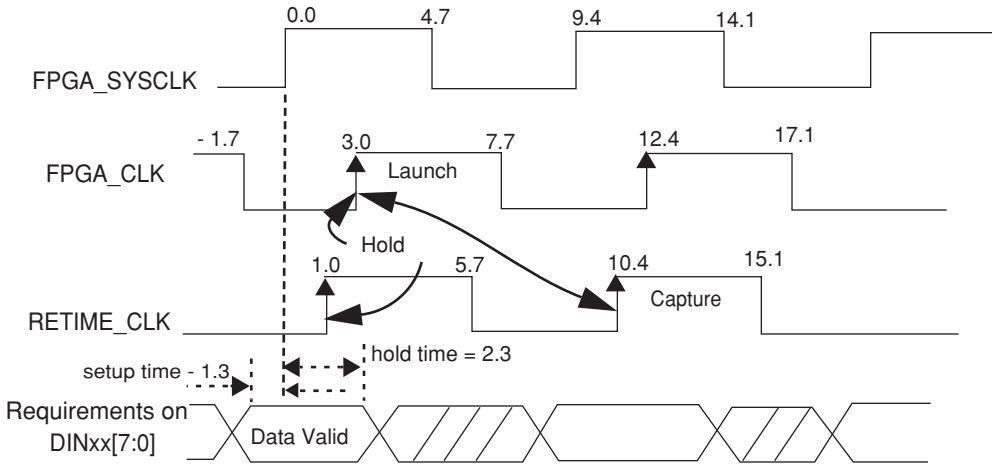


Table 18. LVDS Protection Switching (Continued)

FPGA Interface Signal	When '0'	When '1'
LVDS_PROT_BA	Channel BA gets TXD_BA_W_[P:N]	Channel BA gets TXD_BA_P_[P:N]
LVDS_PROT_BB	Channel BB gets TXD_BB_W_[P:N]	Channel BB gets TXD_BB_P_[P:N]
LVDS_PROT_BC	Channel BC gets TXD_BC_W_[P:N]	Channel BC gets TXD_BC_P_[P:N]
LVDS_PROT_BD	Channel BD gets TXD_BD_W_[P:N]	Channel BD gets TXD_BD_P_[P:N]

For software control of the LVDS protection switching there is an enable bit to enable software control, and a bit per channel which selects main or protect. The enable register is at 0x30008 in the memory map (Table 19).

Memory Map

The memory map for the ORT8850 core is only part of the full memory map of the ORT8850 device. The ORT8850 is an ORCA Series4 based device and thus uses the system bus as a communication bridge. The ORT8850 core register map contained in this data sheet only covers the embedded ASIC core of the device, not the entire device. The system bus itself, and the generic FPGA memory map, are fully documented in the MPI/System Bus Application Note. As part of the system bus, the embedded ASIC core of an FPSC is located at address offset 0x30000. The ORT8850 embedded core is an eight-bit slave interface on the Series 4 system bus.

Each ORCA device contains a device ID. This device ID is unique to each ORCA device and can be used for device identification and assist in system debugging. The device ID is located at absolute address 0x00000 - 0x00003. The ORT8850H's device ID is 0xDC0123C0 and the ORT8850L's device ID is 0xDC0121C0. More information on the device ID and other Series 4 generic registers can be found in the MPI/System Bus Application Note.

The ORT8850 core registers are clocked by the reference clock SYS_CLK_P/N. If a clock is not provided to the reference clock, the registers will fail to operate.

The ORT8850 core registers do not check for parity on a write operation. On a read operation, no parity is generated, and a "0" is passed back to the initiating bus master interface on the parity signal line.

Registers Access and General Description

The memory map comprises three address blocks:

- Generic register block: ID, revision, scratch pad, lock and reset register.
- Device register block: control and status bits, common to the eight channels in each of the two quad interfaces.
- Channel register blocks: each of the four channels in both quads have an address block. The four address blocks in both quads have the same structure, with a constant address offset between channel register blocks.

All registers are write-protected by the lock register, except for the scratch pad register. The lock register is a 16-bit read/write register. Write access is given to registers only when the key value 0x0580 is present in the lock register. An error flag will be set upon detecting a write access when write permission is denied. The default value is 0x0000.

After power-up reset or soft reset, unused register bits will be read as zeros. Unused address locations are also read as zeros. Bit in write-only registers will always be read as zeros.

This table is constructed to show the correct values when read and written via the system bus MPI interface. **When using this table while interfacing with the system bus user logic master interface, the data values will need to be byte flipped.** This is due to the opposite orientation of the MPI and master interface bus ordering. More information on this can be found in the MPI/System Bus Application Note (TN1017).

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30037* 3004F 30067 3007F 30097 300AF 300C7 300DF	[0]	R/W	Bypass pointer mover	0	0 = use pointer mover 1 = Bypass pointer mover.
	[1]	R/W	Bypass pointer mover and alignment FIFO	0	0 = uses alignment FIFO and pointer mover 1 = Bypass alignment FIFO and pointer mover.
	[2]		Enable work/protect channels	0	Bit to control the LVDS receivers to CDR. 0 = Use LVDS receivers from HSI work channels. 1 = Use LVDS receivers from HSI protect channels.
	[3:4]	R	Multichannel alignment control	00	00 = No alignment. 10 = Align with twin (i.e., STM B stream A). 01 = Align with all 4 (i.e., STM A all streams). 11 = Align with all 8 (i.e., STM A and B all streams).
	[5]	R	RX path SONET framer	0	0 = Enable framer. 1 = Disable SONET framing data is passed through
	[6-7]	-	Not Used	0	
300E0	[0]	R/W	Reserved	0	Reserved, must be set to 0.
	[1]	R/W	CDR control register	0	Always set to zero
	[2]	-	Not Used	0	
	[3]	R/W	CDR control register	0	When set to 1, controls bypass of 16 PLL generated phases with 16 low-speed phases.
	[4]	R/W	CDR control register	0	Enables CDR loopback. 0 = No loopback. 1 = Loopback TX to RX.
	[5]	R/W	CDR control register	0	Enables bypassing of the internal 622 MHz clock with TSTCLK. Must be used for simulation 0 = Use PLL. 1 = Bypass PLL (uses TSTCLK as reference clock).
	[6]	R/W	CDR control register	0	Enables CDR test mode. Initiates CDR's built-in self-test: 0 = Regular mode. 1 = Test mode.
	[7]	-	Not Used	0	
300E1	[0:7]	R/W	Half Rate		Per Channel select for half rate mode can only be used in pure bypass mode. Bit 7 is for channel BD, bit 6 is for BC etc. 0 = full rate 1 = half rate
300E2	[0:7]	R/W	Quad Rate		Per Channel select for quad rate mode can only be used in pure bypass mode. Bit 7 is for channel BD, bit 6 is for BC etc. 0 = full rate 1 = quad rate

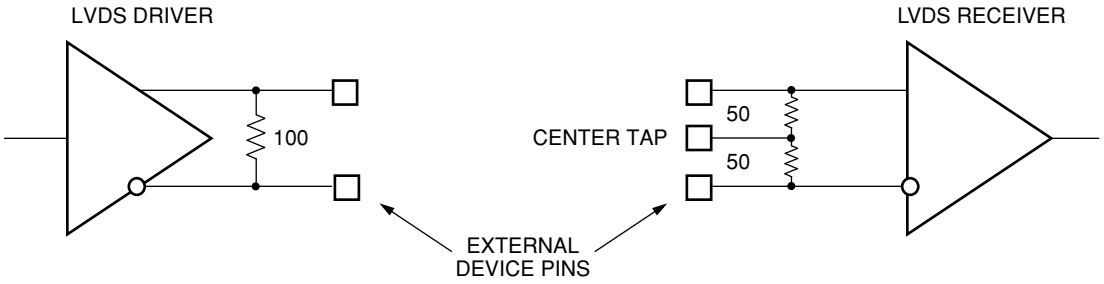
* For Channels AA, AB, AC, AD, BA, BB, BC, BD respectively

Note: Registers at addresses ≥ 300E3 must remain at their default (reset) settings and must not be changed by the user.

Termination Resistor

The LVDS drivers and receivers operate on a 100 Ω differential impedance, as shown below. External resistors are not required. The differential driver and receiver buffers include termination resistors inside the device package, as shown in Figure 36 below.

Figure 36. LVDS Driver and Receiver and Associated Internal Components



LVDS Driver Buffer Capabilities

Under worst-case operating condition, the LVDS driver must withstand a disabled or unpowered receiver for an unlimited period of time without being damaged. Similarly, when its outputs are short-circuited to each other or to ground, the LVDS driver will not suffer permanent damage Figure 37 illustrates the terms associated with LVDS driver and receiver pairs.

Figure 37. LVDS Driver and Receiver

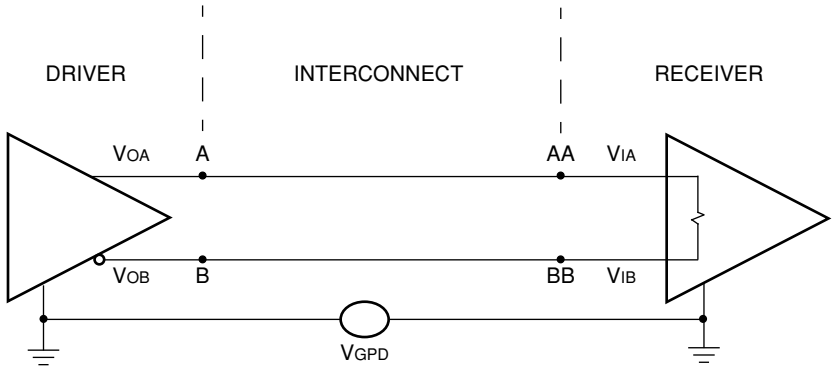


Figure 38. LVDS Driver

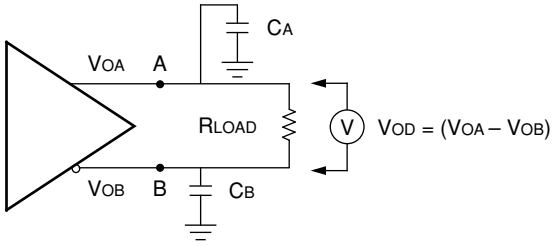


Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
N5	7 (CL)	1	IO	PL9C	PL16C	D4	L3T_D3
AM22	—	—	VSS	VSS	VSS	—	—
L2	7 (CL)	2	IO	PL9B	PL17D	—	L4C_D1
N4	7 (CL)	2	IO	PL9A	PL17C	—	L4T_D1
P5	7 (CL)	2	IO	PL10D	PL18D	RDY/BUSY_N/RCLK	L5C_D2
M2	7 (CL)	2	IO	PL10C	PL18C	VREF_7_02	L5T_D2
M3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
M1	7 (CL)	2	IO	PL10B	PL19D	A13/PPC_A27	L6C_D2
P4	7 (CL)	2	IO	PL10A	PL19C	A12/PPC_A26	L6T_D2
N2	7 (CL)	3	IO	PL11D	PL20D	—	L7C_D0
P3	7 (CL)	3	IO	PL11C	PL20C	—	L7T_D0
AM32	—	—	VSS	VSS	VSS	—	—
R4	7 (CL)	3	IO	PL11B	PL21D	A11/PPC_A25	L8C_D2
N1	7 (CL)	3	IO	PL11A	PL21C	VREF_7_03	L8T_D2
P2	7 (CL)	3	IO	PL12D	PL22D	—	L9C_A0
P1	7 (CL)	3	IO	PL12C	PL22C	—	L9T_A0
T4	7 (CL)	3	IO	PL12B	PL22B	—	L10C_D1
R2	7 (CL)	3	IO	PL12A	PL22A	—	L10T_D1
U5	7 (CL)	4	IO	PL13D	PL23D	RD_N/MPI_STRB_N	L11C_D3
R1	7 (CL)	4	IO	PL13C	PL23C	VREF_7_04	L11T_D3
AN1	—	—	VSS	VSS	VSS	—	—
V5	7 (CL)	4	IO	PL13B	PL23B	—	L12C_D1
T3	7 (CL)	4	IO	PL13A	PL23A	—	L12T_D1
T2	7 (CL)	4	IO	PL14D	PL24D	PLCK0C	L13C_A0
T1	7 (CL)	4	IO	PL14C	PL24C	PLCK0T	L13T_A0
R3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
U4	7 (CL)	4	IO	PL14B	PL24B	—	L14C_A0
U3	7 (CL)	4	IO	PL14A	PL24A	—	L14T_A0
AN2	—	—	VSS	VSS	VSS	—	—
U2	7 (CL)	5	IO	PL15D	PL25D	A10/PPC_A24	L15C_A0
V2	7 (CL)	5	IO	PL15C	PL25C	A9/PPC_A23	L15T_A0
AN33	—	—	VSS	VSS	VSS	—	—
V3	7 (CL)	5	IO	PL15B	PL25B	—	L16C_A0
V4	7 (CL)	5	IO	PL15A	PL25A	—	L16T_A0
W5	7 (CL)	5	IO	PL16D	PL26D	A8/PPC_A22	L17C_A2
W2	7 (CL)	5	IO	PL16C	PL26C	VREF_7_05	L17T_A2
W3	7 (CL)	5	IO	PL16B	PL27D	—	L18C_D1
Y1	7 (CL)	5	IO	PL16A	PL27C	—	L18T_D1
Y2	7 (CL)	6	IO	PL17D	PL28D	PLCK1C	L19C_D0
AA1	7 (CL)	6	IO	PL17C	PL28C	PLCK1T	L19T_D0
AN34	—	—	VSS	VSS	VSS	—	—
Y5	7 (CL)	6	IO	PL17B	PL29D	VREF_7_06	L20C_D3

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
T32	—	—	O	TXDAA_W_N	TXDAA_W_N	—	L32N_D1
R34	—	—	O	TXDAA_W_P	TXDAA_W_P	—	L32P_D1
AM33	—	—	VDD33	VDD33	VDD33	—	—
U30	—	—	O	TXDAB_W_N	TXDAB_W_N	—	L33N_D0
T31	—	—	O	TXDAB_W_P	TXDAB_W_P	—	L33P_D0
V17	—	—	VSS	VSS	VSS	—	—
R33	—	—	O	TXDAC_W_N	TXDAC_W_N	—	L34N_D0
P34	—	—	O	TXDAC_W_P	TXDAC_W_P	—	L34P_D0
AM34	—	—	VDD33	VDD33	VDD33	—	—
P33	—	—	O	TXDAD_W_N	TXDAD_W_N	—	L35N_D0
N34	—	—	O	TXDAD_W_P	TXDAD_W_P	—	L35P_D0
V18	—	—	VSS	VSS	VSS	—	—
T30	—	—	O	Reserved	Reserved	—	L36N_D0
R31	—	—	O	Reserved	Reserved	—	L36P_D0
AN32	—	—	VDD33	VDD33	VDD33	—	—
P32	—	—	O	Reserved	Reserved	—	L37N_D1
R30	—	—	O	Reserved	Reserved	—	L37P_D1
V19	—	—	VSS	VSS	VSS	—	—
N33	—	—	O	TXDBA_W_N	TXDBA_W_N	—	L38N_D0
M34	—	—	O	TXDBA_W_P	TXDBA_W_P	—	L38P_D0
AP32	—	—	VDD33	VDD33	VDD33	—	—
P31	—	—	O	TXDBB_W_N	TXDBB_W_N	—	L39N_D1
M33	—	—	O	TXDBB_W_P	TXDBB_W_P	—	L39P_D1
V34	—	—	VSS	VSS	VSS	—	—
N31	—	—	O	TXDBC_W_N	TXDBC_W_N	—	L40N_D0
P30	—	—	O	TXDBC_W_P	TXDBC_W_P	—	L40P_D0
L33	—	—	O	TXDBD_W_N	TXDBD_W_N	—	L41N_D0
K34	—	—	O	TXDBD_W_P	TXDBD_W_P	—	L41P_D0
W16	—	—	VSS	VSS	VSS	—	—
M31	—	—	I	Reserved	Reserved	—	L42N_D0
L32	—	—	I	Reserved	Reserved	—	L42P_D0
K33	—	—	I	Reserved	Reserved	—	—
W17	—	—	VSS	VSS	VSS	—	—
N30	—	—	VDDA_PDI	Reserved	Reserved	—	—
L30	—	—	VSSA_PDI	Reserved	Reserved	—	—
W18	—	—	VSS	VSS	VSS	—	—
M30	—	—	I	Reserved	Reserved	—	L43N_D0
L31	—	—	I	Reserved	Reserved	—	L43P_D0
W19	—	—	VSS	VSS	VSS	—	—
J34	—	—	I	Reserved	Reserved	—	L44N_D1
K32	—	—	I	Reserved	Reserved	—	L44P_D1
J33	—	—	I	Reserved	Reserved	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
A29	—	—	O	Reserved	Reserved	—	L56P_D0
D27	—	—	O	Reserved	Reserved	—	L57N_D0
E26	—	—	O	Reserved	Reserved	—	L57P_D0
C27	—	—	O	Reserved	Reserved	—	L58N_D0
D26	—	—	O	Reserved	Reserved	—	L58P_D0
A28	—	—	O	Reserved	Reserved	—	L59N_D0
B27	—	—	O	Reserved	Reserved	—	L59P_D0
C26	—	—	O	Reserved	Reserved	—	L60N_D0
D25	—	—	O	Reserved	Reserved	—	L60P_D0
A27	—	—	O	Reserved	Reserved	—	L61N_D0
B26	—	—	O	Reserved	Reserved	—	L61P_D0
D24	—	—	O	Reserved	Reserved	—	L62N_D0
C25	—	—	O	Reserved	Reserved	—	L62P_D0
C22	—	—	VSS	VSS	VSS	—	—
A26	1 (TC)	1	IO	PT26D	PT35D	—	L1C_D3
E25	1 (TC)	1	IO	PT26C	PT35C	—	L1T_D3
A25	1 (TC)	1	IO	PT26B	PT35B	—	L2C_A0
B25	1 (TC)	1	IO	PT26A	PT35A	—	L2T_A0
C24	1 (TC)	1	IO	PT25D	PT34D	VREF_1_01	L3C_D0
D23	1 (TC)	1	IO	PT25C	PT34C	—	L3T_D0
C32	—	—	VSS	VSS	VSS	—	—
B24	1 (TC)	1	IO	PT25B	PT33D	—	L4C_A2
E24	1 (TC)	1	IO	PT25A	PT33C	—	L4T_A2
D22	1 (TC)	2	IO	PT24D	PT32D	—	L5C_D1
B23	1 (TC)	2	IO	PT24C	PT32C	VREF_1_02	L5T_D1
E23	1 (TC)	2	IO	PT24B	PT31D	—	L6C_A3
A23	1 (TC)	2	IO	PT24A	PT31C	—	L6T_A3
D21	1 (TC)	2	IO	PT23D	PT30D	—	L7C_D1
B22	1 (TC)	2	IO	PT23C	PT30C	—	L7T_D1
D4	—	—	VSS	VSS	VSS	—	—
A22	1 (TC)	3	IO	PT22D	PT29D	—	L8C_D1
C21	1 (TC)	3	IO	PT22C	PT29C	VREF_1_03	L8T_D1
E22	1 (TC)	3	IO	PT22A	PT29A	—	—
D20	1 (TC)	3	IO	PT21D	PT28D	—	L9C_D1
B21	1 (TC)	3	IO	PT21C	PT28C	—	L9T_D1
D31	—	—	VSS	VSS	VSS	—	—
E21	1 (TC)	3	IO	PT21A	PT28A	—	—
A21	1 (TC)	3	IO	PT20D	PT27D	—	L10C_D0
B20	1 (TC)	3	IO	PT20C	PT27C	—	L10T_D0
A11	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
A20	1 (TC)	3	IO	PT20A	PT27A	—	—
E20	1 (TC)	4	IO	PT19D	PT26D	—	L11C_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
D19	1 (TC)	4	IO	PT19C	PT26C	—	L11T_D0
C19	1 (TC)	4	IO	PT19B	PT25D	—	L12C_A0
B19	1 (TC)	4	IO	PT19A	PT25C	—	L12T_A0
N3	—	—	VSS	VSS	VSS	—	—
E19	1 (TC)	4	IO	PT18D	PT24D	—	L13C_D0
D18	1 (TC)	4	IO	PT18C	PT24C	VREF_1_04	L13T_D0
A17	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
B18	1 (TC)	4	IO	PT18B	PT24B	—	L14C_A0
C18	1 (TC)	4	IO	PT18A	PT24A	—	L14T_A0
B17	1 (TC)	5	IO	PT17D	PT23D	PTCK1C	L15C_A0
C17	1 (TC)	5	IO	PT17C	PT23C	PTCK1T	L15T_A0
N13	—	—	VSS	VSS	VSS	—	—
A16	1 (TC)	5	IO	PT17B	PT23B	—	L16C_D2
D17	1 (TC)	5	IO	PT17A	PT23A	—	L16T_D2
B16	1 (TC)	5	IO	PT16D	PT22D	PTCK0C	L17C_A0
C16	1 (TC)	5	IO	PT16C	PT22C	PTCK0T	L17T_A0
D16	1 (TC)	5	IO	PT16A	PT22A	—	—
E18	1 (TC)	5	IO	PT15D	PT21D	VREF_1_05	L18C_D3
A15	1 (TC)	5	IO	PT15C	PT21C	—	L18T_D3
A19	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
B15	1 (TC)	5	IO	PT15A	PT21A	—	—
D15	1 (TC)	6	IO	PT14D	PT20D	—	L19C_D2
A14	1 (TC)	6	IO	PT14C	PT20C	—	L19T_D2
N14	—	—	VSS	VSS	VSS	—	—
B14	1 (TC)	6	IO	PT14A	PT20A	—	—
E17	1 (TC)	6	IO	PT13D	PT19D	—	L20C_D2
C14	1 (TC)	6	IO	PT13C	PT19C	VREF_1_06	L20T_D2
D14	1 (TC)	6	IO	PT13A	PT19A	—	—
N15	—	—	VSS	VSS	VSS	—	—
E16	0 (TL)	1	IO	PT11D	PT18D	MPI_RTRY_N	L1C_D3
A13	0 (TL)	1	IO	PT11C	PT18C	MPI_ACK_N	L1T_D3
B13	0 (TL)	1	IO	PT11B	PT17D	—	L2C_D0
A12	0 (TL)	1	IO	PT11A	PT17C	VREF_0_01	L2T_D0
B12	0 (TL)	1	IO	PT10D	PT16D	M0	L3C_D1
D13	0 (TL)	1	IO	PT10C	PT16C	M1	L3T_D1
A34	—	—	VSS	VSS	VSS	—	—
E15	0 (TL)	2	IO	PT10B	PT15D	MPI_CLK	L4C_D3
B11	0 (TL)	2	IO	PT10A	PT15C	A21/MPI_BURST_N	L4T_D3
A10	0 (TL)	2	IO	PT9D	PT14D	M2	L5C_D3
E14	0 (TL)	2	IO	PT9C	PT14C	M3	L5T_D3
A3	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
D12	0 (TL)	2	IO	PT9B	PT13D	VREF_0_02	L6C_D0