Lattice Semiconductor Corporation - ORT8850H-2BMN680C Datasheet

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	16192
Total RAM Bits	151552
Number of I/O	297
Number of Gates	899000
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/ort8850h-2bmn680c

ispLEVER Development System

The ispLEVER development system is used to process a design from a netlist to a configured FPGA. This system is used to map a design onto the *ORCA* architecture and then place and route it using ispLEVER's timing-driven tools. The development system also includes interfaces to, and libraries for, other popular CAE tools for design entry, synthesis, simulation, and timing analysis.

The ispLEVER development system interfaces to front-end design entry tools and provides the tools to produce a configured FPGA. In the design flow, the user defines the functionality of the FPGA at two points in the design flow, the design entry and the bit stream generation stage. Recent improvements in ispLEVER allow the user to provide timing requirement information through logical preferences only; thus, the designer is not required to have physical knowledge of the implementation.

Following design entry, the development system's map, place, and route tools translate the netlist into a routed FPGA. A floor planner is available for layout feedback and control. A static timing analysis tool is provided to determine design speed, and a back-annotated netlist can be created to allow simulation and timing.

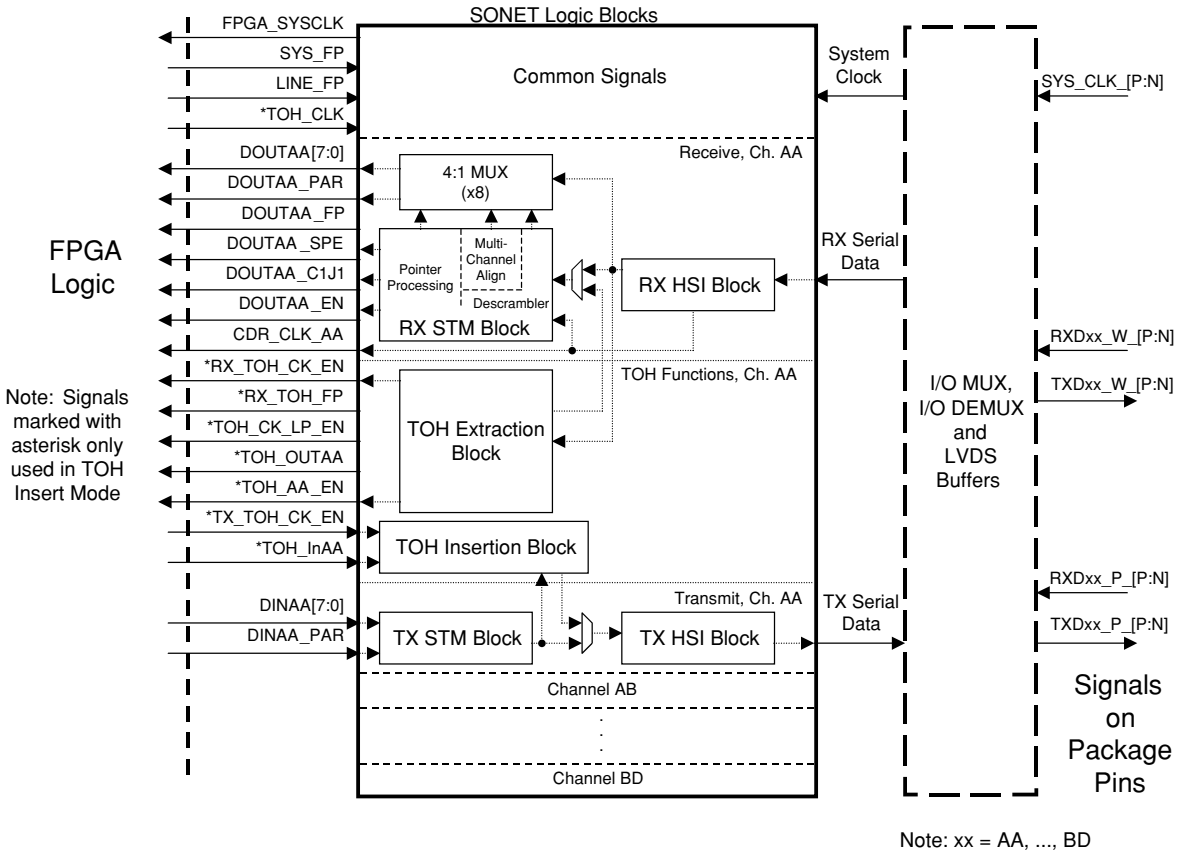
Timing and simulation output files from ispLEVER are also compatible with many third-party analysis tools. A bit stream generator is then used to generate the configuration data which is loaded into the FPGAs internal configuration RAM, embedded block RAM, and/or FPSC memory.

When using the bit stream generator, the user selects options that affect the functionality of the FPGA. Combined with the front-end tools, ispLEVER produces configuration data that implements the various logic and routing options discussed in this data sheet.

FPSC Design Kit

Development is facilitated by an FPSC design kit which, together with ispLEVER software and third-party synthesis and simulation engines, provides all software and documentation required to design and verify an FPSC implementation. Included in the kit are the FPSC configuration manager, *Synopsys Smart Model*®, and/or compiled *Verilog*® simulation model, *HSPICE*® and/or IBIS models for I/O buffers, and complete online documentation. The kit's software couples with ispLEVER software, providing a seamless FPSC design environment. More information can be obtained by visiting the Lattice website at www.latticesemi.com or contacting a local sales office.

Figure 3. Top Level Block Diagram ORT8850 Embedded Core SONET Logic Block Common Signals and Channel AA Data Flow



Each quad can frame independently in STS-3, STS-12 or STS-48 format. If using STS-48 format all channels in the quad will be used and be treated as a single STS-48 channel using the quad STS-12 format in which each independent channel carries entire STS-12 frames. The byte order for STS-48 must be created by the designer in the FPGA design. Note that the recovered data will always continue to be in the same order as transmitted data.

Each channel contains transmit path and receive path logic, both of which are organized around High Speed Interconnect (HSI) and Synchronous Transport Mode (STM) macrocells. Additional logic allows insertion and extraction of information in the Transport Overhead area of the SONET frame. (Support for loopback and for switching between redundant serial links is also provided but is not shown in Figure 3). The following sections will give an overview of the pseudo-SONET protocol supported by the ORT8850 and a top level overview of the Synchronous Transport Module (STM) and High Speed Interconnect (HSI) macrocells, which provide the SONET functionality.

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.

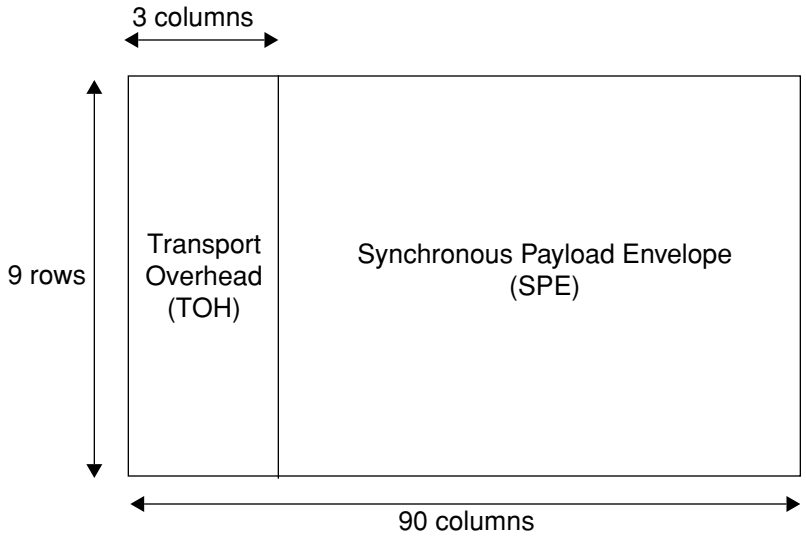
The data rates for SONET are covered in the following table. Values that fall in between those shown in the table for each mode are supported (126.00 Mbits/s - 212.50 Mbits/s, 504.00 Mbits/s - 850.00 Mbits/s). 63.00 MHz is the slowest reference clock while 106.25 MHz is the fastest reference clock frequency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



output. Although all TOH bytes from the 12 STS-1s are transferred into the device from each serial port, not all of them get inserted in the frame. There are three hard coded exceptions to the TOH byte insertion:

- Framing bytes (A1/A2 of all STS-1s) are not inserted from the serial input bus. Instead, they can always be regenerated.
- Parity byte (B1 of STS#1) is not inserted from the serial input bus. Instead, it is always recalculated (the 11 bytes following B1 are replaced with all zeros).
- Pointer bytes (H1/H2/H3 of all STS-1s) are not inserted from the serial input bus. Instead, they always flow transparently from parallel input to LVDS output.

The data stream is scrambled in the transmit direction and descrambled in the receive direction using a frame synchronous scrambler of sequence length 127, operating at the line rate. The generating polynomial for the scrambler is $1+x^6+x^7$. The polynomial conforms to the standard SONET STS-12 data format. The scrambler is reset to '1111111' on the first byte of the SPE (byte following the Z0 byte in the 12th STS-1). That byte and all subsequent bytes to be scrambled are XOR'd, with the output from the bytewise scrambler. The scrambler runs continuously from that byte, through the remainder of the frame. A1, A2, and J0/Z0 bytes are not scrambled. The B1 byte is calculated (in both transmitter and receiver) on the non-scrambled data. There is a global scrambler/descrambler disable feature, allowing the user to disable the scrambler of the transmitter and the descrambler of the receiver. Following the scrambler block, byte wide data streams are sent to the HSI macrocell.

Receive STM Macrocell Logic - Overview

In the receive direction (backplane to the FPGA interface) each STM macrocell receives four byte wide data streams at the reference clock rate (i.e., 8 X SYS_CLK_[P:N] in normal operation) and four associated clocks from the HSI. The incoming streams are framed and (optionally) descrambled before they are written into a FIFO which absorbs phase and delay variations and allows the shift to system clock and optionally allows frames to be aligned both between quads and between streams on the same quad. Optionally, the pointer interpreter logic will then put the STS SPEs into a small elastic store from which the pointer generator will produce four byte wide STS-12 streams of data that are aligned to the system timing pulse.

The alignment FIFO depth allows for 18 clocks of difference in the arriving A1/A2. If any of the channels in an alignment group are too far out of alignment for the FIFO to absorb the difference an alarm register will indicate the error. Alarm indicators can be programmed to trigger an alarm at different levels of misalignment.

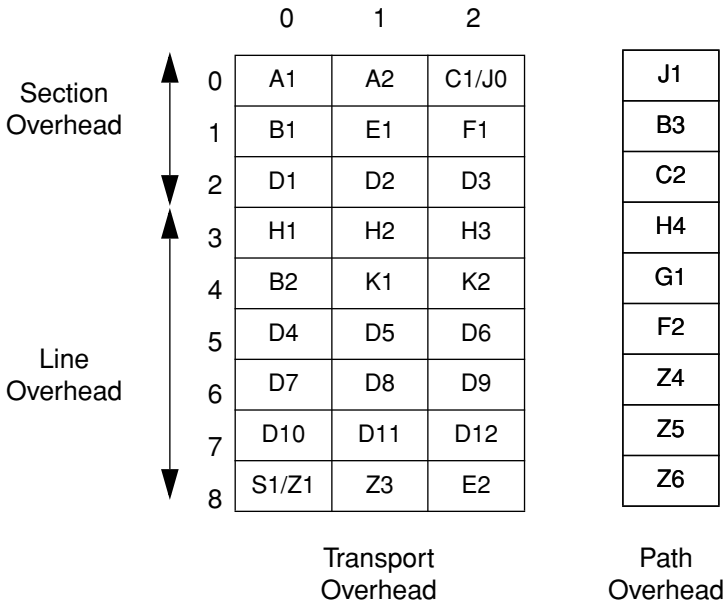
The multichannel alignment option allows separate SERDES data channels to be byte aligned based on the SONET A1/A2 bytes. Data is written into the alignment FIFO using the per channel recovered clocks from the SERDES channel. Data is always read from the alignment FIFO using the local reference clock. (SYS_CLK pin, FPGA_SYS CLK)

SERDES data channels can be placed into an alignment group by 2, by 4, or all 8. In by 2 mode, channels AA and BA, AB and BB, AC and BC, and AD and BD are byte aligned. In by 4 mode channels AA, AB, AC, AD and BA, BB, BC, BD are byte aligned. In the by 8 mode all of the channels are byte aligned.

After the alignment FIFO the receive data can optionally go through the pointer interpreter and pointer mover. The pointer interpreter will identify the SONET payload envelope (SPE) and the C1(J0) bytes and the J1 bytes. For data applications where the user is simply using SONET to carry user defined cells in the payload the SPE signal is very useful as an enable to the cell processor. C1J1 for data applications can be ignored. If the pointer interpreter and pointer mover are bypassed, then the SPE and C1J1 signals to the FPGA logic will be always '0'.In the ORT8850 each frame consists of 12 STS-1 format sub-frames. Thus, in the SPE region, there are 12 J1 pulses, one for each STS-1. There is one C1(J0) (current SONET specifications use J0 instead of C1 as section trace to identify each STS-1 in an STS-N) pulse in the TOH area for one frame. Thus, there are a total of 12 J1 pulses and one C1(J0) pulse per frame. The C1(J0) pulse is coincident with the J0 of STS-1 #1 which is the first byte following the last A2 byte.

With the pointer interpreter option enabled, the SPE flag is active when the data stream is in SPE area. SPE behavior is dependent on pointer movement and concatenation. Note: in the TOH area, H3 can also carry valid

Figure 13. SONET Overhead Bytes



When used in true SONET applications, most TOH bytes would be generated in the FPGA logic or by an external device. The TOH bytes have the following functions. Table 9 and Table 10 show how the Embedded Core modifies these bytes in the transmit direction and Table 12 shows how the bytes are modified in the receive direction.

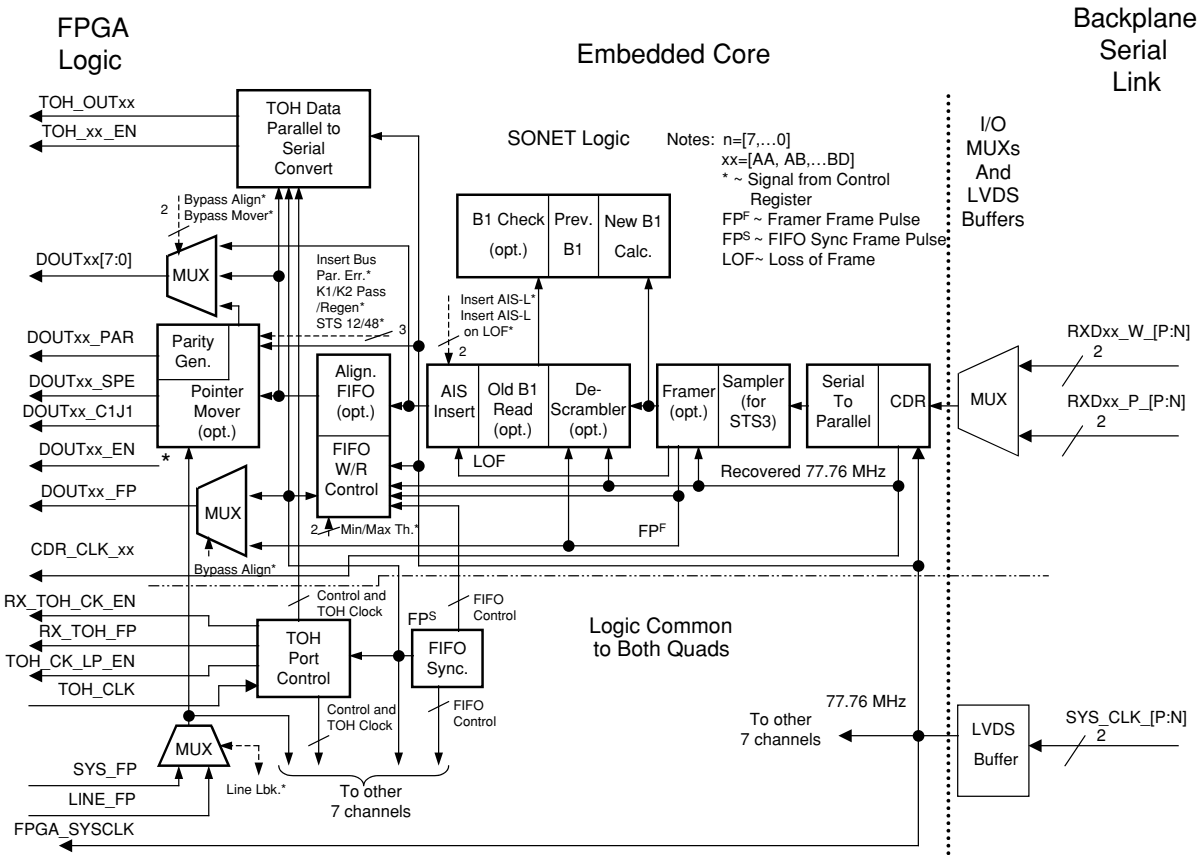
Section Overhead Bytes:

- A1, A2 - These bytes are used for framing and to mark the beginning of a SONET frame. A1 has the value 0xF6 and A2 has the value 0x28.
- C1/J0 - Section Trace Message - This byte carries the section trace message. The message is interpreted to verify connectivity to a particular node in the network.
- B1 - Section Bit Interleaved Parity (BIP-8) byte - This byte carries the parity information which is used to check for transmission errors in a section. The computed parity value is transmitted in the next frame in the B1 position. It is defined only for the first STS-1 of a STS-N signal. The other bytes have a default value of 0x00 if using serial TOH insertion. In transparent TOH mode the other bytes are passed through from DINxx bus.
- E1 - Section orderwire byte - This byte carries local orderwire information, which provides for a 64 Kbits/s voice channel between two Section Termination Equipment (STE) devices.
- F1 - Section user channel byte - This byte provides a 64 Kbits/s user channel which can be used in a proprietary fashion.
- D1, D2, D3 - Section Data Communications Channel (SDCC) bytes - These bytes provide a 192 Kbits/s channel for transmission of information across STEs. This information could be for control and configuration, status monitoring, alarms, network administration data etc.

Line Overhead Bytes:

- H1, H2 - STS Payload Pointers (H1 and H2) - These bytes are used to locate the start of the SPE in a SONET frame. These two bytes contain the offset value, in bytes, between the pointer bytes and the start of the SPE. These bytes are used for all the STS-1 signals contained in an STS-N signal to indicate the individual starting positions of the SPEs. They bytes also contain justification indications, concatenation indications and path alarm indication (AIS-P).
- H3 - Pointer Action Byte (H3) - This byte is used during frequency justifications. When a negative justification is

Figure 16. Basic Logic Blocks, Receive Path, Single Channel



HSI Functions (Clock Recovery and Deserializer)

The HSI receive path functions include Clock and Data Recovery (CDR) and deserialization of the incoming data from the selected work or protect input stream to the byte-wide internal data bus format. The serial data received from the LVDS buffer does not have an accompanying clock. Based on data transitions, the receiver selects an appropriate internal clock phase for each channel to retime the data. The retimed data and clock are then passed to the DEMUX (deserializer) module. The DEMUX module performs serial-to-parallel conversion and provides parallel data and clock to the SONET framer block. For a 622 Mbits/s SONET stream, the HSI will perform Clock and Data Recovery (CDR) and MUX/DEMUX between 77.76 MHz byte-wide internal data buses and 622 Mbits/s serial LVDS links.

Sampler

This block operates on the byte-wide data directly from the HSI macro. The HSI external interface always runs at 622 Mbits/s (STS-12), or 850 Mbits/s, but it can be connected directly to a 155 Mbits/s STS-3 stream. If connected to a 155 Mbits/s stream, each incoming bit is received four times. This block is used to return the byte stream to the expected STS-12 format. The mode of operation is controlled by a register and can either be STS-12 (pass-through) or STS-3. The output from this block is not bit aligned (i.e., an 8-bit sample does not necessarily contain an entire SONET byte), but it is in standard SONET STS-12 format (i.e., four STS-3s) and is suitable for framing.

SONET Framer Block

The framer block takes byte-wide data from the HSI, and outputs a byte-aligned, byte-wide data stream and 8 kHz sync pulse. The framer algorithm determines the out-of-frame/in-frame status of the incoming data and will set alarm register bits on both an errored frame and an Out-Of-Frame (OOF) state.

The framer block takes byte wide data from the HSI, and outputs a byte aligned byte wide stream and 8 kHz sync pulse asserted coincident the first A1 byte which will be used by following blocks. (Note however that if the pointer

Pointer Interpreter and Pointer Mover

After the alignment FIFO the receive data can optionally go through the pointer interpreter and pointer mover. The pointer interpreter will identify the SONET payload envelope (SPE), the C1 bytes and the J1 bytes, and provide this information to the FPGA logic. For data applications where the user is simply using SONET to carry user defined cells in the payload the SPE signal is very useful as an enable to the cell processor. C1J1 for generic data applications can be ignored. If the pointer interpreter and pointer mover are bypassed, then the SPE and C1J1 signals will be always '0'.

Since the start of an SPE can be located at any point in a SONET frame, the starting point is identified using pointer bytes H1 and H2. The pointer bytes indicate the offset of the start of the SPE from the pointer byte position. Two payload pointer bytes (H1 and H2) are allocated to a pointer that indicates the offset in bytes between the pointer and the first byte of the STS SPE. The pointer bytes are used in all STS-1s within an STS-N to align the STS-1 Transport Overhead in the STS-N, and to perform frequency justification. These bytes are also used to indicate concatenation, and to detect Alarm Indication Signals (AIS).

The resulting 2 byte pointer is divided into three parts:

1. Four bits of New Data Flag (NDF)
2. Two bits of unassigned bits (These bits are set to 00.)
3. Ten bits for pointer value, which are alternately considered increment (I) bits or decrement (D) bits. The 10 bit pointer is required to represent the maximum SPE offset of 782 (9 rows * 87 columns - 1). Specific combinations of pointer byte values indicate that positive or negative frequency justification will occur and also whether or not the current frame is a concatenated frame.

Normally the NDF bits are set to 0110, which indicates that the current pointer values are unchanged. The inverse bit pattern, 1001, indicates that some data has changed. Any other bit configuration is interpreted using the 3 of 4 rule, i.e., 1110 is interpreted as 0110, etc. Patterns that cannot be resolved are undefined, however all one's in the NDF and in the pointer bits indicates AIS detection.The ORT8850 can correctly process any length of concatenation of STS frames (multiple of three) as long as it begins on an STS-3 boundary (i.e., STS-1 number one, four, seven, ten, etc.) and is contained within the smaller of STS-3, 12, or 48.

Table 13. Valid Starting Positions for and STS MC

STS-1 Number	STS-3cSPE	STS-6cSPE	STS-9cSPE	STS-12cSPE	STS-15cSPE	STS-18c to STS-48c SPEs
1	Yes	Yes	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	No	Yes	—
7	Yes	Yes	No	No	Yes	—
10	Yes	No	No	No	Yes	—
13	Yes	Yes	Yes	Yes	Yes	—
16	Yes	Yes	Yes	No	Yes	—
19	Yes	Yes	No	No	Yes	—
22	Yes	No	No	No	Yes	—
25	Yes	Yes	Yes	Yes	Yes	—
28	Yes	Yes	Yes	No	Yes	—
31	Yes	Yes	No	No	Yes	—
34	Yes	No	No	No	Yes	No
37	Yes	Yes	Yes	Yes	No	No
40	Yes	Yes	Yes	No	No	No
43	Yes	Yes	No	No	No	No

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

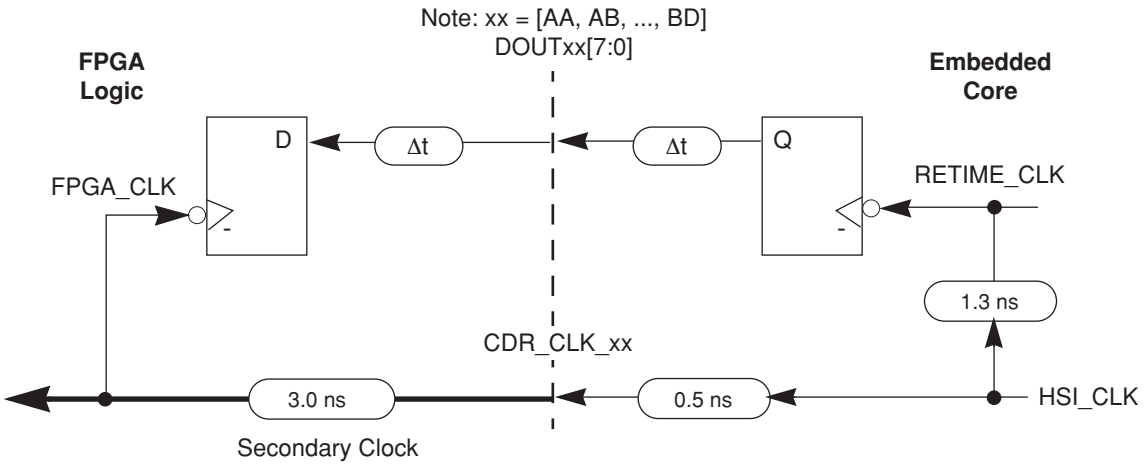
Table 15. FPGA/Embedded Core Interface Signals

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)

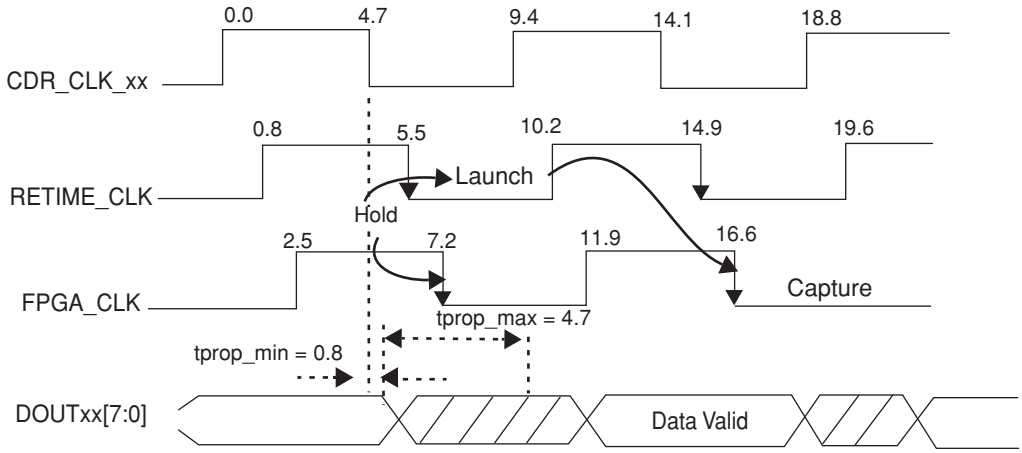
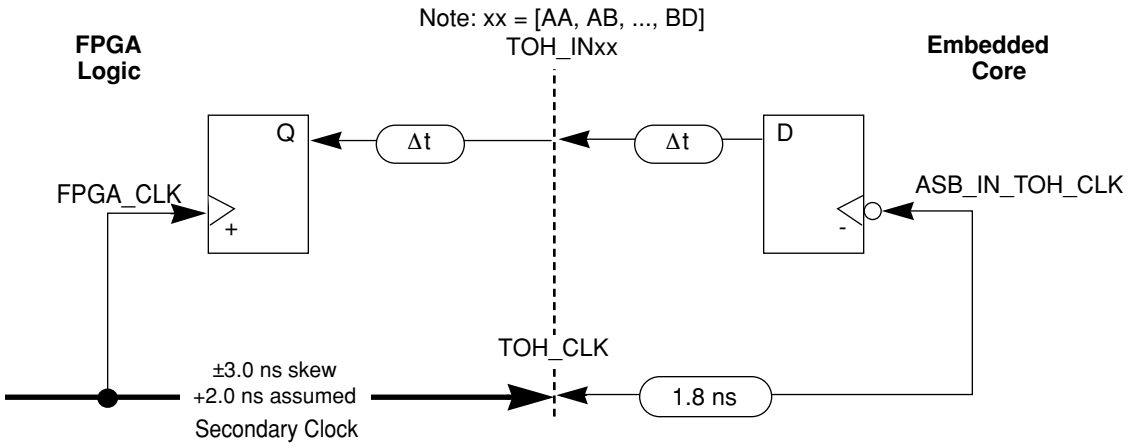


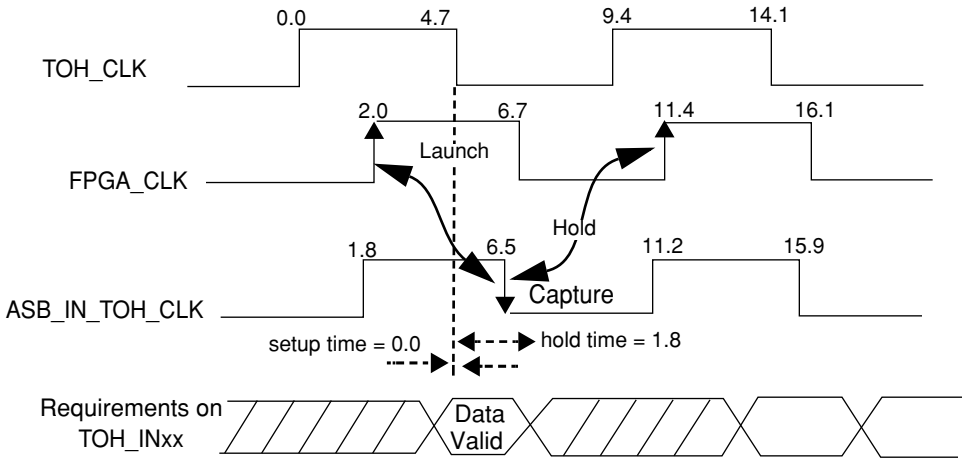
Figure 31 shows the timing for sending TOH data from the FPGA logic to the Core. As in the earlier input example, the constraints on the data are specified in terms of setup and hold times on the data at the interface relative to the clock at the interface. In the case shown, launch and capture occur on different clock edges (rising edge in the FPGA). 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 for the timing relationships assumed in the example. Actual timing analysis should be performed for each application because of the wide range of possible skew values.

Figure 31. Half Cycle, TOH Input Configuration and Timing (-1 Speed Grade)

a. Configuration



b. Timing (ns)



Powerdown Mode

Powerdown mode will be entered when the corresponding channel is disabled. Channels can be independently enabled or disabled under software control.

Parallel data bus output enable and TOH serial data output enable signals are made available to the FPGA logic. The HSI macrocell's corresponding channel is also powered down. The device will power up with all eight channels in powerdown mode.

Protection Switching

There is built-in protection switching between the SERDES channels, in the receive direction of the ORT8850. Protection switching allows pairs of SERDES channels to act as main and protect data links, and to switch between the main and protect links via a control register or FPGA interface port. There are two types of protection switches: parallel and LVDS.

Parallel protection switching takes place just before the FPGA interface ports, and after the alignment FIFO. The alignment FIFO must be used for this type of protection switching. It is possible to bypass the pointer interpreter/mover and still use the parallel protection switching. In this mode, SERDES channels AA and AB are used as main and protect. When selected for main, channel AA is used to provide data on interface ports AA. When selected for protect, channel AB is used to provide data on FPGA interface ports AA. The same scheme is used for channel groupings AC/AD, BA/BB, and BC/BD

There are two ways to control the parallel protection switching, interface signal and software control. On the FPGA interface, there are 4 input signals to the ORT8850 core that will select between a main and a protect channel. When using the interface signal to control protection switching, only the parallel data is switched; the serial TOH data outputs are not switched.

Software control will switch both the parallel data and the serial TOH data outputs to the FPGA. The software control register is found at 0x30009 in the memory map (Table 19).

Table 17. Register Settings, Parallel Protection Switching

FPGA Interface Signal	When '0'	When '1'
PROT_SWITCH_AA	Channel AB data on DOUTAA	Channel AA data on DOUTAA
PROT_SWITCH_AC	Channel AD data on DOUTAC	Channel AC data on DOUTAC
PROT_SWITCH_BA	Channel BB data on DOUTBA	Channel BA data on DOUTBA
PROT_SWITCH_BC	Channel BD data on DOUTBC	Channel BC data on DOUTBC

LVDS protection switching takes place at the LVDS buffer before the serial data is sent into the Data Recovery (CDR). The selection is between the main LVDS buffer and the protect LVDS buffer. The work LVDS buffers are TXDxx_W_[P:N], while the protect LVDS buffers are TXDxx_P_[P:N]. When operating using the LVDS buffers (default), no status information is available on the protect LVDS buffers since the serial stream must reach the SONET framer before status information is available on the data stream. The same is also true for the work LVDS buffers when operating with the protect buffers.

There are two ways to control the LVDS protection switching, interface and software control. On the FPGA interface, there are eight input signals to the ORT8850 core that will select between the work and protect LVDS buffers.

Table 18. LVDS Protection Switching

FPGA Interface Signal	When '0'	When '1'
LVDS_PROT_AA	Channel AA gets TXD_AA_W_[P:N]	Channel AA gets TXD_AA_P_[P:N]
LVDS_PROT_AB	Channel AB gets TXD_AB_W_[P:N]	Channel AB gets TXD_AB_P_[P:N]
LVDS_PROT_AC	Channel AC gets TXD_AC_W_[P:N]	Channel AC gets TXD_AC_P_[P:N]
LVDS_PROT_AD	Channel AD gets TXD_AD_W_[P:N]	Channel AD gets TXD_AD_P_[P:N]

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30017	[0]	R/W	BD resync	0	Channel BD alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[1]	R/W	BC resync	0	Channel BC alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[2]	R/W	BB resync	0	Channel BB alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[3]	R/W	BA resync	0	Channel BA alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[4]	R/W	AD resync	0	Channel AD alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[5]	R/W	AC resync	0	Channel AC alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[6]	R/W	AB resync	0	Channel AB alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[7]	R/W	AA resync	0	Channel AA alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
30018	[0]	R/W	AD/BD resync	0	2-link AD/BD alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[1]	R/W	AC/BC resync	0	2-link AC/BC alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[2]	R/W	AB/BB resync	0	2-link AB/BB alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[3]	R/W	AA/BA resync	0	2-link AA/BA alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[4]	R/W	STM B resync	0	Quad B alignment resync. Write “0” for normal operation.
	[5]	R/W	STM A resync	0	Quad A alignment FIFO resync Write “0” for normal operation.
	[6]	R/W	All 8 resync	0	All 8 channel alignment FIFO resync. Write “0” for normal operation.
	[7]	-	Not Used	N/A	

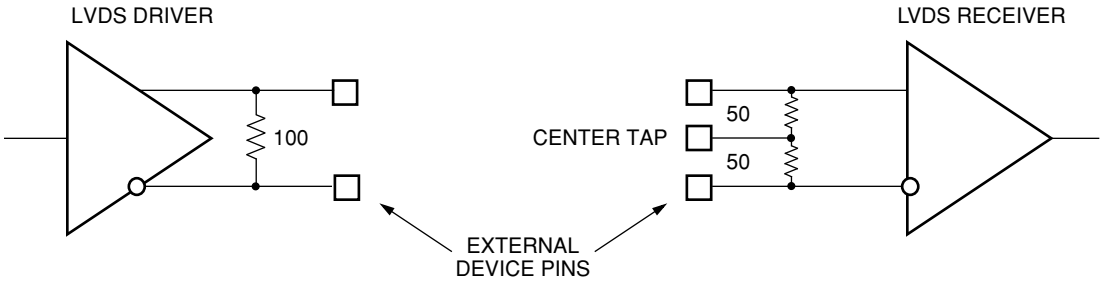
Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30027* 0303F 30057 3006F 30087 3009F 300B7 300CF	[0]	R/W	enable channel alarm	0	Channel alarm bit (30026, ...) enable. Set to 1 to enable alarm bit to propagate to alarm 0x30010
	[1]	R/W	enable AIS-P flag	0	AIS -P flag alarm enable. Set to 1 to enable alarm bit to propagate to alarm 0x30010
	[2]		enable pointer mover elastic store overflow flag	0	Pointer mover elastic store overflow flag enable. Set to 1 to enable alarm bit to propagate to 0x30010
	[3-7]	-	Not Used	0	
30028* 30040 30058 30070 30088 300A0 300B8 300D0	[0]	R	FIFO aligner threshold error flag	00	Alarm is set to 1 if either the min or max FIFO threshold levels are violated, the min and max threshold levels can be set in address 0x3000A and 0x300B. Alarm enable is 0x30029 bit 0. Write 1 to clear this alarm bit This alarm is only valid when FIFO OOS flag is also set.
	[1]		RX internal path parity error flag		Alarm indicator on receive path internal parity error. Alarm is enabled in 0x30029 bit 1. Write 1 to clear
	[2]		OOF flag		Alarm indicator channel is OOF. Alarm enable is 0x30029 bit 2. Write 1 to clear.
	[3]		LVDS link B1 parity error flag		Alarm indicator that channel has found a B1 parity error. Alarm enable is 0x30029 bit 3. Write 1 to clear.
	[4]		DINxx parallel bus parity error flag	0	Alarm indicator channel has found a parity error on the DINxx input from the FPGA.Alarm enable is 0x30029 bit 4. Write 1 to clear.
	[5]		TOH serial input port parity error flag	0	Alarm indicator channel has found a parity error on the TOH_INxx input from the FPGA. Write 1 to clear this alarm. Alarm enable is 0x30028 bit 5.
	[6]		FIFO OOS error flag	0	Alarm indicates channel group is out of sync. Write 1 to clear. Alarm enable is 0x30028.
	[7]	-	Not Used	0	
30029* 30041 30059 30071 30089 300A1 300B9 300D1	[0:6]	R/W	channel alarm enable	00	Enable bits for channel alarm register 0x30028. Set to 1 to enable and to propagate the alarm to register 0x30026 bit 0.
	[7]	-	Not Used	0	
3002A* 30042 3005A 30072 3008A 300A2 300BA 300D2	[0:3]	R	AIS alarm flags 3, 6, 9, 12	0	These are the AIS-P alarm flags. 1 if the LVDS input STS # contains AIS.
	[4-7]	-	Not Used	0	

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

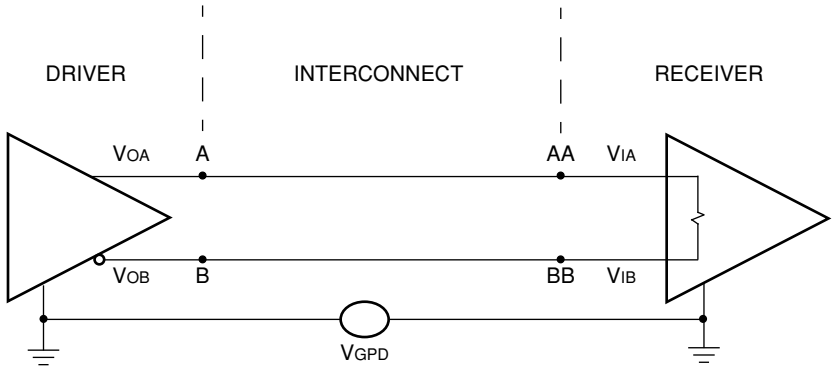


Figure 38. LVDS Driver

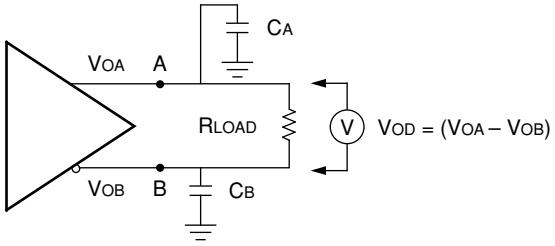


Table 32. FPGA Common-Function Pin Descriptions (Continued)

Symbol	I/O	Description
$\overline{\text{MPI\_BDIP}}$	I	$\overline{\text{MPI\_BDIP}}$ is driven by the *PowerPC* processor in MPI mode. Assertion of this pin indicates that the second beat in front of the current one is requested by the master. Negated before the burst transfer ends to abort the burst data phase.
MPI_TSZ[0:1]	I	MPI_TSZ[0:1] signals are driven by the bus master in MPI mode to indicate the data transfer size for the transaction. Set 01 for byte, 10 for half-word, and 00 for word.
A[21:0]	O	During master parallel mode A[21:0] address the configuration EPROMs up to 4M bytes.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_ACK	O	In MPI mode this is driven low indicating the MPI received the data on the write cycle or returned data on a read cycle.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_CLK	I	This is the *PowerPC* synchronous, positive-edge bus clock used for the MPI interface. It can be a source of the clock for the embedded system bus. If MPI is used this will be the *AMBA* bus clock.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_TEA	O	A low on the MPI transfer error acknowledge indicates that the MPI detects a bus error on the internal system bus for the current transaction.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_RTRY	O	This pin requests the MPC860 to relinquish the bus and retry the cycle.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
D[0:31]	I/O	Selectable data bus width from 8, 16, 32-bit in MPI mode. Driven by the bus master in a write transaction and driven by MPI in a read transaction.
	I	D[7:0] receive configuration data during master parallel, peripheral, and slave parallel configuration modes when $\overline{\text{WR}}$ is low and each pin has a pull-up enabled. During serial configuration modes, D0 is the DIN input.
	O	D[7:3] output internal status for asynchronous peripheral mode when $\overline{\text{RD}}$ is low.
	I/O	After configuration, if MPI is not used, the pins are user-programmable I/O pins.*
DP[0:3]	I/O	Selectable parity bus width in MPI mode from 1, 2, 4-bit, DP[0] for D[0:7], DP[1] for D[8:15], DP[2] for D[16:23], and DP[3] for D[24:31]. After configuration, if MPI is not used, the pins are user-programmable I/O pin.*
DIN	I	During slave serial or master serial configuration modes, DIN accepts serial configuration data synchronous with CCLK. During parallel configuration modes, DIN is the D0 input. During configuration, a pull-up is enabled.
	I/O	After configuration, this pin is a user-programmable I/O pin.*
DOUT	O	During configuration, DOUT is the serial data output that can drive the DIN of daisy-chained slave devices. Data out on DOUT changes on the rising edge of CCLK.
	I/O	After configuration, DOUT is a user-programmable I/O pin.*
TESTCFG	I	During configuration this pin should be held high, to allow configuration to occur. A pull up is enabled during configuration.
	I/O	After configuration, TESTCFG is a user programmable I/O pin.*
LVDS_R	—	Reference resistor connection for controlled impedance termination of Series 4 FPGA LVDS inputs.

1. The FPGA States of Operation section contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Package Information

Table 34 summarizes the programmable I/O clock and power pins available to the ORT8850 devices.

Table 34. ORT8850 IO and Power Pin Summary

I/O or Power Type	ORT8850L	ORT8850H
User I/O Single Ended	278	297
User I/O Differential Pairs (LVDS, LVPECL)	129	129
Configuration	7	7
Dedicated Function	3	3
VDD15	48	48
VDD33	28	28
VDDIO	38	38
Vss	89	89
Single-Ended/Differential I/O per Bank		
Bank 0	64/32	68/32
Bank 1	47/20	47/20
Bank 2	ASIC I/O	ASIC I/O
Bank 3	ASIC I/O	ASIC I/O
Bank 4	ASIC I/O	ASIC I/O
Bank 5	44/18	44/18
Bank 6	76/32	76/32
Bank 7	55/27	62/27

There are some incompatibilities between the ORT8850H and ORT8850L due to the fact that the ORT8850L is a much smaller array and hence does not provide as many programmable IOs (PIOs). In order to allow pin-for-pin compatible board layouts that can accommodate either device, key compatibility issues include the following:

- **Unused Pins** Table 35 shows a list of bonded ORT8850H PIOs that are unused in the ORT8850L. As shown in the table, there are 19 balls that are not available in the ORT8850L, but are available in the ORT8850H. These user I/Os should not be used if an ORT8850L will be used.
- **Shared Control Signals on I/O Registers.** The ORCA Series 4 architecture shares clock and control signals between two adjacent I/O pads. If I/O registers are used, incompatibilities may arise between ORT8850L and ORT8850H when different clock or control signals are needed on adjacent package pins. This is because one device may allow independent clock or control signals on these adjacent pins, while the other may force them to be the same. There are two ways to avoid this issue.
  - Always keep an open bonded pin (non-bonded pins for the ORT8850L do not count) between pins that require different clock or control signals. Note that this open pin can be used to connect signals that do not require the use of I/O registers to meet timing.
  - Place and route the design in both the ORT8850H and ORT8850L to verify both produce valid designs. Note that this method guarantees the current design, but does not necessarily guard against issues that can occur when design changes are made that affect I/O registers.
  - **2X/4X I/O Shift Registers.** If 2X I/O shift registers or 4X I/O shift registers are used in the design, this may cause incompatibilities between the ORT880L and ORT8850H because only the A and C I/Os in a PIC support 2X I/O shift registers and only A I/Os supports 4X I/O shift register mode. A and C I/Os are shown in the following pinout tables under the I/O pad columns as those ending in A or C.
- **Edge Clock Input Pins.** The input buffers for fast edge clocks are only available at the C I/O pad. The C I/Os are shown in the following pinout tables under the I/O pad columns as those ending in C.

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
AL19	5 (BC)	2	IO	PB17D	PB25D	—	L7C_A0
AP20	5 (BC)	3	IO	PB18A	PB26C	—	L8T_D3
AK19	5 (BC)	3	IO	PB18B	PB26D	VREF_5_03	L8C_D3
AM15	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AN20	5 (BC)	3	IO	PB18C	PB27A	—	—
Y21	—	—	VSS	VSS	VSS	—	—
AP21	5 (BC)	3	IO	PB19A	PB27C	—	L9T_D2
AL20	5 (BC)	3	IO	PB19B	PB27D	—	L9C_D2
Y22	—	—	VSS	VSS	VSS	—	—
AK20	5 (BC)	3	IO	PB19C	PB28A	—	—
AN21	5 (BC)	3	IO	PB20A	PB28C	PBCK1T	L10T_A0
AM21	5 (BC)	3	IO	PB20B	PB28D	PBCK1C	L10C_A0
AM20	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AK21	5 (BC)	3	IO	PB20C	PB29A	—	—
AP22	5 (BC)	4	IO	PB21A	PB29C	—	L11T_D2
AL21	5 (BC)	4	IO	PB21B	PB29D	—	L11C_D2
AA15	—	—	VSS	VSS	VSS	—	—
AN22	5 (BC)	4	IO	PB21C	PB30A	—	—
AP23	5 (BC)	4	IO	PB22A	PB30C	—	L12T_A0
AN23	5 (BC)	4	IO	PB22B	PB30D	VREF_5_04	L12C_A0
AA13	—	—	VSS	VSS	VSS	—	—
AK22	5 (BC)	4	IO	PB22C	PB31C	—	L13T_A0
AL22	5 (BC)	4	IO	PB22D	PB31D	—	L13C_A0
AN24	5 (BC)	5	IO	PB23C	PB32C	—	L14T_D2
AK23	5 (BC)	5	IO	PB23D	PB32D	VREF_5_05	L14C_D2
AA14	—	—	VSS	VSS	VSS	—	—
AL23	5 (BC)	5	IO	PB24C	PB33C	—	L15T_D0
AM24	5 (BC)	5	IO	PB24D	PB33D	—	L15C_D0
AP25	5 (BC)	5	IO	PB25A	PB34C	—	L16T_A0
AN25	5 (BC)	5	IO	PB25B	PB34D	—	L16T_A0
AP26	5 (BC)	6	IO	PB25C	PB35A	—	—
AK25	5 (BC)	6	IO	PB26A	PB35C	—	L17T_A0
AN26	5 (BC)	6	IO	PB26B	PB35D	VREF_5_06	L17C_A0
AP27	5 (BC)	6	IO	PB26C	PB36A	—	—
AM25	5 (BC)	6	IO	PB27A	PB36C	—	L18T_D3
AK26	5 (BC)	6	IO	PB27B	PB36D	—	L18C_D3
N32	—	—	VSS	VSS	VSS	—	—
AL24	—	—	O	TXDAA_P_N	TXDAA_P_N	—	L1N_A0
AK24	—	—	O	TXDAA_P_P	TXDAA_P_P	—	L1P_A0
A32	—	—	VDD33	VDD33	VDD33	—	—
AN27	—	—	O	TXDAB_P_N	TXDAB_P_N	—	L2N_D0
AP28	—	—	O	TXDAB_P_P	TXDAB_P_P	—	L2P_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
P13	—	—	VSS	VSS	VSS	—	—
AL25	—	—	O	TXDAC_P_N	TXDAC_P_N	—	L3N_A0
AL26	—	—	O	TXDAC_P_P	TXDAC_P_P	—	L3P_A0
B32	—	—	VDD33	VDD33	VDD33	—	—
AM26	—	—	O	TXDAD_P_N	TXDAD_P_N	—	L4N_A0
AM27	—	—	O	TXDAD_P_P	TXDAD_P_P	—	L4P_A0
P14	—	—	VSS	VSS	VSS	—	—
AN28	—	—	O	Reserved	Reserved	—	L5N_D0
AP29	—	—	O	Reserved	Reserved	—	L5P_D0
C31	—	—	VDD33	VDD33	VDD33	—	—
AL27	—	—	O	TXCLK_P_N	TXCLK_P_N	—	L6N_A0
AK27	—	—	O	TXCLK_P_P	TXCLK_P_P	—	L6P_A0
P15	—	—	VSS	VSS	VSS	—	—
AL28	—	—	O	TXDBA_P_N	TXDBA_P_N	—	L7N_A0
AK28	—	—	O	TXDBA_P_P	TXDBA_P_P	—	L7P_A0
C33	—	—	VDD33	VDD33	VDD33	—	—
AM28	—	—	O	TXDBB_P_N	TXDBB_P_N	—	L8N_D0
AN29	—	—	O	TXDBB_P_P	TXDBB_P_P	—	L8P_D0
P20	—	—	VSS	VSS	VSS	—	—
AL29	—	—	O	TXDBC_P_N	TXDBC_P_N	—	L9N_A0
AK29	—	—	O	TXDBC_P_P	TXDBC_P_P	—	L9P_A0
C34	—	—	VDD33	VDD33	VDD33	—	—
AP30	—	—	O	TXDBD_P_N	TXDBD_P7_N	—	L10N_D0
AN30	—	—	O	TXDBD_P_P	TXDBD_P_P	—	L10P_D0
P21	—	—	VSS	VSS	VSS	—	—
AM29	—	—	I	DAUTREC	DAUTREC	—	—
AP31	—	—	I	TSTCLK	TSTCLK	—	—
D32	—	—	VDD33	VDD33	VDD33	—	—
AM30	—	—	I	TESTRST	TESTRST	—	—
AN31	—	—	I	TSTSHFTLD	TSTSHFTLD	—	—
P22	—	—	VSS	VSS	VSS	—	—
R13	—	—	VSS	VSS	VSS	—	—
R14	—	—	VSS	VSS	VSS	—	—
E30	—	—	VDD33	VDD33	VDD33	—	—
AL30	—	—	I	RESETTX	RESETTX	—	—
E31	—	—	VDD33	VDD33	VDD33	—	—
AH30	—	—	I	ETOGGLE	ETOGGLE	—	—
AJ30	—	—	I	ELSEL	ELSEL	—	—
R15	—	—	VSS	VSS	VSS	—	—
AL33	—	—	I	EXDNUP	EXDNUP	—	—
AH31	—	—	I	MRESET	MRESET	—	—
L34	—	—	VDD33	VDD33	VDD33	—	—

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
AK32	—	—	I	RXDAA_P_N	RXDAA_P_N	—	L11N_D0
AJ31	—	—	I	RXDAA_P_P	RXDAA_P_P	—	L11P_D0
R20	—	—	VSS	VSS	VSS	—	—
AL34	—	—	I	RXDAB_P_N	RXDAB_P_N	—	L12N_D0
AK33	—	—	I	RXDAB_P_P	RXDAB_P_P	—	L12P_D0
AJ32	—	—	I	LVCTAP_P_0	LVCTAP_P_0	—	—
M32	—	—	VDD33	VDD33	VDD33	—	—
AF30	—	—	I	RXDAC_P_N	RXDAC_P_N	—	L13N_A0
AG30	—	—	I	RXDAC_P_P	RXDAC_P_P	—	L13P_A0
R21	—	—	VSS	VSS	VSS	—	—
AG31	—	—	I	RXDAD_P_P	RXDAD_P_N	—	L14N_A0
AF31	—	—	I	RXDAD_P_P	RXDAD_P_P	—	L14P_A0
AK34	—	—	I	LVCTAP_P_1	LVCTAP_P_1	—	—
R32	—	—	VDD33	VDD33	VDD33	—	—
AJ33	—	—	I	Reserved	Reserved	—	L15N_A0
AH32	—	—	I	Reserved	Reserved	—	L15P_A0
R22	—	—	VSS	VSS	VSS	—	—
AJ34	—	—	I	Reserved	Reserved	—	L16N_D0
AH33	—	—	I	Reserved	Reserved	—	L16P_D0
AD30	—	—	I	LVCTAP_P_2	LVCTAP_P_2	—	—
U34	—	—	VDD33	VDD33	VDD33	—	—
AG32	—	—	I	RXDBA_P_N	RXDBA_P_N	—	L17N_A0
AG33	—	—	I	RXDBA_P_P	RXDBA_P_P	—	L17P_A0
T16	—	—	VSS	VSS	VSS	—	—
AH34	—	—	I	LVCTAP_P_3	LVCTAP_P_3	—	—
AE30	—	—	I	RXDBB_P_N	RXDBB_P_N	—	L18N_A0
AE31	—	—	I	RXDBB_P_P	RXDBB_P_P	—	L18P_A0
W34	—	—	VDD33	VDD33	VDD33	—	—
AF32	—	—	I	RXDBC_P_N	RXDBC_P_N	—	L19N_A0
AF33	—	—	I	RXDBC_P_P	RXDBC_P_P	—	L19P_A0
T17	—	—	VSS	VSS	VSS	—	—
AC30	—	—	I	LVCTAP_P_4	LVCTAP_P_4	—	—
AG34	—	—	I	RXDBD_P_N	RXDBD_P_N	—	L20N_A0
AF34	—	—	I	RXDBD_P_P	RXDBD_P_P	—	L20P_A0
Y32	—	—	VDD33	VDD33	VDD33	—	—
AB30	—	—	VDDA_STM	VDDA_STM	VDDA_STM	—	—
AD31	—	—	VSSA_STM	VSSA_STM	VSSA_STM	—	—
T18	—	—	VSS	VSS	VSS	—	—
AE32	—	—	I	SYS_CLK_N	SYS_CLK_N	—	L21N_D0
AE33	—	—	I	SYS_CLK_P	SYS_CLK_P	—	L21P_D0
AC32	—	—	VDD33	VDD33	VDD33	—	—
AE34	—	—	O	LVCTAP_SK	LVCTAP_SK	—	—

Package Outline Drawings

Figure 40. 680-Pin PBGAM Outline Drawings

Dimensions are in millimeters.