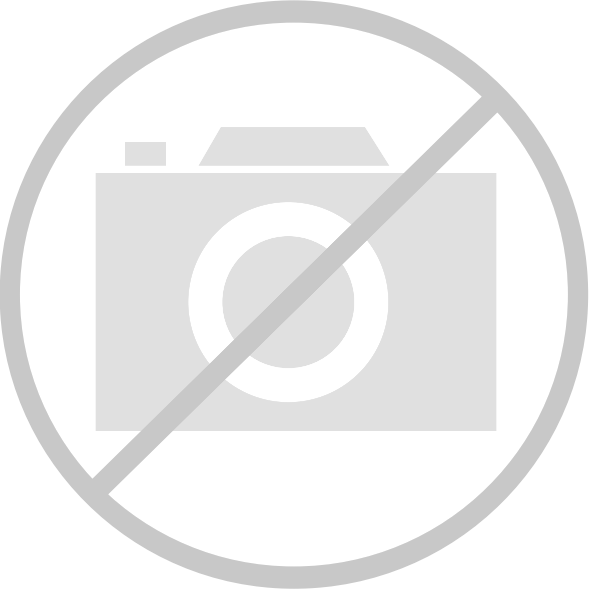
# E·XFL Lattice Semiconductor Corporation - ORT8850H-1BM680I Datasheet



**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 | -40°C ~ 85°C (TA) |
| Package / Case | 680-BBGA |
| Supplier Device Package | 680-FPBGA (35x35) |
| Purchase URL | https://www.e-xfl.com/product-detail/lattice-semiconductor/ort8850h-1bm680i |

# 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.

- Meets Universal Test and Operations PHY Interface for ATM (UTOPIA) Levels 1, 2, and 3. Also meets proposed specifications for UTOPIA Level 4 POS-PHY, Level 3 (2.5 Gbits/s), and POS-PHY 4 (10 Gbits/s) interface standards for Packet-over-SONET as defined by the Saturn Group.
- ispLEVER development system software. Supported by industry-standard CAE tools for design entry, synthesis, simulation, and timing analysis.

# Description

## What is an FPSC?

FPSCs, or Field Programmable System-on-a-Chip devices, are devices that combine field-programmable logic with ASIC or mask-programmed logic on a single device. FPSCs provide the time to market and the flexibility of FPGAs, the design effort savings of using soft Intellectual Property (IP) cores, and the speed, design density, and economy of ASICs.

## FPSC Overview

Lattice's Series 4 FPSCs are created from Series 4 *ORCA* FPGAs. To create a Series 4 FPSC, several columns of Programmable Logic Cells (see FPGA Logic Overview section for FPGA logic details) are added to an embedded logic core. Other than replacing some FPGA gates with ASIC gates, at greater than 10:1 efficiency, none of the FPGA functionality is changed—all of the Series 4 FPGA capability is retained: embedded block RAMs, MPI, PCMs, boundary scan, etc. Columns of programmable logic are replaced on one side of the device, allowing pins from the replaced columns to be used as I/O pins for the embedded core. The remainder of the device pins retain their FPGA functionality.

### FPSC Gate Counting

The total gate count for an FPSC is the sum of its embedded core (standard-cell/ASIC gates) and its FPGA gates. Because FPGA gates are generally expressed as a usable range with a nominal value, the total FPSC gate count is sometimes expressed in the same manner. Standard-cell ASIC gates are, however, 10 to 25 times more silicon-area efficient than FPGA gates. Therefore, an FPSC with an embedded function is gate equivalent to an FPGA with a much larger gate count.

### FPGA/Embedded Core Interface

The interface between the FPGA logic and the embedded core has been enhanced to provide a greater number of interface signals than on previous FPSC architectures. Compared to bringing embedded core signals off-chip, this on-chip interface is much faster and requires less power.

Series 4 based FPSCs expand this interface by providing a link between the embedded block and the multi-master 32-bit system bus in the FPGA logic. This system bus allows the core easy access to many of the FPGA logic functions including the embedded block RAMs and the MicroProcessor Interface.

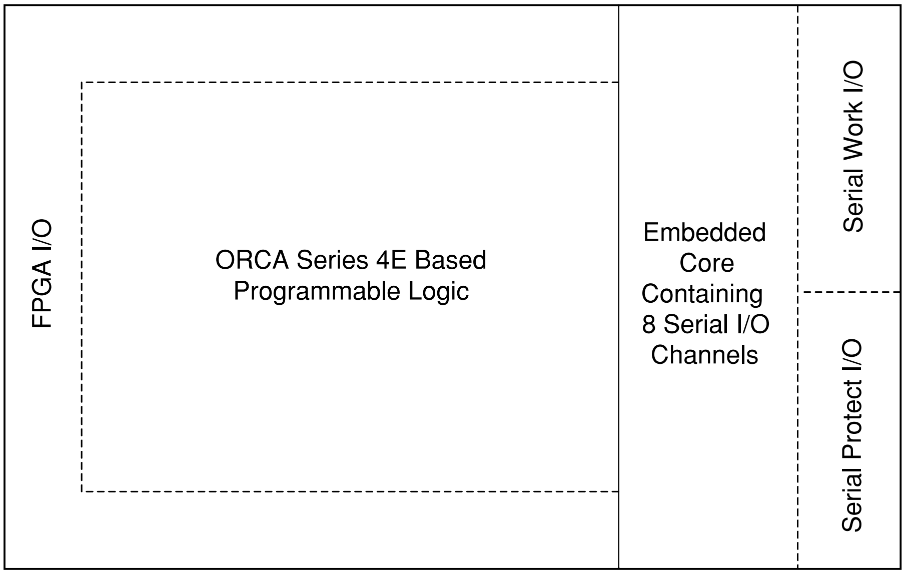
Clock spines also can pass across the FPGA/embedded core boundary. This allows fast, low-skew clocking between the FPGA and the embedded core. Many of the special signals from the FPGA, such as DONE and global set/reset, are also available to the embedded core, making it possible to fully integrate the embedded core with the FPGA as a system.

For even greater system flexibility, FPGA configuration RAMs are available for use by the embedded core. This supports user-programmable options in the embedded core, in turn allowing greater flexibility. Multiple embedded core configurations may be designed into a single device with user-programmable control over which configurations are implemented, as well as the capability to change core functionality simply by reconfiguring the device.

## ORT8850 Overview

The ORT8850 FPSCs provide high-speed backplane transceivers combined with FPGA logic. There are two devices in the ORT8850 family. The ORT8850L device is based on 1.5 V OR4E02 ORCA FPGA and has a 26 x 24 array of Programmable Logic Cells (PLCs). The ORT8850H device is based on 1.5V OR4E06 ORCA FPGA and has a 46 x 44 array. The embedded core which contains the backplane transceivers is attached to the right side of the device and is integrated directly into the FPGA array. A top level diagram of the basic chip configuration is shown in Figure 1.

*Figure 1. ORT8850 Top Level Diagram*



### Embedded Core Overview

The ORT8850 embedded core contains a pseudo-SONET block for backplane or intra-board, chip-to-chip communication. The SONET block includes a High-Speed Interface (HSI) macrocell and a Synchronous Transport Module (STM) macrocell. It supports eight full-duplex channels and performs data transfer, scrambling/descrambling and SONET framing at the maximum rate of 850 Mbits/s. Figure 2 shows a top level diagram of the ORT8850 and the basic data flows through the device.

*Table 5. Channel Alignment, Transparent TOH (Continued)*

| Register Address | Value | Description |
|---|---|---|
| **Initial Register Settings** | | |
| 0x30038 | 0x07 | Channel AB in functional mode without AIS-L |
| 0x30050 | 0x07 | Channel AC in functional mode without AIS-L |
| 0x30068 | 0x07 | Channel AD in functional mode without AIS-L |
| 0x30080 | 0x07 | Channel BA in functional mode without AIS-L |
| 0x30098 | 0x07 | Channel BB in functional mode without AIS-L |
| 0x300B0 | 0x07 | Channel BC in functional mode without AIS-L |
| 0x300C8 | 0x07 | Channel BD in functional mode without AIS-L |

# Backplane Transceiver Core Detailed Description

## SONET Logic Blocks, Detailed Description

The following sections describe the data processing performed in the SONET logic blocks. A 622 Mbits/s is assumed in the descriptions however, as noted in the Overview sections, the ORT8850 can operate at variable rates up to 850 Mbits/s. At a top level, the descriptions are separated into processing in the transmit path (FPGA to serial link) and processing in the receive path (serial link to FPGA). A top level drawing of the two data paths and associated clocks is shown in Figure 11. The various processing options are selected by setting bits in control registers and status information is written to status registers. Both types of registers can be written and/or read from the System Bus or the MicroProcessor Interface (MPI). Memory maps and descriptions for the registers are given in Table 19.

*Figure 11. ORT8850 Top Level Data Flow*

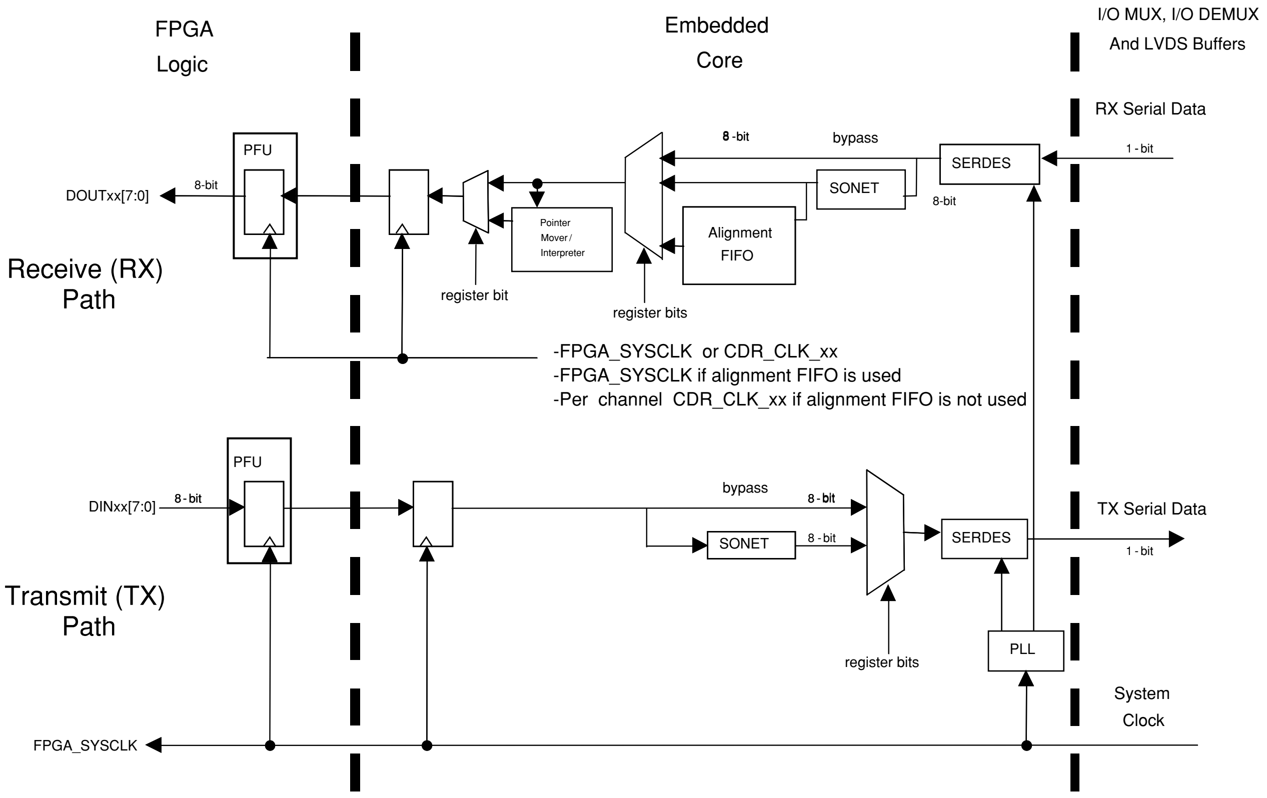


*Table 8. Byte Ordering, Quad STS-12 (OC-48) Format*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STS-12 A --> | 12 | 9 | 6 | 3 | 11 | 8 | 5 | 2 | 10 | 7 | 4 | 1 |
| STS-12 B --> | 24 | 21 | 18 | 15 | 23 | 20 | 17 | 14 | 22 | 19 | 16 | 13 |
| STS-12 C --> | 36 | 33 | 30 | 27 | 35 | 32 | 29 | 26 | 34 | 31 | 28 | 25 |
| STS-12 D --> | 48 | 45 | 42 | 39 | 47 | 44 | 41 | 38 | 46 | 43 | 40 | 37 |

All internal framing is based on the system frame pulse (SYS_FP) which is a one-cycle pulse at an 8kHz rate. There is one system frame pulse for all 8 channels or both quads. When the framer receives the system frame pulse the individual overhead bytes are identified.

### HSI Macrocell

The ORT8850 High-Speed Interface (HSI) provides a physical medium for high-speed asynchronous serial data transfer between ASIC devices. The devices can be mounted on the same PC board or mounted on different boards and connected through the shelf back-plane. The ORT8850 CDR macro is an eight-channel Clock-Phase Select (CPS) and data retime function with serial-to-parallel demultiplexing for the incoming data stream and parallel-to-serial multiplexing for outgoing data. The ORT8850 uses an eight-channel HSI macro cell. The HSI macro consists of three functionally independent blocks: receiver, transmitter, and PLL synthesizer.

The PLL synthesizer block generates the necessary 850 MHz clock for operation from a 106.25 MHz, reference. The PLL synthesizer block is a common asset shared by all eight receive and transmit channels. The PLL reference clock must match the interface frequency.

The HSI_RX block receives differential 850 Mbits/s serial data without clock at its LVDS receiver input. Based on data transitions, the receiver selects an appropriate 850 MHz clock phase for each channel to retime the data. The retimed data and clock are then passed to the deMUX (deserializer) module. DeMUX module performs serial-to-parallel conversion and provides the 106 Mbits/s data and clock.

The HSI_TX block receives 106 Mbits/s parallel data at its input. MUX (serializer) module performs a parallel-to-serial conversion using an 850 MHz clock provided by the PLL/synthesizer block. The resulting 850 Mbits/s serial data stream is then transmitted through the LVDS driver.

The loopback feature built into the HSI macro provides looping of the transmitter data output into the receiver input when desired.

All rate examples described here are the maximum rates possible. The actual HSI internal clock rate is determined by the provided reference clock rate. For example, if a 77.76 MHz reference clock is provided, the HSI macro will operate at 622 Mbits/s.

### Transmit Path Logic

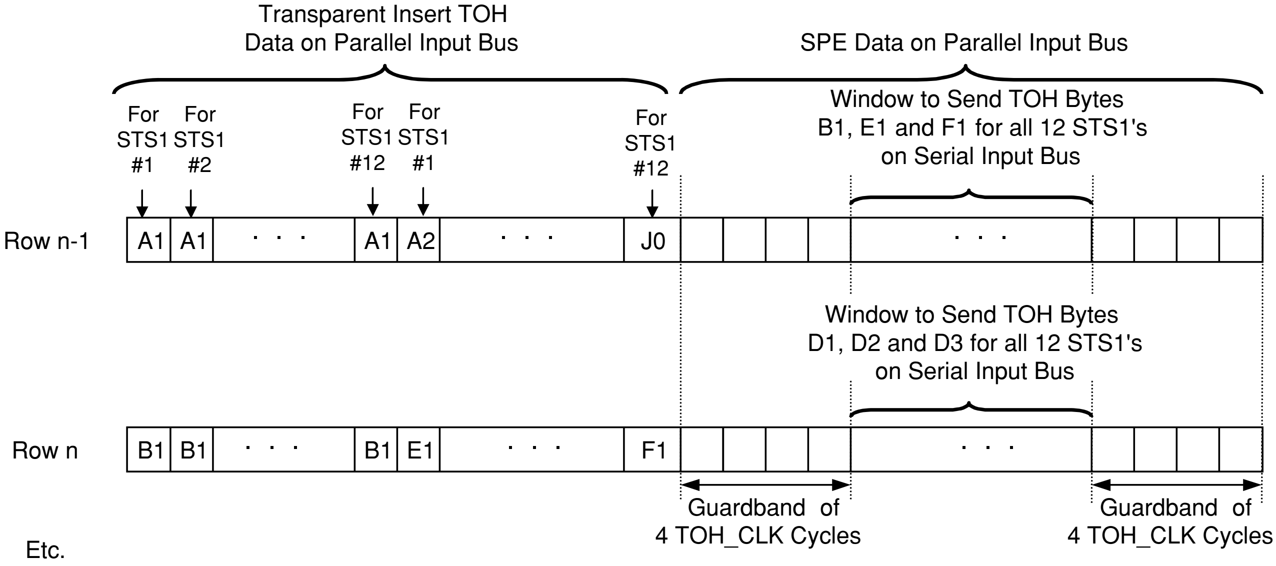
In the transmit direction each STM quad will receive frame aligned streams of STS-12 data (maximum of four streams per quad) from the FPGA logic. The transmitter receives data interface in a parallel 8-bit format. A common frame pulse for all 8 channels is provided as an input from the FPGA logic to the transmit SONET block.

The system frame pulse is a single pulse at the reference clock rate every 9720 clock cycles. For a 77.76 MHz reference clock this creates an 8KHz pulse rate. The system frame pulse (SYS_FP) is used to generate the A1/A2 in the transmit direction. It is also used by the Pointer Mover Block to perform the line side loopback, which otherwise uses the LINE_FP frame pulse also provided by the user from the FPGA to the Embeddded ASIC Block. The Function of the LINE_FP is mentioned in the Pointer Mover bypass description.

The system frame pulse is common to all channels in the transmit direction. Once it is received from the FPGA logic, the data to be transmitted goes through the following processing steps:

- A parity check is performed on the data
- The Transport Overhead (TOH) data is modified (optional)

*Figure 15. TOH Serial Port Input Framing Signals (FPGA to Core)*



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.

Except for the above hardcode exceptions, the source of some TOH bytes can be controlled by bits in the control registers. The 12 STS-1 bytes forming a single STS-12 TOH header block are controlled as a whole. When configured to be in the transparent mode, the specific bytes must flow transparently from the parallel input. The 15 overhead bytes that can be controlled on a per STS-1 basis are the following:

- K1 and K2 bytes of the 12 STS-1s (24 bytes)
- S1 and M0 bytes of the 12 STS-1s (24 bytes)
- E1, F1, E2 bytes of the STS-1s (36 bytes)
- D1 through D12 bytes of the STS-1s (144 bytes)

The C1(J0) and B2 bytes (unshaded in the following table) are also passed through transparently from the parallel bus to the serial link.

Table 10 shows the order in which data is transferred to the serial LVDS output, starting with the most significant bit of the first A1 byte. The first bit of the first byte is replaced by an even parity check bit over all TOH bytes from the previous TOH frame. The source for the TOH bytes in the Serial TOH insert mode is summarized in the table.

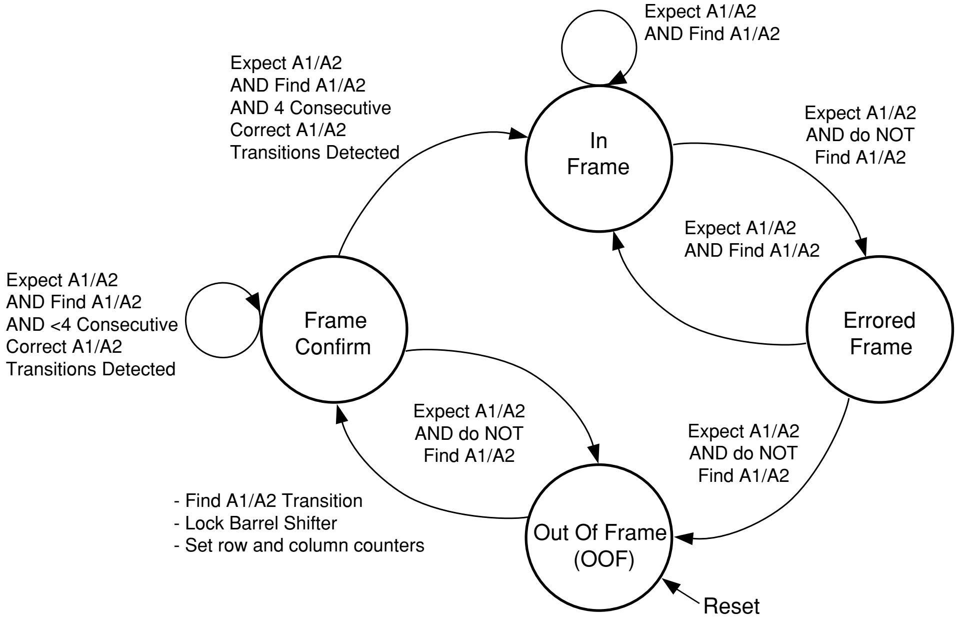
mover is active, there is no fixed timing relationship between the data sent to the FPSC and DOUTxx_FP and the DOUTxx_SPE and DOUTxx_C1. J1 signals should be used instead to determine data alignment within the frame.) The framer algorithm determines the out-of-frame/in-frame status of the incoming data and will cause alarms on both an errored frame and an OOF (out-of-frame) state. Functions performed by this block include:

- A1-A2 framing pattern detection. (Framing similar to SONET specification)
- Generation of timing and an 8kHz frame pulse.
- Detections of Out Of Frame (OOF) (generates an alarm).
- Errored frame detection (increments error counter).

### Framer State Machine

Figure 17 shows the state machine for the framer. Because the ORT8850 is primarily intended for use between itself and another ORT8850 or other devices via a backplane, there is only one errored frame state. Thus there is no Severely Errored Frame (SEF) or Loss-Of-Frame (LOF) indication.

*Figure 17. Framer State Machine*



Notes:
1) Row and column counters are only set/reset by a state transition from the OOF state to to the Frame Confirm state.
2) "Expect A1/A2" means that the row and column counters have counted to the place for the last (12 th.) A1 byte and that the next byte should be an A2 byte.

### OOF State

This is the initial state for the state machine after a reset. In this state the A1 pattern is searched for on every clock cycle. A second stage of comparison is implemented to locate the A1/A2 transition. When the A1/A2 transition is found, the following occurs:

- The state machine moves from the OOF state to the Frame Confirm State.
- The A1offset for the byte start location is locked.
- Row and column counters are set

### Frame Confirm State

In this state the A1/A2 transition is only compared for at the appropriate location, i.e. beginning at the 12th A1 location. This location is determined from the row and column counters which were set at the transition from OOF to Frame Confirm. If at this time the comparison fails, the state machine reverts to the OOF state. If the comparison

### 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 |

### SPE and C1J1 Identification

In the ORT8850 each frame can be considered as 12 STS-1s. In the SPE region, there are 12 J1 pulses for each STS-1s. There is one C1(J0, new 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, for non concatenated data 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.

The pointer interpreter identifies the payload area of each frame. The SPE flag is active when the data stream is in SPE area. SPE behavior is dependent on pointer movement and concatenation. Note that in the TOH area, H3 can also carry valid data. When valid SPE data is carried in this H3 slot, SPE is high in this particular TOH time slot. In the SPE region, if there is no valid data during any SPE column, the SPE signal will be set to low. SPE allows a pointer processor to extract payload without interpreting the pointers.

*Figure 23. SPE and C1J1 Functionality for STS -12*

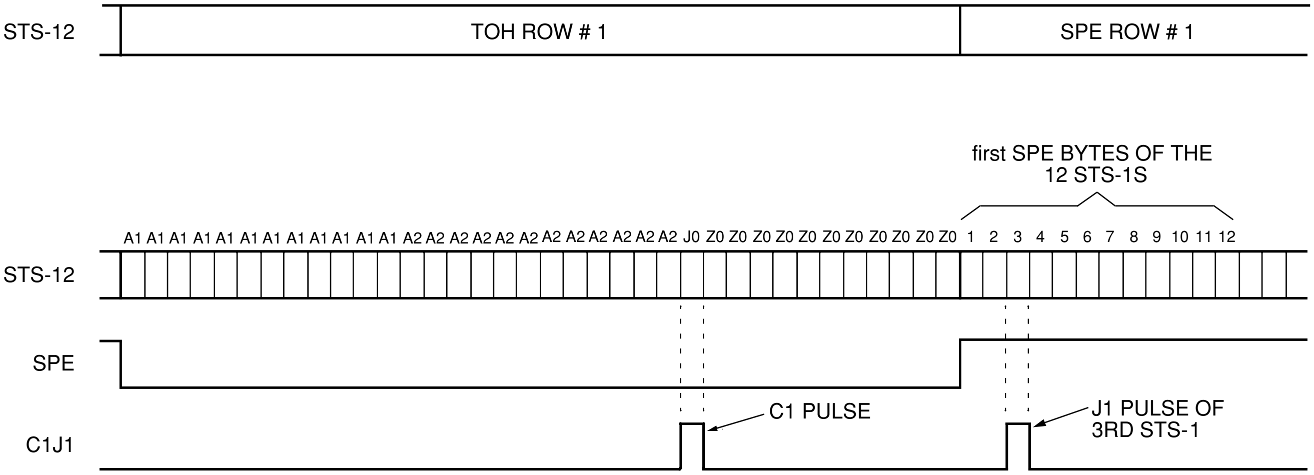
| SPE | C1J1 | Description |
|---|---|---|
| 0 | 0 | TOH information excluding C1(J0) of STS-1 #1. |
| 0 | 1 | Position of C1(J0) of STS-1 #1 (one per frame). Typically used to provide a unique link identification (256 possible unique links) to help ensure cards are connected into the backplane correctly or cables are connected correctly. |
| 1 | 0 | SPE information excluding the 12 J1 bytes. |
| 1 | 1 | Position of the 12 J1 bytes. |

The following rules are observed for generating SPE and C1J1 signals:

- On occurrence of AIS-P on any of the STS-1, there is no corresponding J1 pulse.
- In case of concatenated payloads (up to STS48c), only the head STS-1 of the group has an associated J1 pulse.
- The C1J1 signal tracks any pointer movements.

This behavior is illustrated in the following figure. Note that the actual bit positions are dependent on the actual payload configuration and offset.

*Figure 24. SPE and C1J1 Signals*


STS-12
TOH ROW # 1
SPE ROW # 1
first SPE BYTES OF THE 12 STS-1S
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 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 1 2 3 4 5 6 7 8 9 10 11 12
STS-12
SPE
C1 PULSE
J1 PULSE OF 3RD STS-1
C1J1

### Pointer Mover

After the pointer interpreter comes the pointer mover block. There is a separate pointer mover for the two SONET framer quads, A and B, each of which handles up to one STS-48 (four channels) The K1/K2 bytes and H1-SS bits are also passed through to the pointer generator so that the FPGA can receive them. The pointer mover handles both concatenations inside the STS-12, and to other STS-12s inside the core. Use of this block is optional, as discussed in a later section.

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*

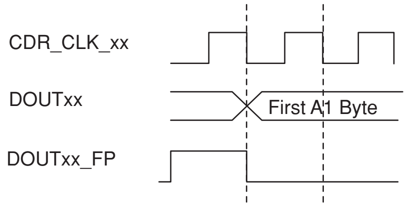


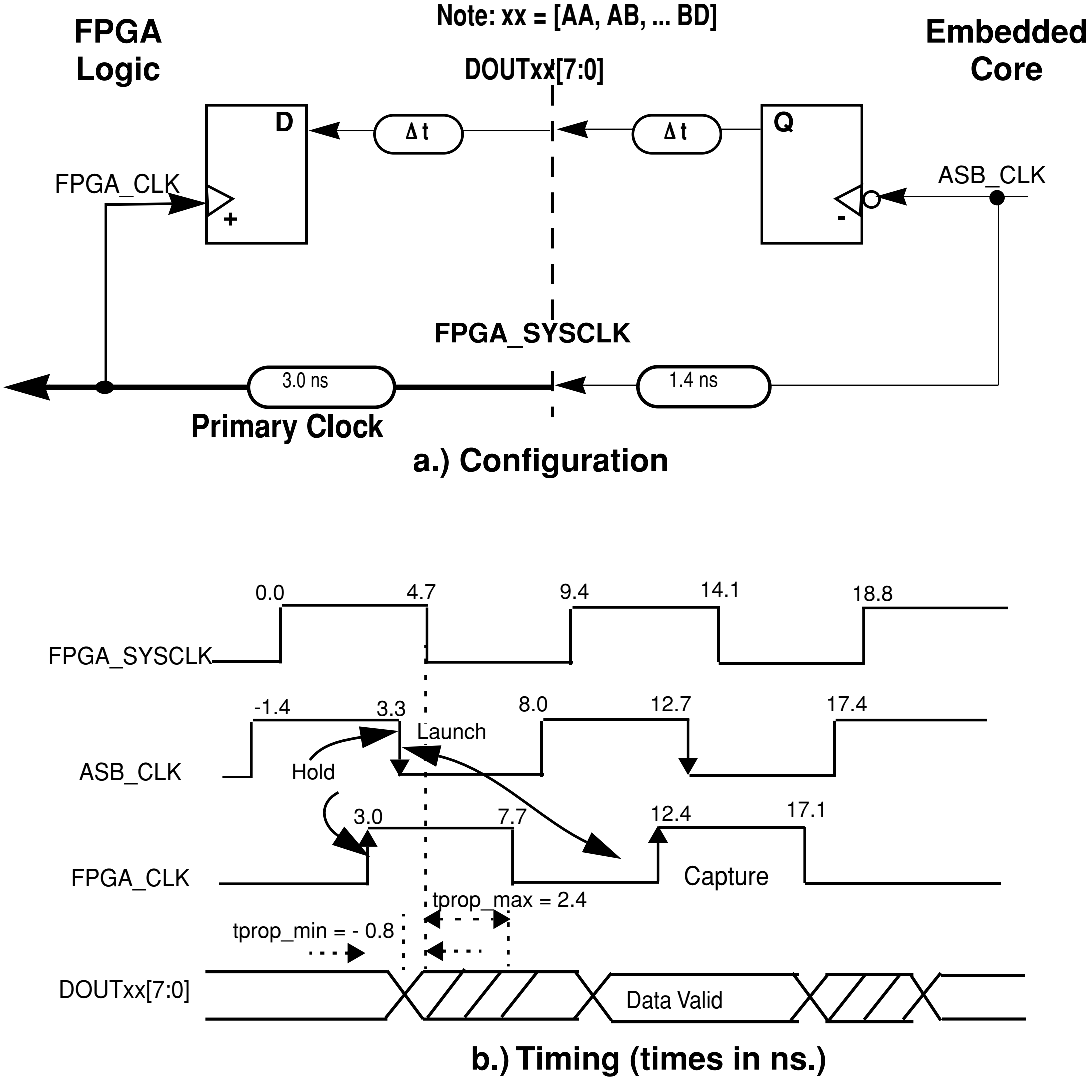
Table 14 shows the register settings to enable the bypass modes.

*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 |

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)***

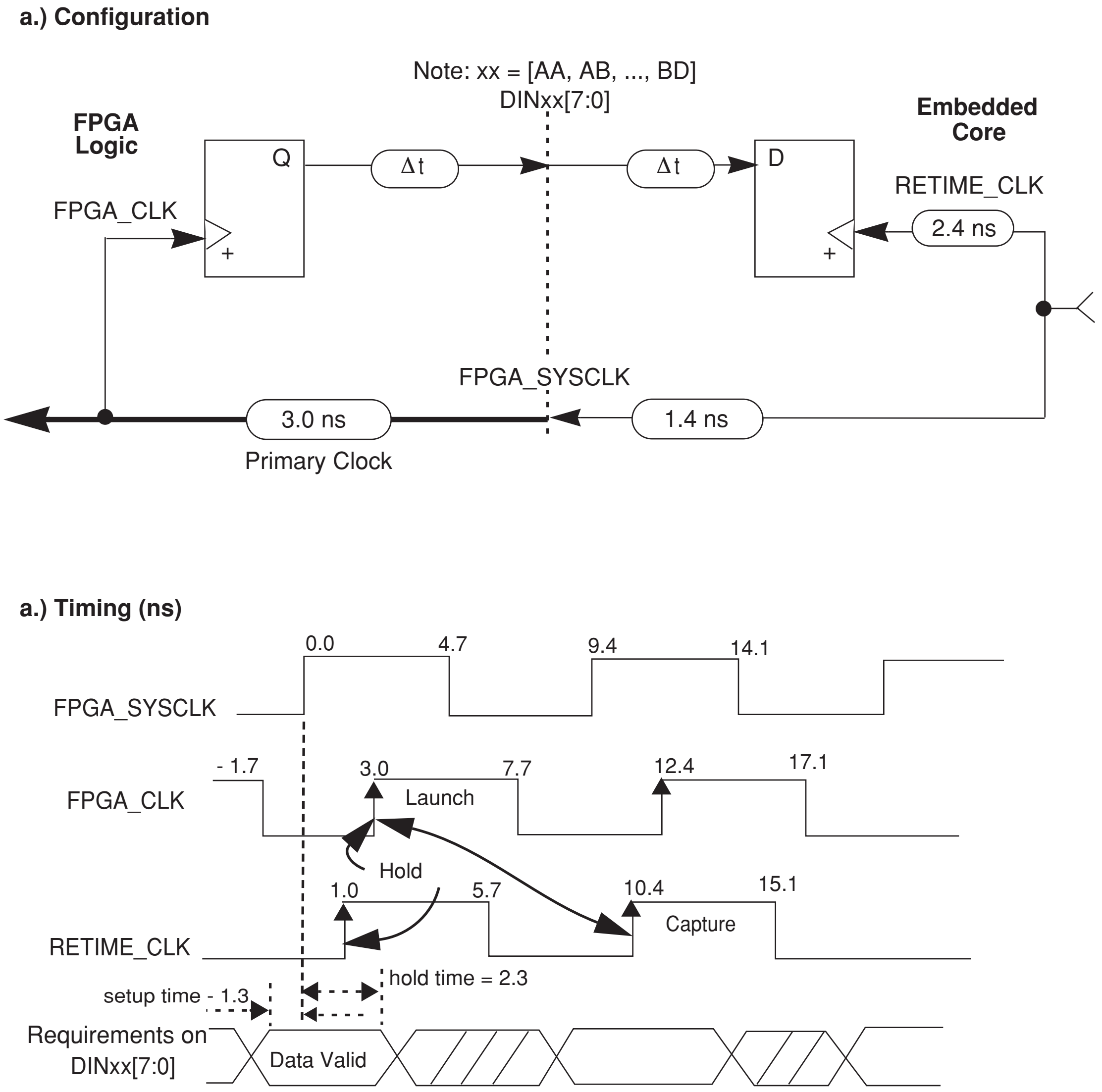


a.) Configuration

b.) Timing (times in ns.)

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 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.<br>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.<br>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.<br>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.<br>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<br>Write “0” for normal operation. |
| | [6] | R/W | All 8 resync | 0 | All 8 channel alignment FIFO resync.<br>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 |
|---|---|---|---|---|---|
| 3002B*<br>30043<br>3005B<br>30073<br>3008B<br>300A3<br>300BB<br>300D3 | [0:7] | R | AIS alarm flags 1, 4, 7, 10, 2, 5, 8, 11 | 00 | These are the AIS-P alarm flags. 1 if the LVDS input STS # contains AIS. |
| 3002C*<br>30044<br>3005C<br>30074<br>3008C<br>300A4<br>300BC<br>300D4 | [0:3] | R/W | enable AIS alarm 3, 6, 9, 12 | 0 | Enable bits for AIS alarms. Set to 1 to enable and propagate the alarm to register 0x30026. |
| | [4-7] | - | Not Used | 0 | |
| 3002D*<br>30045<br>3005D<br>30075<br>3008D<br>300A5<br>300BD<br>300D5 | [0:7] | R/W | AIS alarm enable 1, 4, 7, 10, 2, 5, 8, 11 | 00 | Enable bits for AIS alarms. Set to 1 to enable and propagate the alarm to register 0x30026. |
| 3002E*<br>30046<br>3005E<br>30076<br>3008E<br>300A6<br>300BE<br>300D6 | [0:3] | R | Pointer mover elastic store overflow flags 12, 9, 6, 3 | 0 | Per STS-1 pointer mover elastic store overflow alarm flags. This alarm will propagate to 0x30026 bit 2 when enabled |
| | [4-7] | - | Not Used | 0 | |
| 3002F*<br>30047<br>3005F<br>30077<br>3008F<br>300A7<br>300BF<br>300D7 | [0:7] | R | Pointer mover elastic store overflow flags 4, 7, 10, 2, 5, 8, 11 | 00 | Per STS-1 pointer mover elastic store overflow alarm flags. This alarm will propagate to 0x30026 bit 2 when enabled |
| 30030*<br>30048<br>30060<br>30078<br>30090<br>300A8<br>300C0<br>300D8 | [0:3] | R/W | enable elastic store overflow flag 12, 9, 6, 3 | 0 | Enable Bit for elastic store alarms. Set 1 to enable alarm and propagate alarm to register 0x30026 |
| | [4-7] | - | Not Used | 0 | |

*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 |
|---|---|---|---|---|---|---|---|
| AB1 | 7 (CL) | 6 | IO | PL17A | PL29C | A7/PPC_A21 | L20T_D3 |
| AA5 | 7 (CL) | 6 | IO | PL18D | PL30D | A6/PPC_A20 | L21C_A1 |
| AA3 | 7 (CL) | 6 | IO | PL18C | PL30C | A5/PPC_A19 | L21T_A1 |
| U1 | 7 (CL) | — | VDDIO7 | VDDIO7 | VDDIO7 | — | — |
| AB2 | 7 (CL) | 7 | IO | PL18B | PL31D | — | — |
| AA4 | 7 (CL) | 7 | IO | PL19D | PL32D | WR_N/MPI_RW | L22C_D2 |
| AC1 | 7 (CL) | 7 | IO | PL19C | PL32C | VREF_7_07 | L22T_D2 |
| AB5 | 7 (CL) | 7 | IO | PL19B | PL33D | — | L23C_D2 |
| AC2 | 7 (CL) | 7 | IO | PL19A | PL33C | — | L23T_D2 |
| AB4 | 7 (CL) | 8 | IO | PL20D | PL34D | A4/PPC_A18 | L23C_D0 |
| AC5 | 7 (CL) | 8 | IO | PL20C | PL34C | VREF_7_08 | L23T_D0 |
| W1 | 7 (CL) | — | VDDIO7 | VDDIO7 | VDDIO7 | — | — |
| AD2 | 7 (CL) | 8 | IO | PL20B | PL35D | A3/PPC_A17 | L23C_D0 |
| AE1 | 7 (CL) | 8 | IO | PL20A | PL35C | A2/PPC_A16 | L23T_D0 |
| AD3 | 7 (CL) | 8 | IO | PL21D | PL36D | A1/PPC_A15 | L24C_D0 |
| AE2 | 7 (CL) | 8 | IO | PL21C | PL36C | A0/PPC_A14 | L24T_D0 |
| AF1 | 7 (CL) | 8 | IO | PL21B | PL37D | DP0 | L25C_D2 |
| AD4 | 7 (CL) | 8 | IO | PL21A | PL37C | DP1 | L25T_D2 |
| AE3 | 6 (BL) | 1 | IO | PL22D | PL38D | D8 | L1C_D0 |
| AF2 | 6 (BL) | 1 | IO | PL22C | PL38C | VREF_6_01 | L1T_D0 |
| AB13 | — | — | VSS | VSS | VSS | — | — |
| AE4 | 6 (BL) | 1 | IO | PL22B | PL39D | D9 | L2C_D0 |
| AF3 | 6 (BL) | 1 | IO | PL22A | PL39C | D10 | L2T_D0 |
| AE5 | 6 (BL) | 2 | IO | PL23D | PL40D | — | L3C_D1 |
| AG2 | 6 (BL) | 2 | IO | PL23C | PL40C | VREF_6_02 | L3T_D1 |
| AK5 | 6 (BL) | — | VDDIO6 | VDDIO6 | VDDIO6 | — | — |
| AH1 | 6 (BL) | 2 | IO | PL23B | PL41D | — | L4C_D3 |
| AF5 | 6 (BL) | 2 | IO | PL23A | PL41C | — | L4T_D3 |
| AF4 | 6 (BL) | 3 | IO | PL24D | PL42D | D11 | L5C_D0 |
| AG3 | 6 (BL) | 3 | IO | PL24C | PL42C | D12 | L5T_D0 |
| AB14 | — | — | VSS | VSS | VSS | — | — |
| AH2 | 6 (BL) | 3 | IO | PL24B | PL43D | — | L6C_D0 |
| AJ1 | 6 (BL) | 3 | IO | PL24A | PL43C | — | L6T_D0 |
| AG4 | 6 (BL) | 3 | IO | PL25D | PL44D | VREF_6_03 | L7C_A0 |
| AG5 | 6 (BL) | 3 | IO | PL25C | PL44C | D13 | L7T_A0 |
| AL3 | 6 (BL) | — | VDDIO6 | VDDIO6 | VDDIO6 | — | — |
| AH3 | 6 (BL) | 4 | IO | PL25B | PL44B | — | — |
| AK1 | 6 (BL) | 4 | IO | PL25A | PL45A | — | — |
| AJ2 | 6 (BL) | 4 | IO | PL26D | PL45D | — | L8C_D2 |
| AH5 | 6 (BL) | 4 | IO | PL26C | PL45C | VREF_6_04 | L8T_D2 |
| AB15 | — | — | VSS | VSS | VSS | — | — |
| AH4 | 6 (BL) | 4 | IO | PL26B | PL46D | — | — |

*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 |
|---|---|---|---|---|---|---|---|
| E7 | — | — | IO | PCCLK | PCCLK | CCLK | — |
| D5 | — | — | IO | PDONE | PDONE | DONE | — |
| E6 | — | — | VDD33 | VDD33 | VDD33 | — | — |
| B34 | — | — | VSS | VSS | VSS | — | — |
| A24 | 1 (TC) | — | VDDIO1 | VDDIO1 | VDDIO1 | — | — |
| AM23 | 5 (BC) | — | VDDIO5 | VDDIO5 | VDDIO5 | — | — |
| AP1 | — | — | VSS | VSS | VSS | — | — |
| K4 | 0 (TL) | 10 | IO | UNUSED | PL11A | — | — |
| M5 | 0 (TL) | 10 | IO | UNUSED | PL13A | — | — |
| R5 | 7 (CL) | 3 | IO | UNUSED | PL20A | — | — |
| T5 | 7 (CL) | 3 | IO | UNUSED | PL21A | — | — |
| W4 | 7 (CL) | 5 | IO | UNUSED | PL27A | — | — |
| AA2 | 7 (CL) | 6 | IO | UNUSED | PL28A | — | — |
| Y4 | 7 (CL) | 6 | IO | UNUSED | PL29A | — | — |
| AC4 | 7 (CL) | 8 | IO | UNUSED | PL35A | — | — |
| AD5 | 7 (CL) | 8 | IO | UNUSED | PL37A | — | — |
| AG1 | 6 (BL) | 1 | IO | UNUSED | PL38A | — | — |
| AK10 | 6 (BL) | 7 | IO | UNUSED | PB9A | — | — |
| AK11 | 6 (BL) | 7 | IO | UNUSED | PB10A | — | — |
| AM9 | 6 (BL) | 8 | IO | UNUSED | PB11A | — | — |
| AN9 | 6 (BL) | 8 | IO | UNUSED | PB12A | — | — |
| AM14 | 6 (BL) | 11 | IO | UNUSED | PB19A | — | — |
| AN14 | 6 (BL) | 11 | IO | UNUSED | PB20A | — | — |
| D11 | 0 (TL) | 3 | IO | UNUSED | PT12A | — | — |
| E13 | 0 (TL) | 3 | IO | UNUSED | PT11A | — | — |
| AP4 | 6 (BL) | 5 | IO | UNUSED | PB3A | — | — |
| Y3 | 7 (CL) | — | VDDIO7 | VDDIO7 | VDDIO7 | — | — |
| AC3 | 7 (CL) | — | VDDIO7 | VDDIO7 | VDDIO7 | — | — |
| AD1 | 7 (CL) | — | VDDIO7 | VDDIO7 | VDDIO7 | — | — |
| AP11 | 5 (BC) | — | VDDIO5 | VDDIO5 | VDDIO5 | — | — |
| AP17 | 5 (BC) | — | VDDIO5 | VDDIO5 | VDDIO5 | — | — |
| AP19 | 5 (BC) | — | VDDIO5 | VDDIO5 | VDDIO5 | — | — |
| AP24 | 5 (BC) | — | VDDIO5 | VDDIO5 | VDDIO5 | — | — |
| C12 | 1 (TC) | — | VDDIO1 | VDDIO1 | VDDIO1 | — | — |
| C15 | 1 (TC) | — | VDDIO1 | VDDIO1 | VDDIO1 | — | — |
| C20 | 1 (TC) | — | VDDIO1 | VDDIO1 | VDDIO1 | — | — |
| C23 | 1 (TC) | — | VDDIO1 | VDDIO1 | VDDIO1 | — | — |
| W22 | — | — | VDD15 | VDD15 | VDD15 | — | — |
| Y16 | — | — | VDD15 | VDD15 | VDD15 | — | — |
| V22 | — | — | VDD15 | VDD15 | VDD15 | — | — |
| U22 | — | — | VDD15 | VDD15 | VDD15 | — | — |
| T22 | — | — | VDD15 | VDD15 | VDD15 | — | — |

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

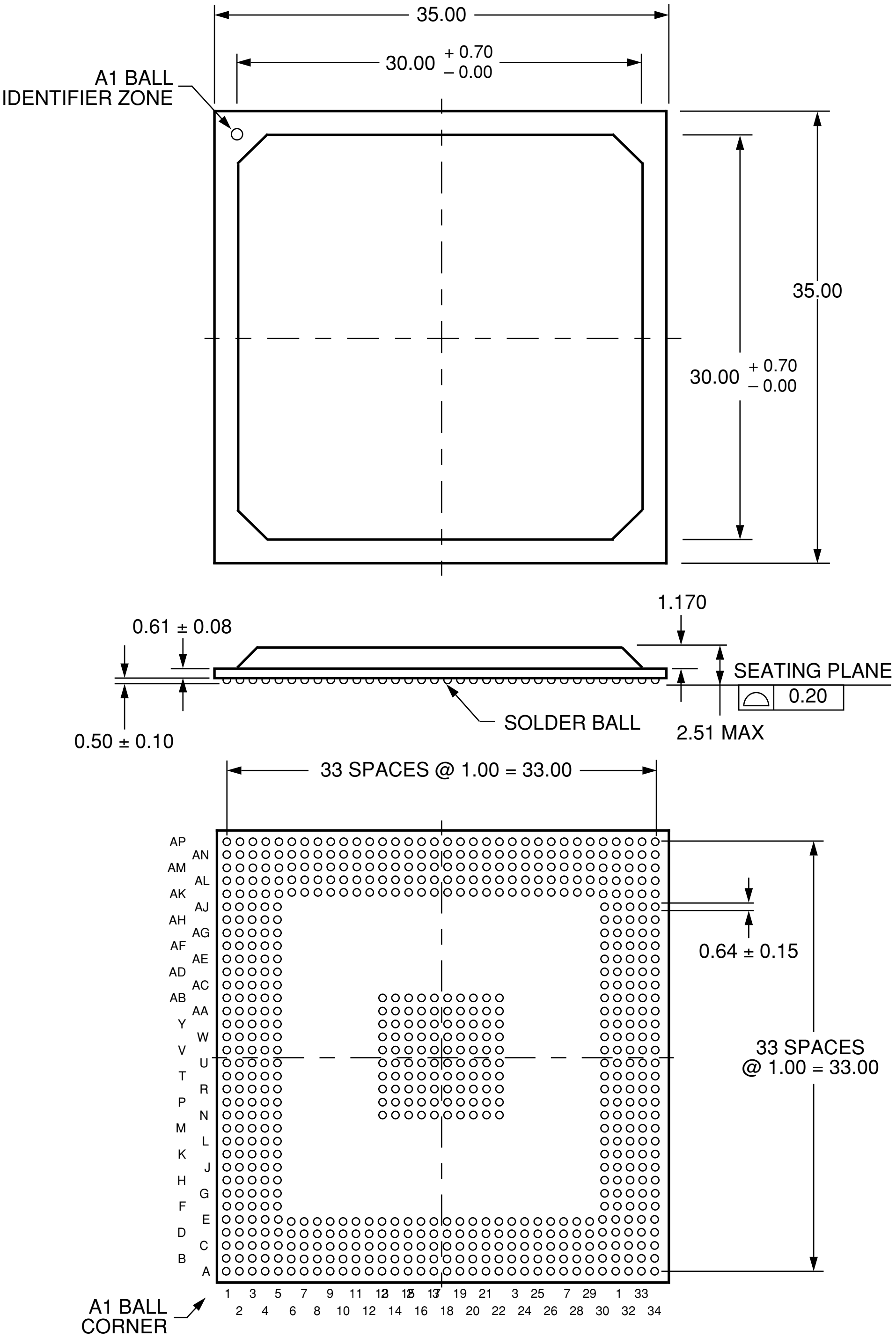
| BM680 | VDDIO Bank | VREF Group | I/O | ORT8850L | ORT8850H | Additional Function | Pair |
|---|---|---|---|---|---|---|---|
| C3 | — | — | VSS | VSS | VSS | — | — |
| C13 | — | — | VSS | VSS | VSS | — | — |
| AP2 | — | — | VSS | VSS | VSS | — | — |
| AP18 | — | — | VSS | VSS | VSS | — | — |
| AP33 | — | — | VSS | VSS | VSS | — | — |
| AP34 | — | — | VSS | VSS | VSS | — | — |
| AA20 | — | — | VSS | VSS | VSS | — | — |
| AA21 | — | — | VSS | VSS | VSS | — | — |
| AA22 | — | — | VSS | VSS | VSS | — | — |
| N21 | — | — | VSS | VSS | VSS | — | — |
| N22 | — | — | VSS | VSS | VSS | — | — |
| AB3 | — | — | VSS | VSS | VSS | — | — |
| AB19 | — | — | VDD15 | VDD15 | VDD15 | — | — |
| N20 | — | — | VSS | VSS | VSS | — | — |

Note: Pins labeled “reserved” should be left unconnected.

## Package Outline Drawings

***Figure 40. 680-Pin PBGAM Outline Drawings***

Dimensions are in millimeters.



## Ordering Information

*Figure 41. Part Number Description*

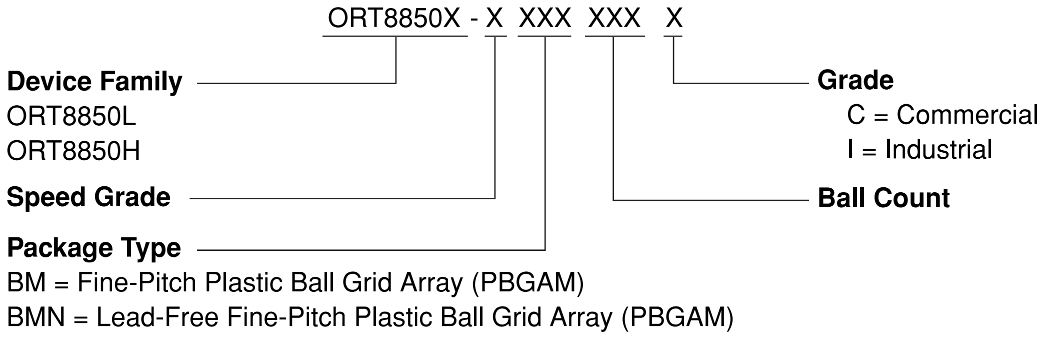


*Table 40. Device Type Options*

| Device | Voltage |
|---|---|
| ORT8850L | 1.5 V internal<br>3.3 V/2.5 V/1.8 V/1.5 V I/O |
| ORT8850H | 1.5 V internal<br>3.3 V/2.5 V/1.8 V/1.5 V I/O |

*Table 41. Temperature Range*

| Symbol | Description | Ambient Temperature | Junction Temperature |
|---|---|---|---|
| C | Commercial | 0 ˚C to +70 ˚C | 0 ˚C to +85 ˚C |
| I | Industrial | –40 ˚C to +85 ˚C | –40 ˚C to +100 ˚C |

*Table 42. Conventional Packaging – Commercial Ordering Information*[1]

| Device Family | Part Number | Speed Grade | Package Type | Ball Count | Grade |
|---|---|---|---|---|---|
| ORT8850L | ORT8850L-3BM680C | 3 | PBGAM (fpBGA) | 680 | C |
| | ORT8850L-2BM680C | 2 | PBGAM (fpBGA) | 680 | C |
| | ORT8850L-1BM680C | 1 | PBGAM (fpBGA) | 680 | C |
| ORT8850H | ORT8850H-2BM680C | 2 | PBGAM (fpBGA) | 680 | C |
| | ORT8850H-1BM680C | 1 | PBGAM (fpBGA) | 680 | C |

*Table 43. Conventional Packaging – Industrial Ordering Information*[1]

| Device Family | Part Number | Speed Grade | Package Type | Ball Count | Grade |
|---|---|---|---|---|---|
| ORT8850L | ORT8850L-2BM680I | 2 | PBGAM (fpBGA) | 680 | I |
| | ORT8850L-1BM680I | 1 | PBGAM (fpBGA) | 680 | I |
| ORT8850H | ORT8850H-1BM680I | 1 | PBGAM (fpBGA) | 680 | I |

[1] For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade is marked as commercial grade only.