

**FEATURES**

- **Functionality**
  - Low speed to high speed SPI exchange device
  - Logical port (LP) mapping (SPI-3 <-> SPI-4) tables per direction
  - Per LP configurable memory allocation
  - Maskable interrupts for fatal errors
  - Fragment and burst length configurable per interface: min 16 bytes, max 256 bytes
- **Standard Interfaces**
  - Four OIF SPI-3: 8 or 32 bit, 19.44-133 MHz, 256 address range, 64 concurrently active LPs per interface
  - One OIF SPI-4 phase 2: 80 - 400 MHz, 256 address range, 256 concurrently active LPs
  - SPI-4 FIFO status channel options:
    - LVDS full-rate
    - LVTTL eighth-rate
  - Compatible with Network Processor Streaming Interface (NPSI) NPE-Framer mode of operation
  - SPI-4 ingress LVDS automatic bit alignment and lane de-skew over the entire frequency range
  - SPI-4 egress LVDS programmable lane pre-skew 0.1 to 0.3 cycle
  - IEEE 1149.1 JTAG
  - Serial or parallel microprocessor interface for control and monitoring
- **Full Suite of Performance Monitoring Counters**
  - Number of packets

- Number of fragments
- Number of errors
- Number of bytes

- Green parts available, see ordering information

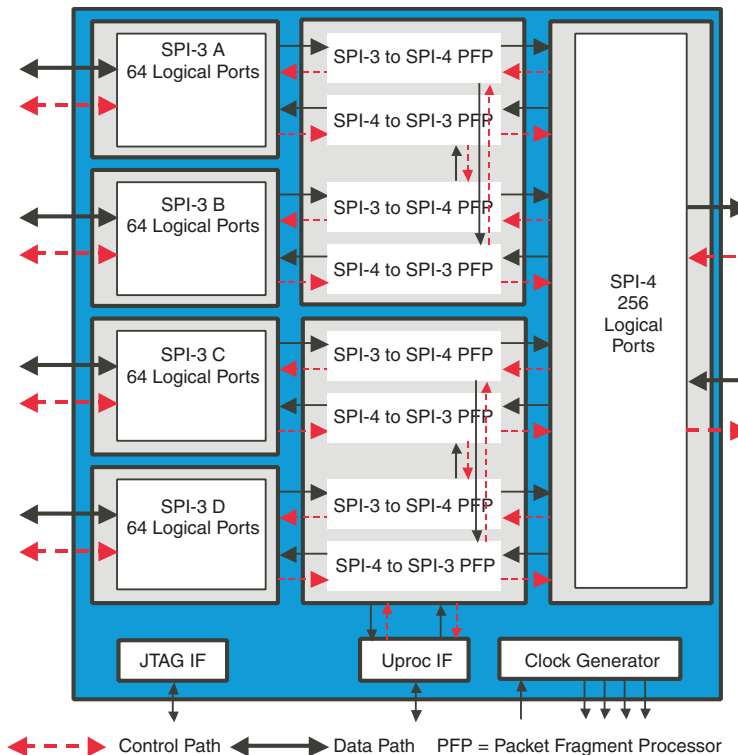
**APPLICATIONS**

- Ethernet transport
- SONET / SDH packet transport line cards
- Broadband aggregation
- Multi-service switches
- IP services equipment

**DESCRIPTION**

The IDT88P8344 is a SPI (System Packet Interface) Exchange with four SPI-3 interfaces and one SPI-4 interface. The data that enter on the low speed interface (SPI-3) are mapped to logical identifiers (LIDs) and enqueued for transmission over the high speed interface (SPI-4). The data that enter on the high speed interface (SPI-4) are mapped to logical identifiers (LIDs) and enqueued for transmission over a low speed interface (SPI-3). A data flow between SPI-3 and SPI-4 interfaces is accomplished with LID maps. The logical port addresses and number of entries in the LID maps may be dynamically configured. Various parameters of a data flow may be configured by the user such as buffer memory size and watermarks. In a typical application, the IDT88P8344 enables connection of multiple SPI-3 devices to a SPI-4 network processor. In other applications SPI-3 or SPI-4 devices may be connected to multiple SPI-3 network processors or traffic managers.

**HIGH LEVEL FUNCTIONAL BLOCK DIAGRAM**



6370 drw01

IDT and the IDT logo are trademarks of Integrated Device Technology, Inc

**INDUSTRIAL TEMPERATURE RANGE**

**APRIL 2006**

# Table of Contents

<b>Features</b> .....	<b>1</b>
<b>Applications</b> .....	<b>1</b>
<b>1. Introduction</b> .....	<b>8</b>
<b>2. Pin description</b> .....	<b>9</b>
<b>3. External interfaces</b> .....	<b>13</b>
3.1 SPI-3 .....	13
3.1.1 SPI-3 ingress .....	13
3.1.2 SPI-3 egress .....	15
3.2 SPI-4 .....	17
3.2.1 SPI-4 ingress .....	17
3.2.2 SPI-4 egress .....	20
3.2.3 SPI-4 startup handshake .....	20
3.3 Microprocessor interface .....	22
<b>4. Datapath and flow control</b> .....	<b>23</b>
4.1 SPI-3 to SPI-4 datapath and flow control .....	25
4.2 SPI-4 to SPI-3 datapath and flow control .....	30
4.3 SPI-3 ingress to SPI-3 egress datapath .....	33
4.4 Microprocessor interface to SPI-3 datapath .....	34
4.4.1 SPI-3 to ingress microprocessor interface datapath .....	34
4.4.2 Microprocessor insert to SPI-3 egress datapath .....	35
4.4.3 Microprocessor interface to SPI-4 egress datapath .....	36
4.4.4 SPI-4 ingress to microprocessor interface datapath .....	37
<b>5. Performance monitor and diagnostics</b> .....	<b>38</b>
5.1 Mode of operation .....	38
5.2 Counters .....	38
5.2.1 LID associated event counters .....	38
5.2.2 Non - LID associated event counters .....	38
5.3 Captured events .....	38
5.3.1 Non LID associated events .....	38
5.3.2 LID associated events .....	38
5.3.2.1 Non critical events .....	38
5.3.2.2 Critical events .....	38
5.3.3 Timebase .....	38
5.3.3.1 Internally generated timebase .....	38
5.3.3.2 Externally generated timebase .....	38
<b>6. Clock generator</b> .....	<b>39</b>
<b>7. Loopbacks</b> .....	<b>40</b>
7.1 SPI-3 Loopback .....	40
<b>8. Operation guide</b> .....	<b>41</b>
8.1 Hardware operation .....	41
8.1.1 System reset .....	41
8.1.2 Power on sequence .....	41
8.1.3 Clock domains .....	41
8.2 Software operation .....	41
8.2.1 Chip configuration sequence .....	41
8.2.2 Logical Port activation and deactivation .....	42
8.2.3 Buffer segment modification .....	42
8.2.4 Manual SPI-4 ingress LVDS bit alignment .....	42
8.2.5 SPI-4 status channel software .....	43
8.2.6 IDT88P8344 layout guidelines .....	43
8.2.7 Software Eye-Opening Check on SPI-4 Interface .....	44
<b>9. Register description</b> .....	<b>46</b>
9.1 Register access summary .....	46
9.1.1 Direct register format .....	46
9.1.2 Indirect register format .....	46
9.2 Direct access registers .....	50

## Table of Contents (Continued)

9.3 Indirect registers for SPI-3A, SPI-3B, SPI-3C, SPI-3D modules .....	56
9.3.1 Block base 0x0000 registers .....	57
9.3.2 Block base 0x0200 registers .....	57
9.3.3 Block base 0x0500 registers .....	58
9.3.4 Block base 0x0700 registers .....	59
9.3.5 Block base 0x0A00 registers .....	61
9.3.6 Block base 0x0C00 registers .....	61
9.3.7 Block base 0x1000 registers .....	64
9.3.8 Block base 0x1100 registers .....	64
9.3.9 Block base 0x1200 registers .....	64
9.3.10 Block base 0x1300 registers .....	65
9.3.11 Block base 0x1600 registers .....	66
9.3.12 Block base 0x1700 registers .....	66
9.3.13 Block base 0x1800 registers .....	66
9.3.14 Block base 0x1900 registers .....	67
9.4 Common module indirect registers (Module_base 0x8000) .....	68
9.4.1 Common module block base 0x0000 registers .....	69
9.4.2 Common module block base 0x0100 registers .....	69
9.4.3 Common module block base 0x0200 registers .....	69
9.4.4 Common module block base 0x0300 registers .....	69
9.4.5 Common module block base 0x0400 registers .....	72
9.4.6 Common module block base 0x0500 registers .....	72
9.4.7 Common module block base 0x0600 registers .....	73
9.4.8 Common module block base 0x0700 registers .....	73
9.4.9 Common module block base 0x0800 registers .....	75
<b>10. JTAG interface .....</b>	<b>79</b>
<b>11. Electrical and Thermal Specifications .....</b>	<b>79</b>
11.1 Absolute maximum ratings .....	79
11.2 Recommended Operating Conditions .....	79
11.3 Terminal Capacitance .....	80
11.4 Thermal Characteristics .....	80
11.5 DC Electrical characteristics .....	81
11.6 AC characteristics .....	82
11.6.1 SPI-3 I/O timing .....	82
11.6.2 SPI-4 LVDS Input / Output .....	83
11.6.3 SPI-4 LVTTTL Status AC characteristics .....	84
11.6.4 REF_CLK clock input .....	84
11.6.5 MCLK internal clock and OCLK[3:0] clock outputs .....	84
11.6.6 Microprocessor interface .....	84
11.6.6.1 Microprocessor parallel port AC timing specifications .....	85
11.6.6.2 Serial microprocessor interface (serial peripheral interface mode) .....	89
<b>12. Mechanical characteristics .....</b>	<b>90</b>
12.1 Device overview .....	90
12.2 Pin name/ball location table .....	91
12.3 Device package .....	94
<b>13. Glossary .....</b>	<b>96</b>
<b>14. Datasheet Document Revision History .....</b>	<b>97</b>
<b>15. Ordering information .....</b>	<b>98</b>

## List of Figures

Figure 1. Typical application: NPU, PHY, and co-processor .....	8
Figure 2. Data path diagram .....	8
Figure 3. Link mode SPI-3 ingress interface .....	14
Figure 4. PHY mode SPI-3 ingress interface .....	14
Figure 5. Link mode SPI-3 egress interface .....	16
Figure 6. PHY mode SPI-3 egress interface .....	16
Figure 7. Data sampling diagram .....	18
Figure 8. SPI-4 ingress state diagram .....	19
Figure 9. SPI-4 egress status state diagram .....	21
Figure 10. Interrupt scheme .....	22
Figure 11. Definition of data flows .....	23
Figure 12. Logical view of datapath configuration using PFPs .....	24
Figure 13. SPI-3 ingress to SPI-4 egress packet fragment processor .....	25
Figure 14. SPI-3 ingress LP to LID map .....	27
Figure 15. SPI-4 egress LID to LP map .....	28
Figure 16. SPI-3 ingress to SPI-4 egress datapath .....	28
Figure 17. SPI-3 ingress to SPI-4 egress flow control path .....	29
Figure 18. SPI-4 ingress to SPI-3 egress packet fragment processor .....	30
Figure 19. SPI-4 ingress to SPI-3 egress datapath .....	31
Figure 20. SPI-4 ingress to SPI-3 egress flow control .....	32
Figure 21. SPI-3 ingress to SPI-3 egress datapath .....	33
Figure 22 . Microprocessor data capture buffer .....	34
Figure 23. SPI-3 ingress to microprocessor capture interface datapath .....	34
Figure 25. Microprocessor interface to SPI-3 egress detailed datapath diagram .....	35
Figure 24 . Microprocessor data insert buffer .....	35
Figure 26. Microprocessor data insert buffer .....	36
Figure 27. Microprocessor data insert interface to SPI-4 egress datapath .....	36
Figure 28. Microprocessor data capture buffer .....	37
Figure 29. SPI-4 ingress to microprocessor data capture interface path .....	37
Figure 30. Clock generator .....	39
Figure 31. SPI-3 Loopback diagram .....	40
Figure 32. Power-on-Reset Sequence .....	41
Figure 33. DDR interface and eye opening check through over sampling .....	44
Figure 34. Direct & indirect access .....	46
Figure 35. SPI-3 I/O timing diagram .....	82
Figure 36. SPI-4 I/O timing diagram .....	83
Figure 37. Microprocessor parallel port Motorola read timing diagram .....	85
Figure 38. Microprocessor parallel port Motorola write timing diagram .....	86
Figure 39. Microprocessor parallel port Intel mode read timing diagram .....	87
Figure 40. Microprocessor parallel port Intel mode write timing diagram .....	88
Figure 41. Microprocessor serial peripheral interface timing diagram .....	89
Figure 42. IDT88P8344 820PBGA package, bottom view .....	94
Figure 43. IDT88P8344 820PBGA package, top and side views .....	95

## List of Tables

Table 1 – I/O types .....	9
Table 2 – SPI-3 ingress interface pin definition .....	9
Table 3 – SPI-3 egress interface pin definition .....	10
Table 4 – SPI-3 status interface pin definition .....	10
Table 5 – SPI-4 ingress interface definition .....	11
Table 6 – SPI-4 egress interface definition .....	11
Table 7 – Parallel microprocessor interface .....	12
Table 8 – Serial microprocessor interface (serial peripheral interface mode) .....	12
Table 9 – Miscellaneous .....	12
Table 10 – Both attached devices start from reset status .....	20
Table 11 – Ingress out of synch, egress in synch .....	20
Table 12 – Ingress in synch, egress out of synch .....	20
Table 13 - DIRECTION code assignment .....	26
Table 14 – CK_SEL[3:0] input pin encoding .....	39
Table 15 - Zero margin SPI-3 timing budget .....	43
Table 16 - Margin check for SPI-3 timing .....	43
Table 17 - Bit order within an 8-Bit data register .....	46
Table 18 - Bit order within a 32-Bit data register .....	46
Table 19 - Bit order within an 8-Bit data register .....	46
Table 20 - Bit order within a 16-Bit address register .....	47
Table 21 - Bit order within an 8-Bit control register .....	47
Table 22 - Module base address (Module_base) .....	47
Table 23 - Indirect access block bases for Module A, Module B, Module C, and Module D .....	47
Table 24 - Indirect access block bases for common module .....	48
Table 25 - Indirect access data registers (direct accessed space) at 0x30 to 0x33 .....	48
Table 26 - Indirect access address register (direct accessed space) at 0x34 to 0x35 .....	48
Table 27 - Indirect access control register (direct accessed space) at 0x3F .....	48
Table 28 - Error coding table .....	49
Table 29 - Direct mapped Module A, Module B, Module C, and Module D registers .....	50
Table 30 - Direct mapped other registers .....	50
Table 31 - SPI-3 data capture control register (registers 0x00, 0x08, 0x10, 0x18) .....	50
Table 32 - SPI-3 data Capture register (registers 0x01, 0x09, 0x11, 0x19) .....	50
Table 33 - SPI-4 data insert control register (registers 0x02, 0x0A, 0x12, 0x1A) .....	51
Table 34 - SPI-4 data insert register (registers 0x03, 0x0B, 0x13, 0x1B) .....	51
Table 35 - SPI-4 data capture control register (registers 0x04, 0x0C, 0x14, 0x1C) .....	51
Table 36 - SPI-3 data insert control register (registers 0x05, 0x0D, 0x15, 0x1D) .....	51
Table 37 - SPI-4 data capture register (registers 0x06, 0x0E, 0x16, 0x1E) .....	51
Table 38 - SPI-3 data insert register (registers 0x07, 0x0F, 0x17, 0x1F) .....	51
Table 39 - Software reset register (0x20 in the direct accessed space) .....	52
Table 40 - SPI-4 status register (0x22 in the direct accessed space) .....	52
Table 41 - SPI-4 enable register (0x23 in the direct accessed space) .....	52
Table 42 - Module status register (0x24 to 0x27 in the direct accessed space) .....	53
Table 43 - Module enable register (0x28 to 0x2B in the direct accessed space) .....	53
Table 44 - Primary interrupt status register (0x2C in the direct accessed space) .....	54
Table 45 - Secondary interrupt status register (0x2D in the direct accessed space) .....	54
Table 46 - Primary interrupt enable register (0x2E in the direct accessed space) .....	55
Table 47 - Secondary interrupt enable register (0x2F in the direct accessed space) .....	55
Table 48 - Module A/B/C/D indirect register .....	56
Table 49 - SPI-3 ingress LP to LID map .....	57
Table 50 - SPI-3 general configuration register (register_offset=0x00) .....	57
Table 51 - SPI-3 ingress configuration register (register_offset=0x01) .....	58
Table 52 - SPI-3 ingress fill level register (register_offset=0x02) .....	58
Table 53 - SPI-3 ingress max fill level register (register_offset=0x03) .....	58
Table 54 - SPI-3 egress LID to LP map .....	58
Table 55 - SPI-3 egress configuration register (register_offset=0x00) .....	59
Table 56 - SPI-4 ingress to SPI-3 egress flow control register (register_offset=0x01) .....	59

## List of Tables (Continued)

Table 57 - SPI-3 egress test register (register_offset=0x02) .....	59
Table 58 - SPI-3 egress fill level register (register_offset=0x03) .....	60
Table 59 - SPI-3 egress max fill level register (register_offset=0x04) .....	60
Table 60 - LID associated event counters (0x000-0x17F) .....	61
Table 61 - Non LID associated event counters (0x00 - 0x0B) .....	61
Table 62 - Non LID associated interrupt indication register (register_offset 0x0c) .....	62
Table 63 - Non LID associated interrupt enable register(register_offset 0x0D) .....	62
Table 64 - LID associated interrupt indication register (register_offset 0x0E) .....	62
Table 65 - LID associated interrupt enable register (register_offset 0x0F) .....	62
Table 66 - Non critical LID associated capture table (register_offset 0x10-0x15) .....	63
Table 67 - SPI-3 to SPI-4 critical LID interrupt indication registers (register_offset 0x16-0x17) .....	63
Table 68 - SPI-3 to SPI-4 critical LID interrupt enable registers (register_offset 0x18-0x19) .....	63
Table 69 - SPI-4 to SPI-3 critical LID interrupt indication registers (register_offset 0x1A-0x1B) .....	63
Table 70 - SPI-4 to SPI-3 critical LID interrupt enable registers (register_offset 0x1C-0x1D) .....	63
Table 71 - Critical events source indication register (register_offset 0x1E) .....	63
Table 72 - SPI-3 ingress packet length configuration register .....	64
Table 73 - SPI-4 egress port descriptor table (64 entries) .....	64
Table 74 - SPI-4 egress DIRECTION code assignment .....	64
Table 75 - SPI-3 ingress port descriptor table (Block_base 0x1200 + Register_offset 0x00-0x3F) .....	64
Table 76 - SPI-3 to SPI-4 PFP register (register_offset 0x00) .....	65
Table 77 - NR_LID field encoding .....	65
Table 78 - SPI-3 to SPI-4 flow control register (register_offset 0x01) .....	65
Table 79 - SPI-4 ingress packet length configuration (64 entries) .....	66
Table 80 - SPI-3 egress port descriptor table (64 entries) .....	66
Table 81 - SPI-3 egress DIRECTION code assignment .....	66
Table 82 - SPI-4 ingress port descriptor table (64 entries) .....	66
Table 83 - SPI-4 to SPI-3 PFP register (0x00) .....	67
Table 84 - NR_LID field encoding .....	67
Table 85 -Common Module (Module_base 0x8000) indirect register table .....	68
Table 86 - SPI-4 ingress LP to LID map (256 entries, one per LP) .....	69
Table 87 - SPI-4 ingress calendar_0 (256 entries) .....	69
Table 88 - SPI-4 ingress calendar_1 (256 entries) .....	69
Table 89 - SPI-4 ingress configuration register (0x00) .....	69
Table 90 - SPI-4 ingress status configuration register (register_offset 0x01) .....	70
Table 91 - SPI-4 ingress status register (register_offset 0x02) .....	70
Table 92 - SPI-4 ingress inactive transfer port (register_offset 0x03) .....	70
Table 93 - SPI-4 ingress calendar configuration register (0x04 to 0x05) .....	71
Table 94 - SPI-4 ingress watermark register (register_offset 0x06) .....	71
Table 95 - SPI-4 ingress fill level register (0x07 to 0x0A) .....	71
Table 96 - SPI-4 ingress max fill level register (0x0B to 0x0E) .....	71
Table 97 - SPI-4 ingress diagnostics register (register_offset 0x0F) .....	71
Table 98 - SPI-4 ingress DIP-4 error counter (register_offset 0x10) .....	72
Table 99 - SPI-4 ingress bit alignment control register (register_offset 0x11) .....	72
Table 100 - SPI-4 ingress start up training threshold register (register_offset 0x12) .....	72
Table 101 - SPI-4 egress LID to LP map (256 entries) .....	72
Table 102 - SPI-4 egress calendar_0 (256 locations) .....	72
Table 103 - SPI-4 egress calendar_1 (256 locations) .....	73
Table 104 - SPI-4 egress configuration register_0 (register_offset 0x00) .....	73
Table 105 - SPI-4 egress configuration register_1 (register_offset 0x01) .....	73
Table 106 - SPI-4 egress status register (register_offset 0x02) .....	74
Table 107 - SPI-4 egress calendar configuration register (Register_offset 0x03 - 0x04) .....	74
Table 108 - SPI-4 egress diagnostics register (register_offset 0x05) .....	74
Table 109 - SPI-4 egress DIP-2 error counter (register_offset 0x06) .....	74
Table 110 - SPI-4 ingress bit alignment window register (register_offset 0x00) .....	75
Table 111 - SPI-4 ingress lane measure register (register_offset 0x01) .....	75

## List of Tables (Continued)

Table 112 - SPI-4 ingress bit alignment counter register (0x02 to 0x0B) .....	75
Table 113 - SPI-4 ingress manual alignment phase/result register (0x0C to 0x1F) .....	75
Table 114 - SPI-4 Egress data lane timing register (register_offset 0x2A) .....	75
Table 115 - SPI-4 egress Control Lane Timing register (Register_offset 0x2B) .....	76
Table 116 - SPI-4 egress data clock timing register (register_offset 0x2C) .....	76
Table 117 - SPI-4 egress status timing register (register_offset 0x2D) .....	76
Table 118 - SPI-4 egress status clock timing register (register_offset 0x2E) .....	76
Table 119 - PMON timebase control register (register_offset 0x00) .....	77
Table 120 - Timebase register (register_offset 0x01) .....	77
Table 121 - Clock generator control register (register_offset 0x10) .....	77
Table 122 - OCLK and MCLK frequency select encoding .....	77
Table 123 - GPIO register (register_offset 0x20) .....	78
Table 124 - GPIO monitor table (5 entries 0x21-0x25 for GPIO[0] through GPIO[4]) .....	78
Table 125 - Version number register (register_offset 0x30) .....	78
Table 126 - JTAG instructions .....	79
Table 127 - Absolute maximum ratings .....	79
Table 128 - Recommended Operating Conditions .....	79
Table 129 - Terminal Capacitance .....	80
Table 130 - Thermal Characteristics .....	80
Table 131 - DC Electrical characteristics .....	81
Table 132 - SPI-3 AC Input / Output timing specifications .....	82
Table 133 - SPI-4.2 LVDS AC Input / Output timing specifications .....	84
Table 134 - SPI-4 LVTTTL status AC Characteristics .....	84
Table 135 - REF_CLK clock input .....	84
Table 136 - OCLK[3:0] clock outputs and MCLK internal clock .....	84
Table 137 - Microprocessor interface .....	84
Table 138 - Microprocessor parallel port Motorola read timing .....	85
Table 139 - Microprocessor parallel port Motorola write timing .....	86
Table 140 - Microprocessor parallel port Intel mode read timing .....	87
Table 141 - Microprocessor parallel port Intel mode write timing .....	88
Table 142 - Microprocessor serial peripheral interface timing .....	89

# TYPICAL APPLICATION

Exchange between secure traffic, clear traffic, 10G NPU and co-processor

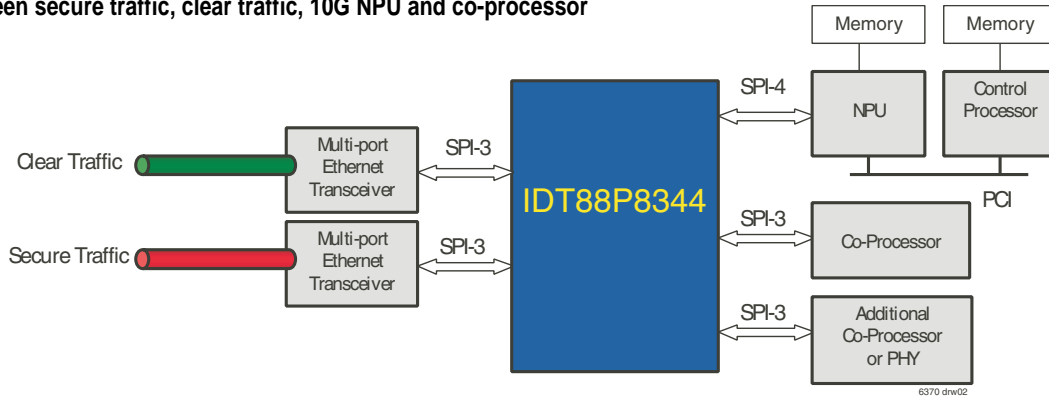


Figure 1. Typical application: NPU, PHY, and co-processor

## 1. INTRODUCTION

The IDT88P8344 device is a quad SPI-3 to single SPI-4 exchange with switching capabilities intended for use in VPN firewall cards, Ethernet transport, and multi-service switches. The SPI-3 and SPI-4 interfaces are defined by the Optical Internetworking Forum (OIF).

The device can be used as an exchange, a switch, or an aggregation device between network processor units, multi-gigabit framers and PHYs, and switch fabric interface devices.

### Data Path Overview

Figure 2. Data Path Diagram shows an overview of the data path through the device.

In normal operation, there are two paths through the IDT88P8344 device: the quad SPI-3 ingress to SPI-4 egress path, and the SPI-4 ingress to quad SPI-3 egress path. SPI-3 and SPI-4 burst sizes are separately configurable.

In the SPI-3 ingress to SPI-4 egress path, data enter in fragments on the

SPI-3 interface and are received by the SPI-3 interface block. The fragments are mapped to a SPI-4 address and stored in memory allocated at the SPI-3 level until such a time that the SPI-3 to SPI-4 packet fragment processor determines that they are to be transmitted on the SPI-4 interface. The data is transferred in bursts, in line with the OIF SPI-4 implementation agreement, to the SPI-4 interface block, and are transmitted on the SPI-4 interface.

In the SPI-4 ingress to SPI-3 egress path, data enter in bursts on the SPI-4 interface and are received by the SPI-4 interface block. The SPI-4 address is translated to a SPI-3 address, and the data contained in the bursts are stored in memory allocated at the SPI-3 level until such a time that the SPI-4 to SPI-3 packet fragment processor determines that they are to be transmitted on the SPI-3 interface. The data is transferred in packet fragments, in line with the OIF SPI-3 implementation agreement, to the SPI-3 interface block, and are transmitted on the SPI-3 interface.

These and additional data paths are described in more detail in the data path section of this document.

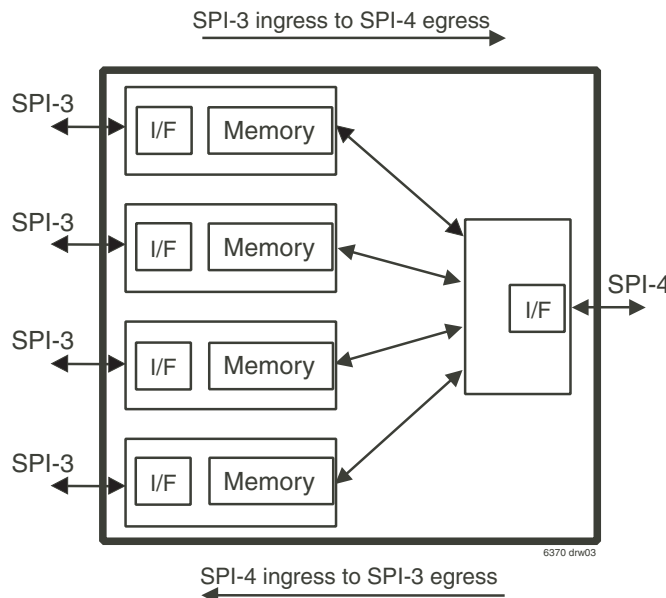


Figure 2. Data path diagram



## 2. PIN DESCRIPTION

### SPI-3 (four instantiations)

For the SPI-3 interfaces, each pin is used differently depending whether the SPI-3 is in Link mode or in PHY mode. Each of the SPI-3 interfaces is separately

configurable for either Link or PHY mode. This configuration holds for both the ingress and egress paths. The device pin is given a generic name, and mapped to the standard pin name according to the mode of the interface (Link or PHY).

**TABLE 1 – I/O TYPES**

I/O type	Function
I-ST	Input with Schmitt trigger with weak pull up
I-PU	Input with weak pull up
B-PU	Bidirectional I/O with weak pull up
I-PD	Input with pull down
I	Input
O	Output
O-Z	Output with tri-state
OD	Output with open drain

**TABLE 2 – SPI-3 INGRESS INTERFACE PIN DEFINITION**

Generic Name	Specific Name	I/O type	Description	Mode	
				Link	PHY
I_FCLK	SPI3A_I_FCLK SPI3B_I_FCLK SPI3C_I_FCLK SPI3D_I_FCLK	I-ST LVTTTL	Ingress SPI-3 write clock	RFCLK	TFCLK
RVAL	SPI3A_I_RVAL SPI3B_I_RVAL SPI3C_I_RVAL SPI3D_I_RVAL	B-PU LVTTTL	Receive data valid	RVAL (I)	RVAL (O)
I_ENB	SPI3A_I_ENB SPI3B_I_ENB SPI3C_I_ENB SPI3D_I_ENB	B-PU LVTTTL	Ingress read enable	RENB (O)	TENB (I)
I_DAT[31:0]	SPI3A_I_DAT[31:0] SPI3B_I_DAT[31:0] SPI3C_I_DAT[31:0] SPI3D_I_DAT[31:0]	I-PU LVTTTL	Ingress data bus	RDAT [31:0]	TDAT [31:0]
I_MOD[1:0]	SPI3A_I_MOD[1:0] SPI3B_I_MOD[1:0] SPI3C_I_MOD[1:0] SPI3D_I_MOD[1:0]	I-PU LVTTTL	Ingress word modulus	RMOD [1:0]	TMOD [1:0]
I_PRTY	SPI3A_I_PRTY SPI3B_I_PRTY SPI3C_I_PRTY SPI3D_I_PRTY	I-PU LVTTTL	Ingress parity	RPRTY	TPRTY
I_SOP	SPI3A_I_SOP SPI3B_I_SOP SPI3C_I_SOP SPI3D_I_SOP	I-PU LVTTTL	Ingress start of packet	RSOP	TSOP
I_EOP	SPI3A_I_EOP SPI3B_I_EOP SPI3C_I_EOP SPI3D_I_EOP	I-PU LVTTTL	Ingress end of packet	REOP	TEOP
I_ERR	SPI3A_I_ERR SPI3B_I_ERR SPI3C_I_ERR SPI3D_I_ERR	I-PU LVTTTL	Ingress EOP error	RERR	TERR
I_SX	SPI3A_I_SX SPI3B_I_SX SPI3C_I_SX SPI3D_I_SX	I-PU LVTTTL	Ingress start of transfer	RSX	TSX

**TABLE 3 – SPI-3 EGRESS INTERFACE PIN DEFINITION**

Generic Name	Specific Name	I/O type	Description	Mode	
				Link	PHY
E_FCLK	SPI3A_E_FCLK SPI3B_E_FCLK SPI3C_E_FCLK SPI3D_E_FCLK	I-ST LVTTTL	Egress SPI-3 write clock	TFCLK	RFCLK
E_ENB	SPI3A_E_ENB SPI3B_E_ENB SPI3C_E_ENB SPI3D_E_ENB	B-PU LVTTTL	Egress read enable	TENB (O)	RENB (I)
E_DAT[31:0]	SPI3A_E_DAT[31:0] SPI3B_E_DAT[31:0] SPI3C_E_DAT[31:0] SPI3D_E_DAT[31:0]	O-Z LVTTTL	Egress data bus	TDAT [31:0]	RDAT [31:0]
E_MOD[1:0]	SPI3A_E_MOD[1:0] SPI3B_E_MOD[1:0] SPI3C_E_MOD[1:0] SPI3D_E_MOD[1:0]	O-Z LVTTTL	Egress word modulus	TMOD [1:0]	RMOD [1:0]
E_PRTY	SPI3A_E_PRTY SPI3B_E_PRTY SPI3C_E_PRTY SPI3D_E_PRTY	O-Z LVTTTL	Egress parity	TPRTY	RPRTY
E_SOP	SPI3A_E_SOP SPI3B_E_SOP SPI3C_E_SOP SPI3D_E_SOP	O-Z LVTTTL	Egress start of packet	TSOP	RSOP
E_EOP	SPI3A_E_EOP SPI3B_E_EOP SPI3C_E_EOP SPI3D_E_EOP	O-Z LVTTTL	Egress end of packet	TEOP	REOP
E_ERR	SPI3A_E_ERR SPI3B_E_ERR SPI3C_E_ERR SPI3D_E_ERR	O-Z LVTTTL	Egress EOP error	TERR	RERR
E_SX	SPI3A_E_SX SPI3B_E_SX SPI3C_E_SX SPI3D_E_SX	O-Z LVTTTL	Egress start of transfer	TSX	RSX

**TABLE 4 – SPI-3 STATUS INTERFACE PIN DEFINITION**

Generic Name	Specific Name	I/O type	Description	Mode	
				Link	PHY
DTPA[3:0]	SPI3A_DTPA[3:0] SPI3B_DTPA[3:0] SPI3C_DTPA[3:0] SPI3D_DTPA[3:0]	B-PU LVTTTL	Direct transmit packet available	DTPA (I)	DTPA (O)
STPA	SPI3A_STPA SPI3B_STPA SPI3C_STPA SPI3D_STPA	B-PU LVTTTL	Selected-PHY transmit packet available	STPA (I)	STPA (O)
PTPA	SPI3A_PTPA SPI3B_PTPA SPI3C_PTPA SPI3D_PTPA	B-PU LVTTTL	Polled-PHY transmit packet available	PTPA (I)	PTPA (O)
ADR[7:0]	SPI3A_ADR[7:0] SPI3B_ADR[7:0] SPI3C_ADR[7:0] SPI3D_ADR[7:0]	B-PU LVTTTL	Polled transmit PHY address	ADR (O)	ADR (I)

**SPI-4 (one instantiation)**

For the SPI-4 interface, each pin is used differently depending whether the SPI-4 is in Link mode or in PHY mode. The pin is given a generic name, shown

in the Name column, and mapped to the OIF standard pin name according to the mode of operation of the interface (Link to PHY).

**TABLE 5 – SPI-4 INGRESS INTERFACE DEFINITION**

Generic Name	Specific Name	I/O type	Description	Mode	
				Link	PHY
I_DCLK (P & N)	SPI4_I_DCLK_P SPI4_I_DCLK_N	I LVDS	Ingress data clock	RDCLK	TDCLK
I_DAT[15:0] (P & N)	SPI4_I_DAT_P[15:0] SPI4_I_DAT_N[15:0]	I LVDS	Ingress data bus	RDAT	TDAT
I_CTRL (P & N)	SPI4_I_CTRL_P SPI4_I_CTRL_N	I LVDS	Ingress control word	RCTL	TCTL
I_SCLK_L (P & N)	SPI4_I_SCLK_P SPI4_I_SCLK_N	O LVDS	Ingress status clock	RSCLK	TSCLK
I_STAT_L[1:0] (P & N)	SPI4_I_STAT_P[1:0] SPI4_I_STAT_N[1:0]	O LVDS	Ingress status info	RSTAT	TSTAT
I_SCLK_T	SPI4_I_SCLK_T	O LVTTTL	Ingress status clock	RSCLK	TSCLK
I_STAT_T[1:0]	SPI4_I_STAT_T[1:0]	O LVTTTL	Ingress status info	RSTAT	TSTAT
BIAS	BIAS	Analog	Use an external 3K Ohm 1% resistor to VSS	-----	-----
LVDS_STA	LVDS_STA	I-PU	LVDS(high)/LVTTTL (low) status selection (See Note below)	-----	-----

**NOTE:**

1. A hardware reset or software reset must be performed after changing the level of this pin.

**TABLE 6 – SPI-4 EGRESS INTERFACE DEFINITION**

Generic Name	Specific Name	I/O type	Description	Mode	
				Link	PHY
E_DCLK (P & N)	SPI4_E_DCLK_P SPI4_E_DCLK_N	O LVDS	Egress data clock	TDCLK	RDCLK
E_DAT[15:0] (P & N)	SPI4_E_DAT_P[15:0] SPI4_E_DAT_N[15:0]	O LVDS	Egress data bus	TDAT[15:0]	RDAT[15:0]
E_CTRL (P & N)	SPI4_E_CTRL_P SPI4_E_CTRL_N	O LVDS	Egress control word	TCTL	RCTL
E_SCLK_L (P & N)	SPI4_E_SCLK_P SPI4_E_SCLK_N	I LVDS	Egress status clock	TSCLK	RSCLK
E_STAT_L[1:0] (P & N)	SPI4_E_STAT_P[1:0] SPI4_E_STAT_N[1:0]	I LVDS	Egress status info	TSTAT[1:0]	RSTAT[1:0]
E_SCLK_T	SPI4_E_SCLK_T	I-ST LVTTTL	Egress status clock	TSCLK	RSCLK
E_STAT_T[1:0]	SPI4_E_STAT_T[1:0]	I-PU LVTTTL	Egress status info	TSTAT	RSTAT[1:0]

## Parallel microprocessor Interface

The Parallel microprocessor interface is configurable to work in Intel or Motorola modes. Be sure to connect SPI\_EN to a logic low when using the parallel microprocessor interface mode.

**TABLE 7 – PARALLEL MICROPROCESSOR INTERFACE**

Name	I/O type	Description
MPM	I-PU CMOS	Microprocessor mode: 0=Motorola Mode, 1=Intel mode (sampled after reset)
CSB	I-ST CMOS	Chip select; active low
RDB	I-ST CMOS	RDB: Read control, active low (in Intel mode), or DSB: Data strobe, active low (in Motorola mode)
WRB	I-ST CMOS	WRB: Write control; active low; (in Intel mode), or R/WB: Read/write control; when high, read is active; when low, write is active; (in Motorola mode)
ADD[5:0]	I-PU CMOS	Address bus
DBUS[7:0]	B-PU CMOS	Data bus
INTB	OD CMOS	Interrupt, active low, open drain
SPI_EN	I-PU CMOS	Logic low selects parallel microprocessor interface (internally pulled up, sampled after reset)

**TABLE 8 – SERIAL MICROPROCESSOR INTERFACE (SERIAL PERIPHERAL INTERFACE MODE)**

Four pins multiplexed with parallel microprocessor pins. Be sure to connect SPI\_EN to a logic high when using the serial microprocessor interface mode.

Name	I/O type	Parallel microprocessor pin used	Serial Peripheral Interface Pin Use Description
SDI	I-ST CMOS	WRB	Serial data in, rise edge sampling
SDO	B-PU CMOS	DBUS[0]	Serial data out, falling edge driving
CSB	I-ST CMOS	CSB	Chip select, active low. SDO is tri-stated when CSB is high
SCLK	I-ST CMOS	RDB	Input clock
INTB	OD CMOS	-----	Interrupt, active low, open drain
SPI_EN	I-PU CMOS	-----	Dedicated input. High selects SPI microprocessor interface (internally pulled up)

**TABLE 9 – MISCELLANEOUS**

Name	I/O type	Description
REF_CLK	I-ST CMOS	Master clock input
OCLK[3:0]	O LVTTTL	Clock outputs that can be used for SPI-3, phase-shifted to avoid simultaneously switching outputs
CLK_SEL[3:0]	I-PU CMOS	Clock select inputs for internal PLL, internal MCLK, and OCLK[3:0] outputs
TIMEBASE	B-PU CMOS	Timeout signal for counters
GPIO[4:0]	B-PU CMOS	General purpose I/O or internal state monitor pins
TDI	I-PU CMOS	JTAG data in (internally pulled up)
TDO	O-Z CMOS	JTAG data out
TCK	I-ST CMOS	JTAG clock
TMS	I-PU CMOS	JTAG mode (internally pulled up)
TRSTB	I-PU CMOS	JTAG reset, active low (internally pulled up). Pull down for normal operation.
RESETB	I-PD CMOS	Master hardware reset, active low

**NOTE:**

- Inputs with internal pull-ups do not need external pull-ups unless connected to PCB trace (except TRSTB).

## 3. EXTERNAL INTERFACES

The external interfaces provided on the IDT88P8344 device are four SPI-3 interfaces, one SPI-4 interface, a serial or parallel microprocessor interface, a JTAG interface, and a set of GPIO pins. Each of the interfaces is defined in the relevant standard.

The following information contains a set of the highlights of the features supported from the relevant standards, and a description of additional features implemented to enhance the usability of these interfaces for the system architect.

### 3.1 SPI-3

Refer to OIF SPI-3 document (see 13. Glossary for a reference) for full details of the implementation agreement.

- Four instantiations of SPI-3 interface; each interface independently configurable
- Device supports a 8-bit and 32-bit data bus structure.
- Clock rate is minimum 19.44 to maximum 133 MHz
- Link, single port PHY, and single device multi port PHY modes supported
- Byte level and packet level transfer control mechanisms supported
  - Four DTPA signals supported, mapped to LP addresses 0–3, for STPA in byte-level mode
  - Eight ADR signals supported for PTPA in packet-level mode
- Address range 0 to 255 with support for 64 simultaneously active logical ports
- Fragment length (section) configurable from 16 to 256 bytes in 16 byte multiples
- Configurable standard and non-standard bit ordering

### SPI-3 implementation features

The following are implemented per SPI-3 interface, and there are four instantiations per device.

- Link / PHY layer device
- Packet / byte level FIFO status information
- Physical port enable
- Width of data bus (32 bit or 8 bit)
- Parity selection (odd or even)
- Enable parity check

#### 3.1.1 SPI-3 ingress

The following are implemented per SPI-3 interface, and there are 4 instantiations per device.

- SPI-3 LP to Link Identifier (LID) map
- 256 entries, one per SPI-3 LP address
- LP enable control
- Only 64 of these entries are to be in the active state simultaneously

#### Backpressure enable

- Link mode only
- Enables the assertion of the I\_ENB when at least one active LID can not accept data
  - If not enabled, the I\_ENB signal will never be asserted in Link mode, possibly leading to fragments being discarded.

#### Minimum packet length

- Packets shorter than the minimum length will be optionally counted in the short packet counter.
- Range 0 – 255 in 1 byte increments

#### Maximum packet length

- Packets longer than the maximum length will be optionally counted in the long packet counter.
- Range 0 – 16,383 in 1 byte increments

#### Backpressure threshold

- Number of free segments allocated below which backpressure will be triggered for the LP

### SPI-3 ingress interface

Multiple independent data streams can be transmitted over the physical SPI-3 port. Each of those data streams is identified by a SPI-3 logical port (LP). Data from a transfer on a SPI-3 logical port and the associated descriptor fields are synchronized to the configurable internal buffer segment pool.

#### Normal operation

Refer to [13. Glossary] for details about the SPI-3 interface.

- A SPI-3 interface (a physical port) is enabled by the SPI-3\_ENABLE flag in the SPI-3 configuration register. A disabled interface tri-states all output pins and does not respond to any input signals.
- The interface is configured in PHY or Link layer mode by the LINK flag in the SPI-3 general configuration register.
- The interface supports a SPI-3 logical port number range [0..255], note that at most 64 logical ports can be configured.
- The SPI-3 interface supports data transport over either a 32 bit data interface or over one single 8 bit interface (data[7:0]) only. The selection is defined by the BUSWIDTH flag in the SPI-3 general configuration register.
- The SPI-3 interface is configured in byte mode or packet mode by the PACKET flag in the SPI-3 general configuration register.
- The SPI-3 interface supports over-clocking.
- Parity checking over data[31:0] is enabled by the PARITY\_EN flag in the Table 50, SPI-3 general configuration register (register\_offset=0x00). The parity type is defined by the EVEN\_PARITY flag. Parity check results over the in-band port address and the data of a transfer are forwarded towards the packet fragment processor.
- SPI Exchange supports zero clock interval spacing between transfers.

### SPI-3 ingress interface errors

Given an I\_FCLK within specification, the SPI-3 will not dead lock due to any combination or sequence on the SPI-3 interface. The SPI Exchange detects for incorrect SOP/EOP sequences on a logical port. The following sequences are detected:

- Successive SOP (SOP-SOP sequence rather than SOP-EOP-SOP-EOP)
- Successive EOP (EOP-EOP sequence rather than SOP-EOP-SOP-EOP)

Detection of an illegal sequence results in the generation of an SPI-3 illegal SOP sequence event or SPI-3 illegal EOP sequence even generated. The event is associated to the physical port. The event is directed towards the PMON & DIAG module.

A clock available process detects a positive I\_FCLK within a 64 MCLK clock cycle period. The result of this process is reported in the I\_FCLK\_AV flag in the Table 52 SPI-3 ingress fill level register (Block\_base 0x0200 + Register\_offset 0x02).

A status change from the clock available status to the clock not available status generates a maskable SPI-3 ingress clock unavailable interrupt indication, SPI3\_ICLK\_UN, in Table 62-Non LID associated interrupt indication register (Block\_Base 0x0C00 + Register\_offset 0x0C).

**SPI-3 ingress Link mode**

Refer to [Glossary] for details about the SPI-3 interface.

- The PHY pushes data into the device in blocks from 1 up to 256 bytes.
- The SPI Exchange provides backpressure for the SPI-3 ingress physical interface by the I\_ENB signal. The I\_ENB is asserted when at least one

active LID can not accept data. This feature is enabled by the BACKPRESSURE\_EN flag in the SPI-3 ingress configuration register (register\_offset = 0x01). When the flag is cleared the I\_ENB signal will not be asserted, hence no backpressure can be generated.

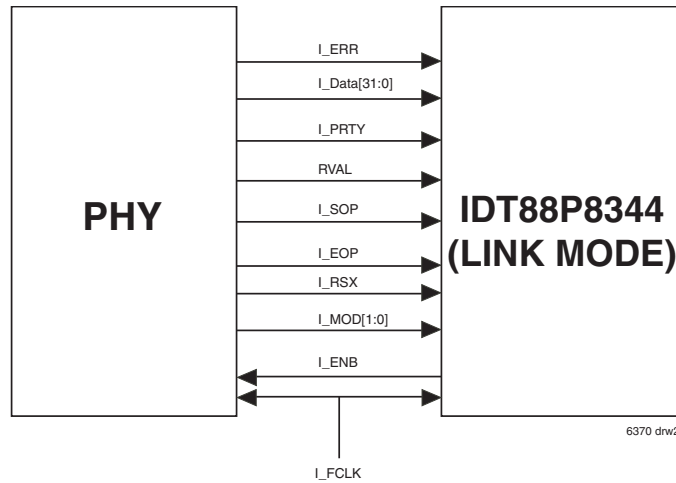


Figure 3. Link mode SPI-3 ingress interface

**SPI-3 ingress PHY mode**

The SPI Exchange indicates to the Link layer it has buffer space available by proper response to either Link layer polling (packet mode) or direct indication on DTPA signals (byte mode). The selection is made by the PACKET flag in the SPI-3 configuration register.

- In packet mode the device responds to polling (by Link layer device)
- In byte mode the direct status indication is limited to 4 addresses (fixed ports [3:0])

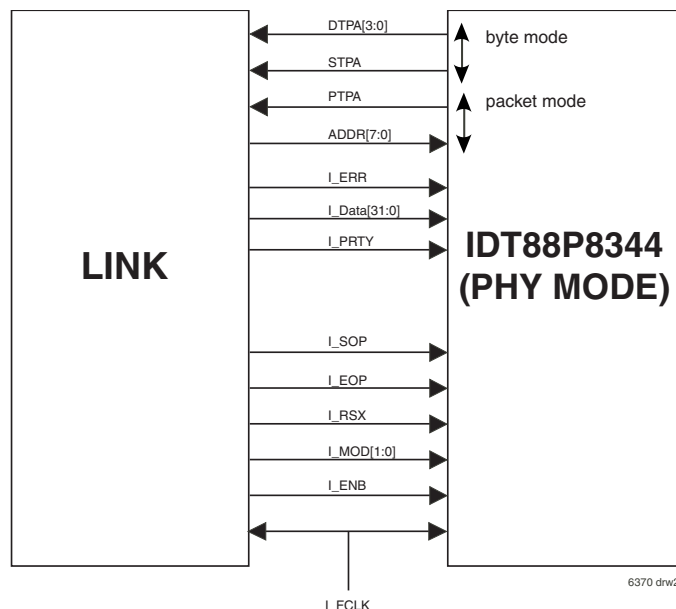


Figure 4. PHY mode SPI-3 ingress interface

### 3.1.2 SPI-3 egress

- All fragments will be of a programmable equal length with the exception of EOP fragment which may be shorter

#### **LID to LP map**

- 64 entries, one per LID, for each SPI-3 egress port
- LP enable control

#### **Multiple burst enable**

- Allows more than one burst to be sent to an LP.

#### **Poll length**

- For use when in Link mode and when using the packet level mode
- Causes polling of the PHY for the logical ports associated to LIDs ranging from [0 up to POLL\_LENGTH] to find logical ports that can accept data
- Range is 0-63

#### **Loopback enable**

- Enables loopback from SPI-3 physical interface to same SPI-3 physical interface for test purposes

#### **Data memory egress control**

The SPI-3 egress port descriptor table (block\_base 0x1700) for both paths out of the data memory. The function of the SPI-3 egress port descriptor table (block\_base 0x1700) is to define where data goes after exiting the main data memory. There are four options configurable:

- SPI-3
- SPI-4
- Capture
- Discard

#### **Maximum number of memory segments**

- Defines the largest BUFFER available to a LP / LID
- Each segment is 256 bytes
- Range 1 – 508 in increments of one segment

### SPI-3 egress interface configuration

- SPI Exchange allows for a pause at least two cycles of E\_FCLK between successive transfers.
- SPI Exchange allows for over clocking for a higher clock frequency supported as opposed to the one defined by the SPI-3 implementation agreement.
  - The Link mode is selected by the Link flag in the SPI-3 general configuration register.
  - The interface operates in PACKET mode or BYTE mode as defined by the PACKET flag in the SPI-3 general configuration register.
  - SPI Exchange generates even or odd parity over E\_DATA[7/31:0] on the E\_PRTY signal as defined by the EVEN flag in the Table 50, SPI-3 general configuration register (register\_offset=0x00).
  - SPI Exchange optionally generates two dummy cycles after assertion of the STX signal. The option is enabled by the STX\_SPACING flag in the Table 50, SPI-3 general configuration register (register\_offset=0x00).
  - SPI Exchange optionally generates two dummy cycles after assertion of an EOP signal. The option is enabled by the EOP\_SPACING flag in the Table 50, SPI-3 general configuration register (register\_offset=0x00).

### SPI-3 egress interface errors

A clock available process detects an E\_FCLK cycle within a 64 MCLK clock cycle period. The result of this process is reported in the E\_FCLK\_AV flag in Table 58, *SPI-3 egress fill level register (Block\_base 0x0700 + Register\_offset=0x03)*.

A status change from the clock available status to the clock not available status generates a maskable SPI-3 egress clock unavailable interrupt indication, SPI3\_ECLK\_UN, in Table 62-Non LID associated interrupt indication register (Block\_Base 0x0C00 + Register\_offset 0x0C).

**SPI-3 egress Link mode**

The SPI Exchange receives status information from the PHY. The PHY indicates its ability to receive data. Status information for all logical ports is directed towards the packet fragment processor.

Status information is received from the PHY.

- In packet mode, the SPI Exchange polls the PHY for the logical ports associated to LIDs ranging from 0 up to POLL\_LENGTH to find logical ports that

can accept data. The POLL\_LENGTH field is defined in the SPI-3 egress configuration register.

- In byte mode the SPI Exchange allows for direct status detection. This status information is directly forwarded to the packet fragment processor if enabled by the BURST\_EN flag. When the BURST\_EN flag is cleared the only one packet fragment per LP is allowed into the SPI-3 egress buffers.

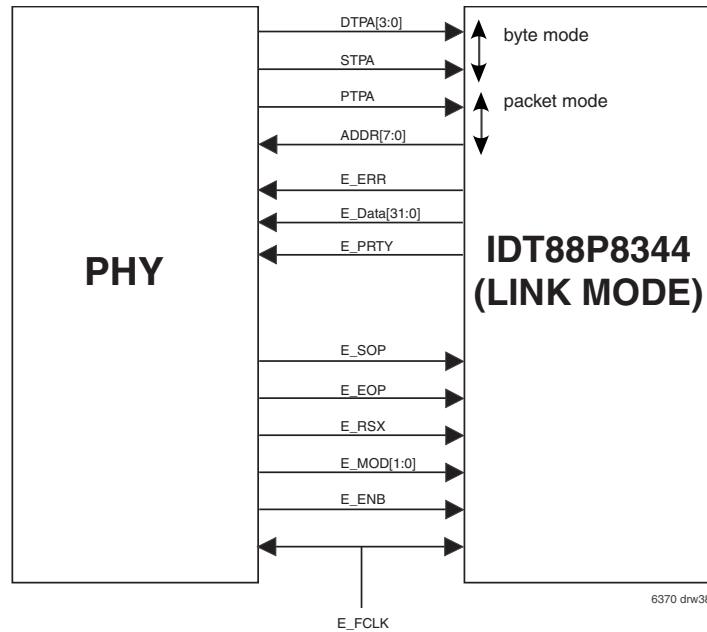


Figure 5. Link mode SPI-3 egress interface

**SPI-3 egress PHY mode**

In PHY mode, the SPI Exchange sends data to the attached Link-mode device as long as the E\_ENB signal is asserted. The SPI-3 packet fragment processor transfers data to the SPI-3 egress buffers.

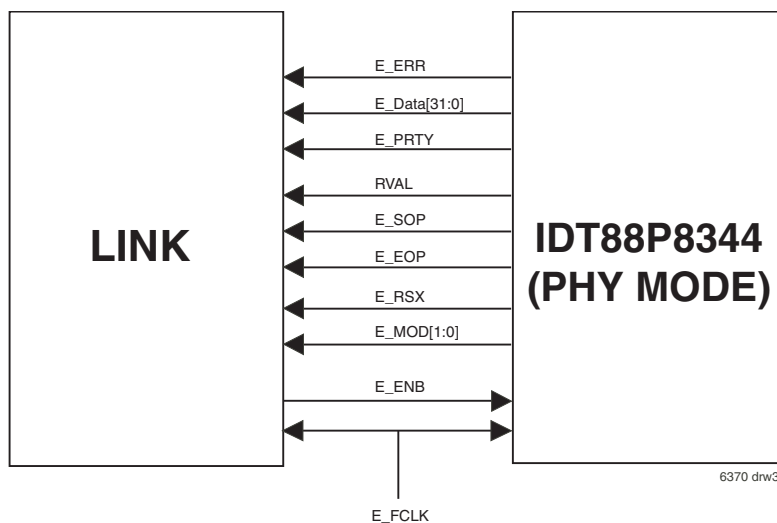


Figure 6. PHY mode SPI-3 egress interface



## 3.2 SPI-4

Refer to OIF SPI-4 document (see Glossary) for full details of the implementation agreement.

- Clock rate is 80 - 400 MHz (160 - 800MHz DDR)
  - Link and PHY modes supported
  - Address range 0 to 255 with support for 256 simultaneously active logical ports
  - MAXBURST parameters configurable 16-256 bytes in 16 byte multiples
  - 256 entry calendar
  - LVTTTL and LVDS status signals supported
- The following are implemented for the SPI-4 interface:
- Link / PHY layer device
  - Physical port active

### 3.2.1 SPI-4 ingress

The SPI-4 ingress includes

- Bit alignment
- Word alignment/ de-skew
- Transfer decode and dispatch
- PFP interface
- Status frame generation

## SPI-4 ingress configurable parameters

### SPI-4 LID map

- 256 entries, one per SPI-4 LP
- SPI-3 physical interface identifier
- Physical port enable

### Word / bit synchronization

- LVDS clock data alignment and LVDS data de-skew

### Minimum packet length

- Packets shorter than the minimum length will be optionally counted in the short packet counter.
- Range 0 – 255 in 1 byte increments

### Maximum packet length

- Packets longer than the maximum length will be optionally counted in the long packet counter.
- Range 0 – 16,383 in 1 byte increments

### Free segment backpressure threshold

- Number of free buffer segments allocated to trigger backpressure for the LP

### Data sampling

The I\_LOW field in the Table 89 SPI-4 ingress configuration register (Block\_base 0x0300 + Register\_offset 0x00) selects an operating mode between 80 MHz and 200 MHz or between 200 MHz and 400 MHz.

Each lane is over-sampled by a factor of five. The over-sampled data is generated by a locked tapped delay line and clocked in to a register at the clock

rate. The current samples  $c(n)$  and the previously generated samples provide samples for the eye computation. The optimized sampling point will be selected based on the eye computation. The tap selector is updated if necessary at the end of the eye pattern measurement interval. The tap selector moves no more than one tap at a time as a result of the eye pattern measurement.

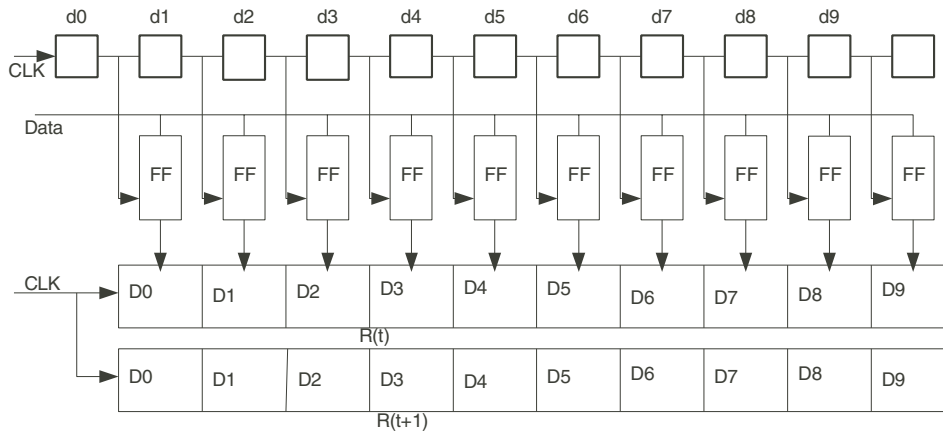


Figure 7. Data sampling diagram

### Eye measurement

$$C[0] = R_t.D2 \wedge R_t.D3$$

$$C[1] = R_t.D3 \wedge R_t.D4$$

.....

$$C[7] = R_t.D9 \wedge R_{t+1}.D0$$

$$C[8] = R_{t+1}.D0 \wedge R_{t+1}.D1$$

$$C[9] = R_{t+1}.D1 \wedge R_{t+1}.D2$$

Accumulation results during a window defined by  $W$  are stored in the diagnostics table.

The latest result can be read out for diagnostic purposes.

### Output tap selection

The sampling tap is automatically selected based on the eye measurement.

### Manual phase selection

The automatic phase adjustment can be overruled by the processor when the FORCE flag is set see Table 99, SPI-4 ingress bit alignment control register (register\_offset 0x11). The PHASE\_ASSIGN field see Table 113, SPI-4 ingress manual alignment phase/result register (0x0C to 0x1F) now defines the selected phase.

### Word alignment

The de-skew block searches for the Training Control Word 0x0FFF. If the Training Control Word is found, then training data is expected to follow the Training Control Word. The orthogonal training data will be used to align the word.

A de-skew control bit (I\_DSC in Table 89-SPI-4 ingress configuration register at Block\_base 0x0300 + Register\_offset 0x00) is used to protect against a random data error during de-skew. If I\_DSC=1, then two consecutive de-skew results are required. It is recommended to set I\_DSC to 1.

For diagnostics, an out of range offset between lines is provided. If the offset is more than two bits between the earliest and latest samples, I\_DSK\_OOR is set to a logic one. I\_DSK\_OOR is cleared to a logic zero when the offset is in range.

### Transfer decode and dispatch

In the OUT\_OF\_SYNC state, the de-skew block will decode the transfer, and check the DIP-4 for validation.

A number of consecutive error free DIP-4 ingress bursts will lead to a transition to the IN\_SYNC. The number is defined by the I\_INSYNC\_THR field in Table 89-SPI-4 ingress configuration register (Block\_base 0x0300 + Register\_offset 0x00).

In the IN\_SYNC state, the PFP decodes the status transfer, check the DIP-4, and dispatches the data.

A number of consecutive DIP-4 errors will lead to the OUT\_OF\_SYNC state. The number is defined by the I\_OUTSYNC\_THR field in Table 89-SPI-4 ingress configuration register (Block\_base 0x0300 + Register\_offset 0x00).

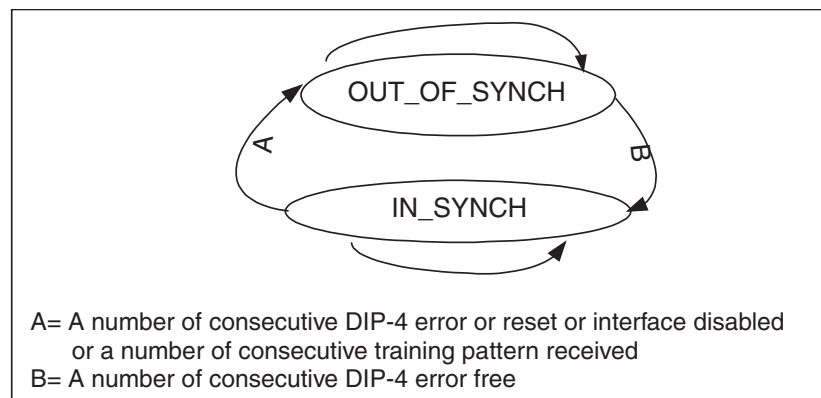
A number of consecutive training patterns will lead to OUT\_OF\_SYNC. The number is defined by the STRT\_TRAIN field in the Table 100 SPI-4 ingress start up training threshold register (Block\_base 0x0300 + Register\_offset 0x12). This feature is disabled if STRT\_TRAIN=0.

### Control word and data

A control word is distinguished by the SPI-4 RTCL signal. (logic one = control word).

### DIP-4 check

For the DIP-4 check algorithm refer to the OIF SPI-4 document [Glossary]. In both IN\_SYNC and OUT\_OF\_SYNC states, only control word previous and following data is checked. Any transition on synch status will be captured. In IN\_SYNC state, each DIP-4 error is captured and counted.



6370 drw35

Figure 8. SPI-4 ingress state diagram

### Transfer decode

The SPI-4 ingress control word contains various fields. Refer to the OIF SPI-4 document [Glossary] for details. If reserved control word, BIT[15:12]=0011, 0001, 0101, or 0111 is detected, a BUS\_ERROR event is generated. If a payload control word is not followed by a data word, or a data word does not follow a payload control word, a BUS\_ERROR event is generated. If abort is detected, the next packet will be tagged with an error.

### Data dispatch

The port address field of a payload control word is extracted as a search key. The search key is used to search the dispatch info in Table 86, SPI-4 ingress LP to LID map (256 entries, one per LP). If the searched port is active, transfer data is sent to the associated PFP with SOP, EOP, LENGTH, PACKET\_ERROR. If the searched port is inactive, a SPI4\_INACTIVE\_TRANSFER event is

generated. A SPI-4 inactive transfer event with its associated LP will be captured in the Table 40, SPI-4 status register (0x22 in the direct accessed space).

### SPI-4 ingress status channel Calendar structure and swapping

The SPI Exchange supports one or two sets of calendars. If I\_CSW\_EN field in the Table 89, SPI-4 ingress configuration register (0x00)=1, two sets of calendars are supported. A calendar selection word must be placed following the framing bit. Refer to the OIF SPI-4 document [see Glossary] for more details.

### SPI-4 ingress status channel frame generation

The status frame can be one of the following cases:

- All '11' when LVTTL is in the out of synch state
- Training pattern when LVDS is in the out of synch state or in periodic training

- Normal status information when in the IN\_SYNCH state

The normal status information is generated based on ingress buffer full information and PFP buffer segment fill level.

For information on DIP-2 generation and training pattern refer to the OIF SPI-4 document [Glossary].

**DIP-2 error insertion**

A number of consecutive DIP-2 errors can be generated. The I\_DIP\_E\_NUM field in Table 97, SPI-4 ingress diagnostics register (register\_offset 0x0F) specifies the number of errors to be generated. A logic one written to I\_ERROR\_INS will activate the I\_DIP\_E\_NUM field and trigger error insertion. The I\_ERROR\_INS field self clears when the number of errors have been generated.

**LVTTTL and LVDS status interface selection**

The LVDS\_STA pin selects which FIFO status interface is being used for SPI-4. HIGH = LVDS status interface, LOW = LVTTTL status interface.

**3.2.2 SPI-4 egress**

The SPI-4 egress includes

- Status channel synchronization
- Status updating
- Data transfer
- Periodic training
- PFP interface

**SPI-4 egress configurable parameters**

All parameters as listed in the OIF SPI-4 document [see Glossary]

CALENDAR\_LEN: 4 to 1,024 in increments of 4

CALENDAR\_M: 1 to 256 in increments of 1

MaxBurst1 (MaxBurst\_S): 16 to 256 in increments of 16

MaxBurst2 (MaxBurst\_H): 16 to 256 in increments of 16

Alpha: 1 to 256 in increments of 1

DATA\_MAX\_T: 1 to 4,294,967,040 in increments of 1

FIFO\_MAX\_T: 1 to 16,777,215 in increments of 1

**Calendar and shadow calendar**

- 256 entries

- E\_CSW\_EN field in Table 104, SPI-4 egress configuration register\_0 (register\_offset 0x00) bit for manual reconfiguration swap

**Multiple burst enable**

- Allows more than one burst to be sent to an LP. Feature included to relieve systems with long latency between updates.

**SPI-4 egress LID to LP map**

- 256 entries, one per SPI-4 LP

- Enable bit

**3.2.3 SPI-4 startup handshake**

**TABLE 10 – BOTH ATTACHED DEVICES START FROM RESET STATUS**

Ingress	egress
Out of synch, send status training	Out of synch, send data training
In synch, send status frame	Out of synch, send data training
In synch, send status frame	In synch, send data/idle

**TABLE 11 – INGRESS OUT OF SYNCH, EGRESS IN SYNCH**

Ingress	Egress
Out of synch, send status training	In synch, send data/idle
Out of synch, send status training	Out of synch, send data training
In synch, send status frame	Out of synch, send data training
In synch, send status frame	In synch, send data/idle

**TABLE 12 – INGRESS IN SYNCH, EGRESS OUT OF SYNCH**

Ingress	Egress
In synch, send status frame	Out of synch, send data training
Out of synch, send status training	Out of synch, send data training
In synch, send status frame	Out of synch, send data training
In synch, send status frame	In synch, send data/idle

## SPI-4 egress status channel

### Status channel bit alignment

The bit alignment algorithm for the status channel is the same as was described for the data channel.

### Status Channel Frame synchronization

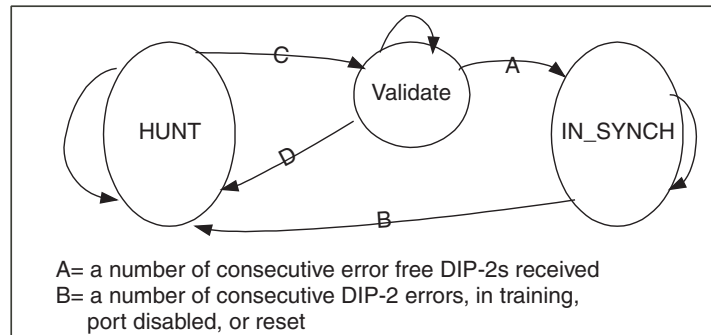


Figure 9. SPI-4 egress status state diagram

6370 dnv30

The status channel frame module has 3 states: HUNT, VALIDATE and IN\_SYNC.

In the HUNT state, the status channel frame module searches for status frame, status clear and status freeze.

In the VALIDATE state, the status channel frame module checks DIP-2.

In the IN\_SYNC state, the status channel frame module checks DIP-2, and updates status.

#### HUNT state

- In the HUNT state, per Link status is fixed to 'satisfied'.
- In HUNT state, the PFP searches frame continuously. It transitions to the VALIDATE state if a single valid frame is found accompanied by a single valid training pattern. A frame is considered to be found if: 1) only one frame word is at the beginning of a frame, 2) the calendar selection word, if enabled, is matched, and 3) the DIP-2 calculation matched the received DIP-2.

#### VALIDATE state

In the validate state, based on the frame found while in the HUNT state, the DIP-2 is checked.

If a single DIP-2 error is found, transition to the HUNT state.

After a number of consecutive DIP-2 calculations proves to be error free, transition to the IN\_SYNC state. The number is defined by the E\_INSYNC\_THR field in Table 104-SPI-4 egress configuration register\_0 (Block\_base 0x0700 + Register\_offset 0x00).

In the validate state, the training pattern is not checked.

#### IN\_SYNC state

In the IN\_SYNC state, training frame and status frame are checked.

DIP-2 is checked for status frame. Each mismatched DIP-2 will generate a DIP-2 error event, each event will be captured and counted.

After a number of consecutive DIP-2 errors, transition to the HUNT state. (Clear status in HUNT mode). The number is defined by the E\_OUTSYNC\_THR field in Table 104-SPI-4 egress configuration register\_0 (Block\_base 0x0700 + Register\_offset 0x00).

The reception of twelve consecutive training patterns forces a transition to HUNT mode. If less than twelve consecutive training patterns are received, synch will not be lost, and status frame starts at the end of training.

Twelve consecutive '11' patterns force a transition to the HUNT state.

Status updating occurs without waiting for the end of a status frame.

### Status channel de-skew

The LVDS status channel deskew uses the same algorithm as the data channel.

### LVTTTL or LVDS status channel option

The LVDS\_STA pin selects the interface type. A logic high enables the LVDS status interface. A logic low enables the LVTTTL status interface.

### Data channel

#### Data transfer and training

At any cycle, the contents on the interface can be one of the following:

- Control word: Payload control word, or idle control word or training control word.
- Data word: Payload data word or training data word.

In the HUNT or the VALIDATE state, the training pattern is sent.

In the IN\_SYNC state, data from is taken from the buffer segments and egressed to the SPI-4 interface. The switch between data burst, IDLE, and training must obey the following rules:

- Send IDLE if no data to transmit
- SOP must not occur less than 8 cycles apart.
- periodic training after current transfer finished

Payload control word generation:

- Bit 15, Control word type=1
- Bit [14:13] EOPS per [see Glossary: SPI-4]. If an error tag is in the descriptor, abort.
- Bit [12] SOP refer to [see Glossary: SPI-4]
- Eight Bit Address. Mapping table defined in Table 101, SPI-4 egress LID to LP map (256 entries)
- DIP-4 bit refer to [see Glossary]

Payload data word

- Bit order refer to [see Glossary: SPI-4]
- If only one byte is valid, 8 LSB (B7 to B0) is set to 0x00.

### No status channel option

Once the NOSTAT bit is set, the status channel is ignored. Refer to Table 104, SPI-4 egress configuration register\_0 (register\_offset 0x00).

Status in default value.

No DIP error check.

No status updating, the received status fixed to STARVING.

Data channel works same as in IN\_SYNC state.

### 3.3 Microprocessor interface

- Parallel microprocessor interface
  - 8 bit data bus for parallel operation
  - Byte access
  - Direct accessed space
  - Indirect access space is used for most registers
  - Read operations to a reserved address or reserved bit fields return 0
  - Write operations to reserved addresses or bit fields are ignored
- Serial microprocessor interface
  - Compliance to Motorola serial processor interface (SPI) specification
  - Byte access
  - Direct accessed space
  - Indirect access space is used for most registers
  - Read operations to a reserved address or reserved bit fields return 0
  - Write operations to reserved addresses or bit fields are ignored

#### General purpose I/O

Five general purpose I/O pins are provided. The direction is independently controlled by the DIR\_OUT field in the GPIO register (Table 123 GPIO Register

(0x20)). The logical level on a pin is controlled by the LEVEL field in the GPIO register if DIR\_OUT=1, or sensed if DIR\_OUT=0. The LEVEL bit monitors the logic level of any bit selected from the indirect access space if MONITOR\_EN is set high. A bit in the indirect access space can be selected for monitoring by the by the ADDRESS and BIT fields in the GPIO Link table (Table 124, *GPIO Monitor Table (5 entries 0x21-0x25 for GPIO[0] through GPIO[4])*).

All GPIO pins must be programmed into or out of monitor mode at the same time.

#### Interrupt scheme

Events are captured in interrupt status registers. Interrupt status flags are cleared by an microprocessor write cycle. A logical one must be written to clear the flag(s) targeted. A two level interrupt scheme is provided comprising a primary level and a secondary level.

The primary level identifies the secondary interrupts sources with a pending interrupt. This information is reflected in the primary interrupt register. Interrupt status can be enabled by associated flags both in the primary and secondary level of the interrupt scheme.

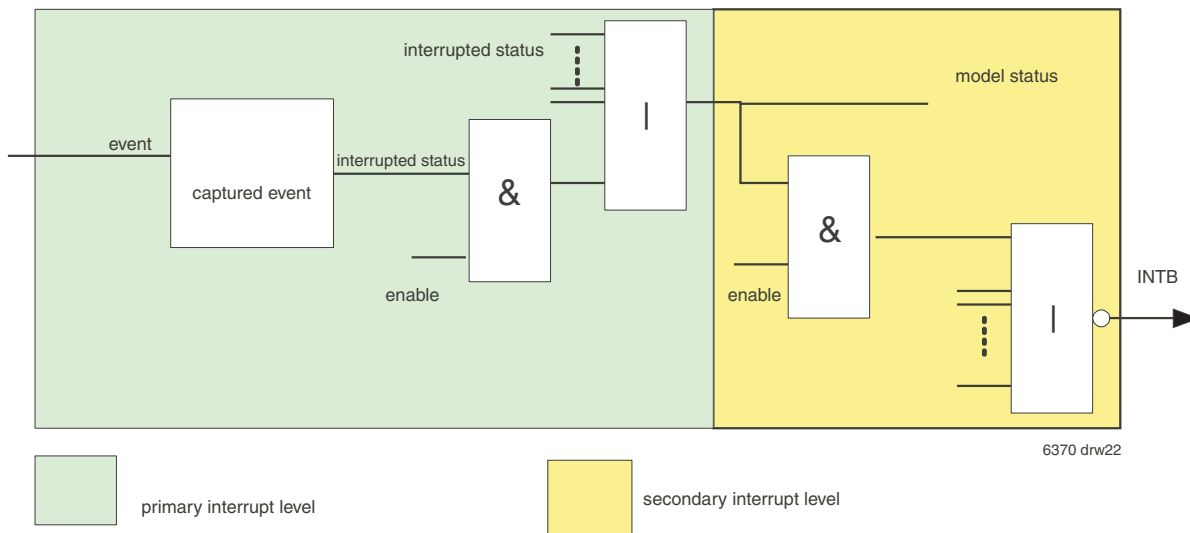


Figure 10. Interrupt scheme

### 4. DATAPATH AND FLOW CONTROL

The following sections describe the datapaths through the device. The datapaths shown are as follows:

- SPI-3A <-> SPI-4
  - SPI-3B <-> SPI-4
  - SPI-3C <-> SPI-4
  - SPI-3D <-> SPI-4
  - SPI-3A <-> SPI-3B
  - SPI-3C <-> SPI-3D
  - SPI-3A <-> microprocessor interface
  - SPI-3B <-> microprocessor interface
  - SPI-3C <-> microprocessor interface
  - SPI-3D <-> microprocessor interface
  - SPI-4 <-> microprocessor interface
- Where <-> indicates a bidirectional datapath.

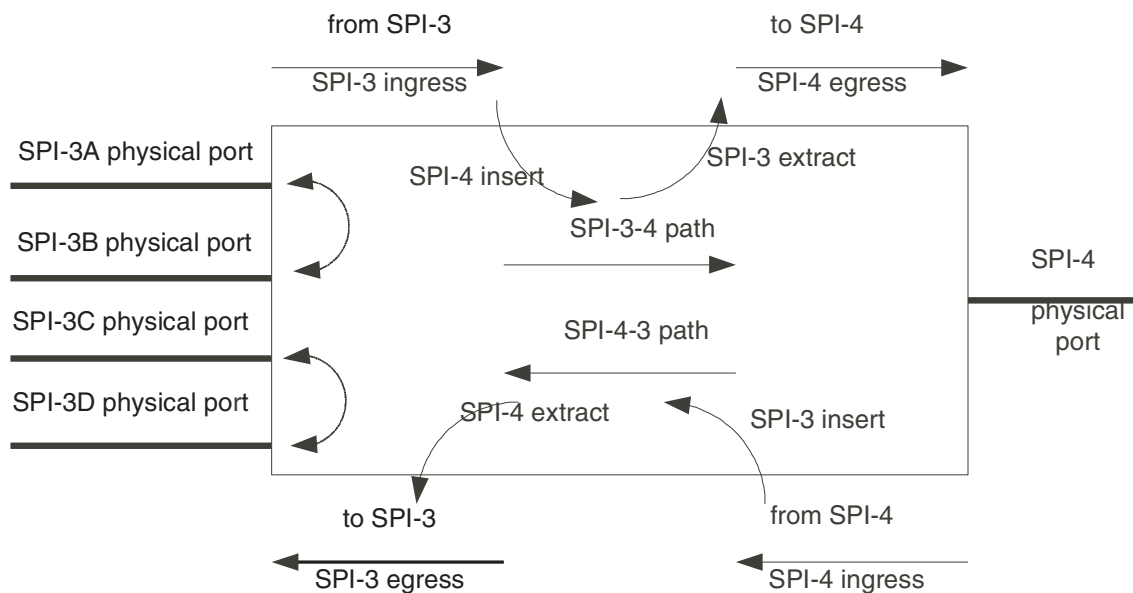
The IDT88P8344 supports four SPI-3 interfaces and a single SPI-4 interface. All SPI-3 interfaces can operate independently in a PHY or Link mode. Refer

to Figure 11, *Definition of Data Flows* for the main data flows in the device. Independent logical data flows are transported over each of the physical ports. Those logical flows are identified by logical port addresses on the physical port and by a Link identification (LID) map in the core of the IDT88P8344.

### DATA BUFFER ALLOCATION

Flexibility has been provided to the user for data buffer allocation. The device has 128 KByte of on chip memory per SPI-3 port per direction—a total of 1MByte of on-chip data memory.

The 128 KByte SPI-3 buffers (8 instantiations per device) are divided into 256 byte segments. The segments are controlled by a packet fragment processor. The user configures the maximum number of segments per LP to allocate to a port and the number of segments allocated from the buffer segment pool that will trigger the flow control mechanism. There is no limitation on the reallocation of freed segments among logical ports, as would be present if the memory had been allocated by a simple address mechanism.



6370 drwXA

Figure 11. Definition of data flows

**DATAPATH CONFIGURATION**

A logical view of datapath configuration using Packet Fragment Processors is shown in Figure 12, *Logical View of Datapath Configuration Using PFPs*. Two PFPs are associated with each SPI-3 port, one for ingress and one for egress. Two PFPs are associated with each SPI-4 port, one for ingress and one for egress. Logical ports are mapped internally into Logical Identifiers (“LIDs”, “LID Map”) for the control of each per-LID data flow to each physical port, logical port, memory queue size, and backpressure threshold (watermark), by programming the LID register files.

egress. Logical ports are mapped internally into Logical Identifiers (“LIDs”, “LID Map”) for the control of each per-LID data flow to each physical port, logical port, memory queue size, and backpressure threshold (watermark), by programming the LID register files.

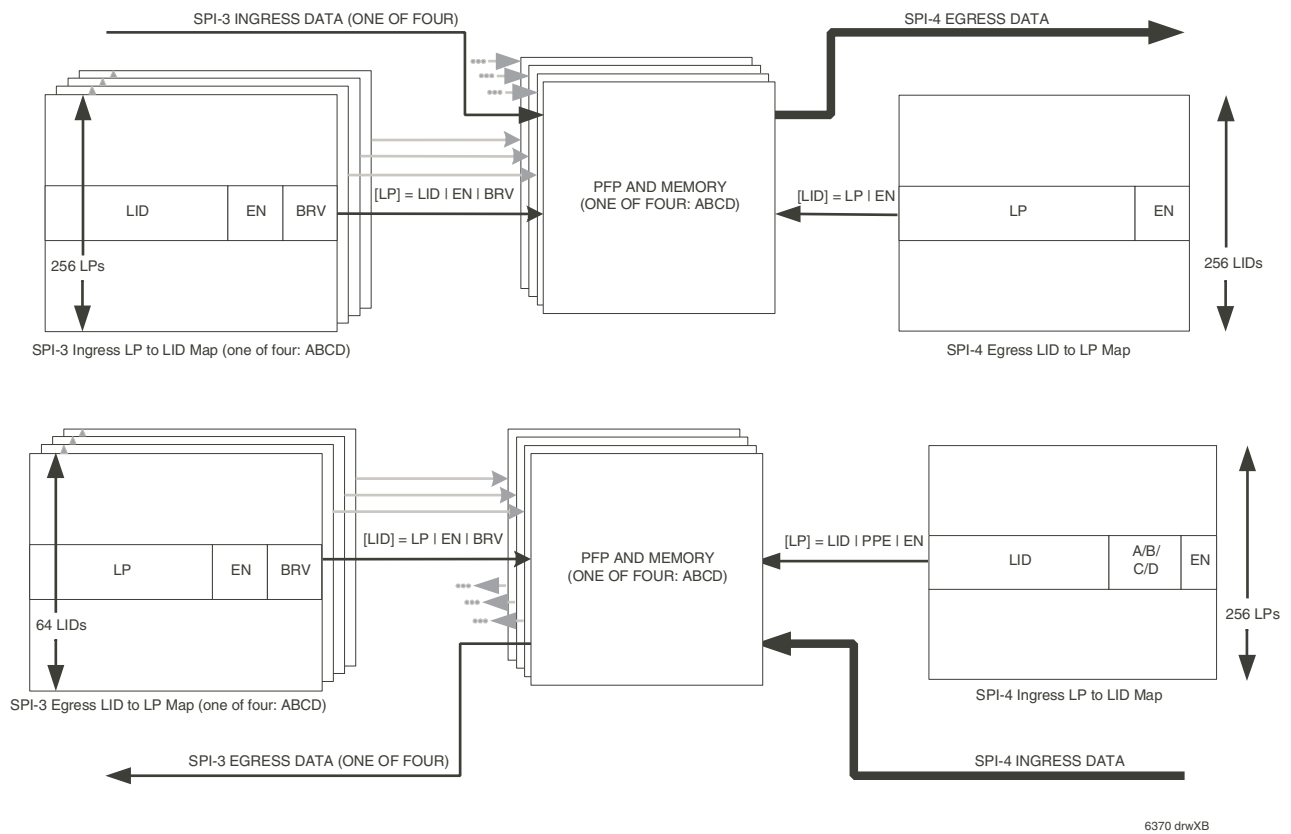


Figure 12. Logical view of datapath configuration using PFPs

- LID - Logical Identifier
- EN - LID enable flag
- BRV - Bit reversal
- LP - Logical port
- PFP - Packet fragment processor



## 4.1 SPI-3 to SPI-4 datapath and flow control

Four packet fragment processor modules from SPI-3 to SPI-4 are provided. One packet fragment processor module is associated with one SPI-3 ingress interface. All four packet fragment processor modules connect to a single SPI-4 interface.

Packet fragments from the SPI-3 ingress are received into the SPI-3 ingress port buffers. A packet fragment processor transfers complete packet fragments from the SPI-3 ingress port buffers to memory segments previously reserved on a per-LP basis in the buffer segment pool. The SPI-3 ingress port buffer watermark and the per-LP free buffer segment threshold information is combined

to produce SPI-3 ingress FIFO status towards the attached device. Packets or packet fragments received on one SPI-3 ingress logical port can be forwarded to any one of:

A logical port on the egress SPI-4 interface.

A logical port on an associated SPI-3 interface (between physical port interfaces A and B, and between C and D only).

The microprocessor interface, using the capture buffer.

The connection on the logical port level is performed through an intermediate mapping to a Link Identification number (LID).

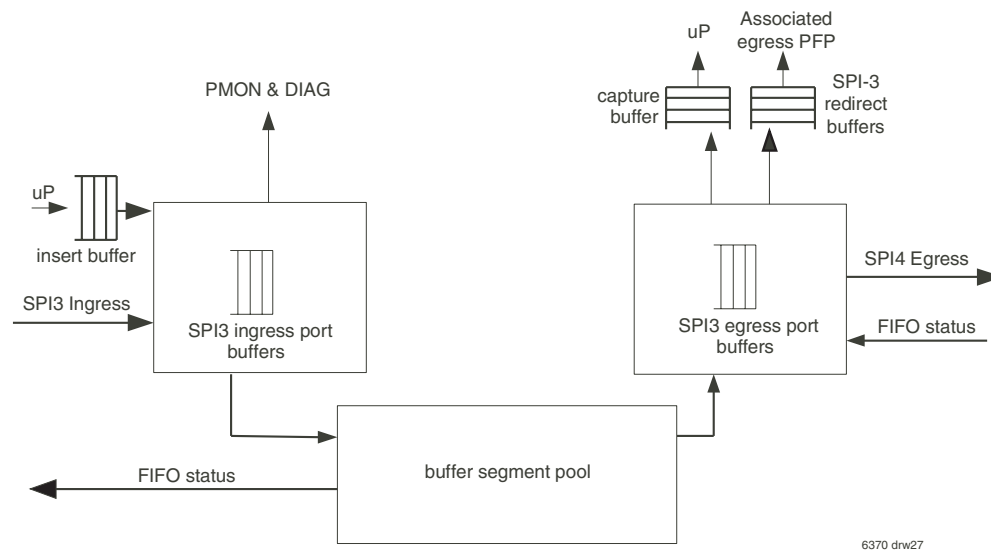


Figure 13. SPI-3 ingress to SPI-4 egress packet fragment processor

### SPI-3 ingress PFP functions

The packet fragment processor (PFP) receives status information about the SPI-3 ingress buffers and the microprocessor insert buffer. The PFP processes SPI-3 ingress buffers in high priority and the insert buffer with low priority. The PFP copies data into the buffer segment, requests new buffer segments, and generates entries in the SPI4-egress queue.

### SPI-3 ingress buffer processing

The PFP verifies whether a SPI-3 ingress buffer is occupied. If the SPI-3 ingress buffer is not occupied the PFP processes the insert buffer.

### Normal operation

In loopback mode, all of the SPI-3 ingress buffers of a physical SPI-3 port are copied into the SPI-3 egress buffers of that same port. This is a test mode only, as no non-loopback traffic can be transferred at this time.

In non-loopback mode (normal operation) the SPI-3 ingress buffers are forwarded to the LID process by the PFP.

The LID process generates a set of events for an associated LID. The events that are directed towards the PMON&DIAG module are:

- SPI-3 error tagged packet event (errored packets)
- SPI-3 EOP event (all packets)
- SPI-3 fragment event (all fragments) with an associated length field

**Erroneous operation**

SPI-3 ingress buffers marked with an address parity error are always immediately flushed. A *SPI-3 flush event* is generated.

**Store process**

The process parameters are stored in a descriptor table. One entry in the table is required for each of the SPI-3 logical ports. Refer to Table 49, *SPI-3 ingress LP to LID Map*.

- Fragments tagged with an SOP indication trigger the buffer segment request process. The internal Packet\_length variable is initialized. The copy process is triggered.
- Fragments tagged with an EOP indication will trigger the packet length check process and the queue process.
- Non marked (EOP or SOP) buffers are subject to the copy process.

**Buffer segment request process**

A new buffer segment is requested for the logical port from the buffer segment pool.

The request can be accepted or rejected by the buffer pool.

When accepted, the Current\_Seg and current Seg\_Length variables are updated.

When rejected, the SPI-3 ingress buffer is flushed. An *SPI-3 flush event* is generated and directed towards the Table 61, Non LID associated event counters (0x00 - 0x0B). The buffer data is not copied into the SPI-4 buffer.

**Copy process**

Data is retrieved from the buffer and stored in the current segment. The data parity error status is stored in the Pack\_Err variable. The Packet\_Length variables and Seg\_Length variables are updated. The queue and request processes are triggered when the number of bytes in the buffer segment equals the SPI-3 packet fragment size programmed for that physical interface, or an EOP is reached.

**Queue process**

The current segment is entered into the SPI-4 egress queue.

**Packet length check**

The length of the packet is compared to the MIN\_LENGTH and MAX\_LENGTH parameters in the ingress SPI-3 Port descriptor table. If the packet length is less than the programmed field MIN\_LENGTH a "*SPI-3 too short packet event*" is generated. If the packet length is greater than the programmed field MAX\_LENGTH a "*SPI-3 too long packet event*" is generated. The events are directed towards the Table 61, Non LID associated event counters (0x00 - 0x0B).

**SPI-3 to SPI-4 buffer management**

A 128 KB SPI-3 to SPI-4 buffer segment pool is assigned to each physical SPI-3 ingress port. A configurable part of this buffer segment pool is assigned to buffers associated to each of the up to 64 LIDs. The buffer size for a LID can be configured in multiples (M) of 256 bytes. Fewer LIDs allow larger buffers per LID, conversely a large number of LIDs will require smaller buffers per LID. Within this restriction, the buffer size of each LID can be further restricted as

needed to control latency. Modifications of the buffer size allocated to a LID are supported only when the logical port associated to the LID is disabled. Attempts to allocate more memory than available will generate an allocation error event. The indirect access module will discard the attempt.

**Free buffer segment pool****Storage**

The buffer segment pool is divided into 508 segments. The device holds a pool of free buffer segments. The buffer segment pool keeps track of the number of segments assigned to each LID and holds a list of free segments.

**Buffer segment requests**

A new segment for a logical Link (LID) can be requested from the buffer segment pool for that SPI-3 ingress physical port by the SPI-3 ingress packet fragment processor associated to that SPI-3 physical port. A request may be accepted immediately or rejected. When the request is accepted a buffer segment ID is returned immediately.

**Buffer segment pool returned segments**

A buffer segment can be returned to the buffer segment pool when the egress module releases it. This allows the segment to be used once more by the SPI-3 ingress.

**SPI-4 egress queues****Normal operation**

508 SPI-4 egress queue entries are provided. They are evenly allocated to the number of logical ports as defined by the static NR\_LID configuration. One entry in the queue corresponds to a packet or a packet fragment to be forwarded to the SPI-4 egress interface.

**SPI-3 ingress Backpressure**

The module directs status signals for each of the 64 LIDs associated with a SPI-3 physical interface towards the SPI-3 ingress interface. The status signals request to transfer more data on the logical port associated to the LID. The available status is defined by the function (if free segments[LID] > Threshold, status=available).

**SPI-4 egress direction control**

The SPI-4 egress traffic can be captured by the microprocessor, directed to an associated SPI-3 egress port (SPI-3 port A to B, or port C to D, only), to the SPI-4 egress port, or discarded. The selection is defined for each of the 64 LIDs by the associated DIRECTION field in the Table 13, *Direction code assignment*.

**TABLE 13 - DIRECTION CODE ASSIGNMENT**

DIRECTION	Path
00	SPI-4
01	Associated SPI-3
10	Capture to microprocessor
11	Discard

**SPI-4 egress data bursts**

The PFP produces fragments of up to N\*16 bytes. N is defined by the MAX\_BURST\_H or MAX\_BURST\_S parameter associated with each LID. For a high priority (starving) LID the MAX\_BURST\_S parameter is used. For a low priority (hungry) LID the MAX\_BURST\_H parameter is used. The PFP may not fill the buffers to the level granted when a new segment needs to be used in the SPI3-4 buffer memory or when the last fragment of a packet is copied into the buffer. The information received over the FIFO status channel is interpreted as status or credit information as selected by the CREDIT\_EN flag in Table 78, SPI-3 to SPI-4 flow control register (0x01). If the status mode is used, data will be egressed until the status is changed. If the credit mode is used, the SPI-4 egress will issue only one credit's worth data burst and then wait for another credit from the status channel before issuing another LID burst.

**SPI-4 egress FIFO status channel updates**

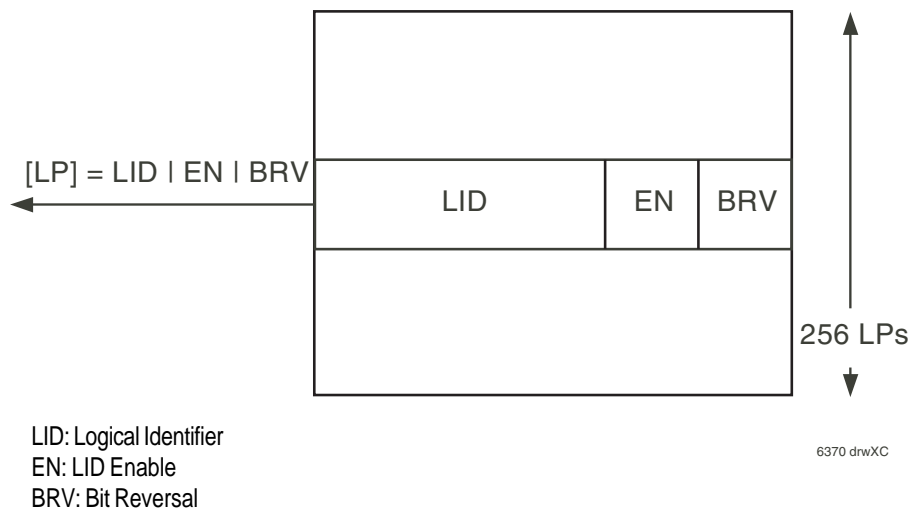
The SPI-4 egress FIFO Status Channel Module continuously verifies the status information for the LIDs associated to SPI-4 logical ports. The PFP searches and selects a LID, fetches the associated information and queues data to the SPI-4 egress. The obsolete buffer segment is returned to the free buffer segment pool (unless the repeat test feature is enabled). Searching the LID to be served is performed for both a high priority and a low priority LID. The priority is defined by the status received from the SPI-4 egress module.

**SPI-3 ingress logical port mapping**

Each of the four SPI-3 interfaces has an associated SPI-3 ingress LP to LID map, (See Table 49) for the purpose of directing the packet fragments from its SPI-3 ingress to its associated SPI-3 ingress main memory buffer segment pool. The SPI-3 LID map has 256 entries, one per SPI-3 LP, but only 64 LPs are supported on any SPI-3 interface at any one time. Each SPI-3 interface has an enable bit, as well as the ability to reverse the bit ordering of the interface. The packet fragment length is associated with a SPI-3 interface. The allowed range is 0 to 255 bytes per packet fragment. The last fragment of a packet can be shorter than the programmed fragment size. Each SPI-3 port can be independently set for either Link or PHY mode of operation.

**SPI-3 ingress LID associated control**

Each LID on a SPI-3 interface has the ability to be programmed for minimum and maximum packet length. The minimum packet length can be set from 0 to 255 bytes in one byte increments. The maximum packet length can be set from 0 to 16,383 bytes in one byte increments. Each LID can be enabled and disabled independently.



**Figure 14. SPI-3 ingress LP to LID map**

**SPI-4 egress interface port associated control**

The SPI-4 interface has an associated LID to LP map (See Table 101 - SPI-4 egress LID to LP Map Block\_base 0x0400 = Register\_offset 0x00 - 0xFF) for the purpose of directing the packet fragments from the selected SPI-3 ingress main memory buffer segment pool to the SPI-4 egress interface. The SPI-4 LID map has 256 entries, one per LID. The SPI-4 interface has an enable bit. The burst length is associated with the SPI-4 interface. The allowed burst range is 16 to 256 bytes per burst. The last burst of a packet can be shorter than the programmed burst size.

**SPI-4 egress LID associated control**

Each of the 256 entries in the SPI-4 egress LID to LP map (See Table 101 - SPI-4 egress LID to LP Map (256 entries)) is used to control the pulling of bursts out of the buffer segment pool and into the SPI-4 egress interface. Each LID can be enabled and disabled independently.

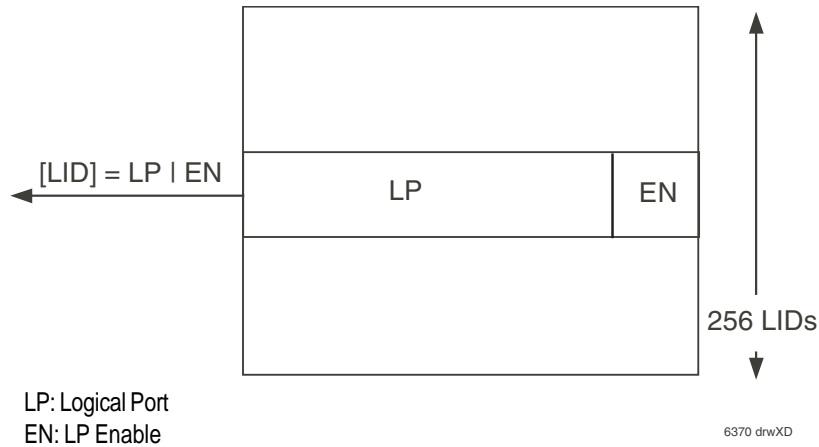


Figure 15. SPI-4 egress LID to LP map

The diagram below shows the datapath through the device from a SPI-3 ingress interface to the SPI-4 egress interface.

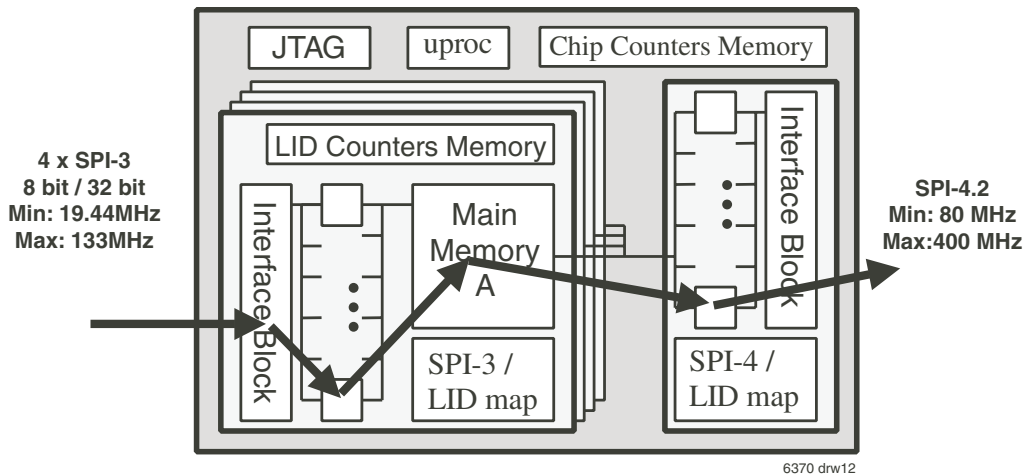


Figure 16. SPI-3 ingress to SPI-4 egress datapath

**SPI-3 ingress to SPI-4 egress flow control**

For control information there are two separate cases to consider: The case that the SPI-3 physical interface port is configured in Link mode, and the case that the SPI-3 is configured in PHY mode. Note that since the SPI-3 physical interfaces are configured separately, the device is able to deal with the case that some of the LP fragments have been received on a Link layer device SPI-3 interface and some have been received on a PHY layer device SPI-3 interface.

For a device in Link mode the Link device can only control the flow of data through the RENB signal. Two modes of operation are implemented and configurable for flow control on this interface – either the data can be allowed to flow freely into the device or the RENB signal will be asserted if a condition arises that one of the LPs is unable to receive another fragment. The first of these modes is considered to have no Link layer device flow control, and the second has Link layer device flow control.

For the no Link flow control mode, any data sent to an LP unable to receive another fragment will cause an LP overflow.

For a device in Link mode the Link has complete knowledge of the fill level of the data buffers in each of the LPs in the PHY. This knowledge is attained either through byte level polling or packet level polling.

Both in Link and PHY modes, the data is collected to buffer segments associated with an LP. The SPI-4 PFP is updated with the number of free segments available to the LP. The SPI-4 PFP determines which LP to service based on two factors: whether the LP contains enough data for a burst, and the starving / hungry / satisfied state of the LP. For details on the mapping of LPs to LIDs, refer to Table 101 - SPI-4 egress LID to LP Map Block\_base 0x0400 = Register\_offset 0x00 - 0xFF.

**SPI-3 ingress flow control registers**

The following are implemented per SPI-3 interface, and there are four instantiations per device.

**Backpressure enable**

Link mode only  
Enables the assertion of the I\_ENB pin when at least one active LID can not accept data

If not enabled, the I\_ENB signal will never be asserted in Link mode, possibly leading to fragments being discarded.

**SPI-4 egress flow control configurable parameters**

All parameters as listed in SPI-4 implementation agreement:

- CALENDAR\_LEN: 4 to 1,024 in increments of 4
- CALENDAR\_M: 1 to 256 in increments of 1
- MaxBurst1 (MaxBurst\_S): 16 to 256 in increments of 16
- MaxBurst2 (MaxBurst\_H): 16 to 256 in increments of 16
- Alpha: 1 to 256 in increments of 1
- DATA\_MAX\_T: 1 to 4,294,967,040 in increments of 1
- FIFO\_MAX\_T: 1 to 16,777,215 in increments of 1

**SPI-4 egress flow control calendar and shadow calendar**  
256 entries

**SPI-4 egress flow control multiple burst enable**

Allows more than one burst to be sent to an LP. This feature was included to increase throughput in systems with long latency between updates.

The diagram below shows the SPI-3 ingress to SPI-4 egress flow control Path through the IDT88P8344 device.

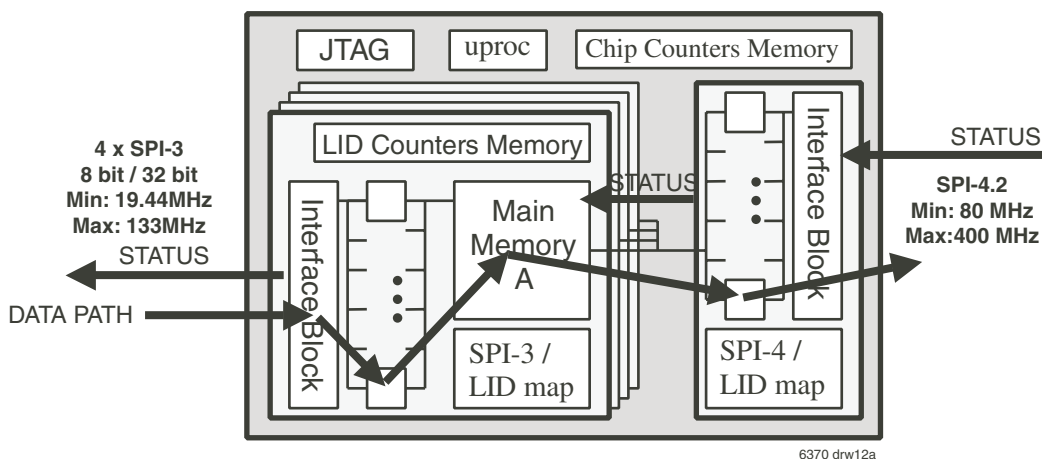


Figure 17. SPI-3 ingress to SPI-4 egress flow control path

## 4.2 SPI-4 to SPI-3 datapath and flow control

Four Packet Fragment Processor modules from SPI-4 ingress to SPI-3 egress are provided, all connected to one SPI-4 ingress interface.

Packet bursts from the SPI-4 ingress are received into the SPI-4 ingress port buffers. A packet fragment processor transfers complete packet bursts from the SPI-4 ingress port buffers to memory segments previously reserved on a per-LP basis in the buffer segment pool. The SPI-4 ingress port buffer watermarks and the per-LP free buffer segment threshold information is combined to produce SPI-4 ingress FIFO status (per-LP starving, hungry, or satisfied) towards the attached SPI-4 device. Per-LP buffer segment threshold information is used to produce FIFO status information for the attached SPI-3 device. Packets or packet fragments are forwarded to the SPI-3 interface when a packet is complete or a predefined number of bytes have been received. Packets or packet fragments received on one SPI-4 logical port are cross connected to an SPI-3 logical port through an intermediate mapping to a Link Identification, or LID. Its mode of operation is similar to the SPI-3 ingress to SPI-4 egress packet fragment processor, with the following differences:

- 1) The PFP4-3 data input has three sources, listed in descending priority: SPI-4 buffers, redirect buffers, and insert buffers.
- 2) The PFP3-4 data output has only three destinations. There is no SPI-3 to SPI-4 redirect path.

Each SPI-3 interface feeds ingress buffer available or ingress buffer unavailable status information to its packet fragment processor.

If the number of free segments available to a LP exceeds the starving threshold, the SPI-4 status is moved to starving for that LP. If the number of free segments available to a LP exceeds the hungry but not the starving threshold, the SPI-4 status is moved to hungry for that LP. If the hungry threshold is not exceeded, the SPI-4 FIFO status channel will indicate satisfied for that LP.

### SPI-4 ingress to SPI-3 egress datapath

The following is a description of the path taken by a burst of data through the device from the SPI-4 ingress to a SPI-3 egress.

Data enters on the SPI-4 ingress interface in bursts. Bursts are normally of equal length except the last burst of a packet which may be shorter. The control word is in-band with the data. Burst data enters a SPI-4 ingress buffer. SPI-4 LP address, error information, SOP, EOP are stored with the burst data. A SPI-4 LP address is mapped to a Logical Identifier (LID). The burst is stored in per LID allocated buffer segments reserved from the buffer segment pool.

The appropriate SPI-3 egress control register (Table 80 - SPI-3 egress port descriptor table (64 entries)) is consulted, and it determines to send this LID to a prescribed SPI-3 egress port.

The selection of which LP is to be transmitted next is dependent on the status of the LP and the availability of a complete fragment. Data is moved to the appropriate SPI-3 egress buffer along with the LP address. SPI-3 LP address, error information, SOP, and EOP information is stored with the packet fragment. Next, data is transmitted in packet fragments over the selected SPI-3 interface.

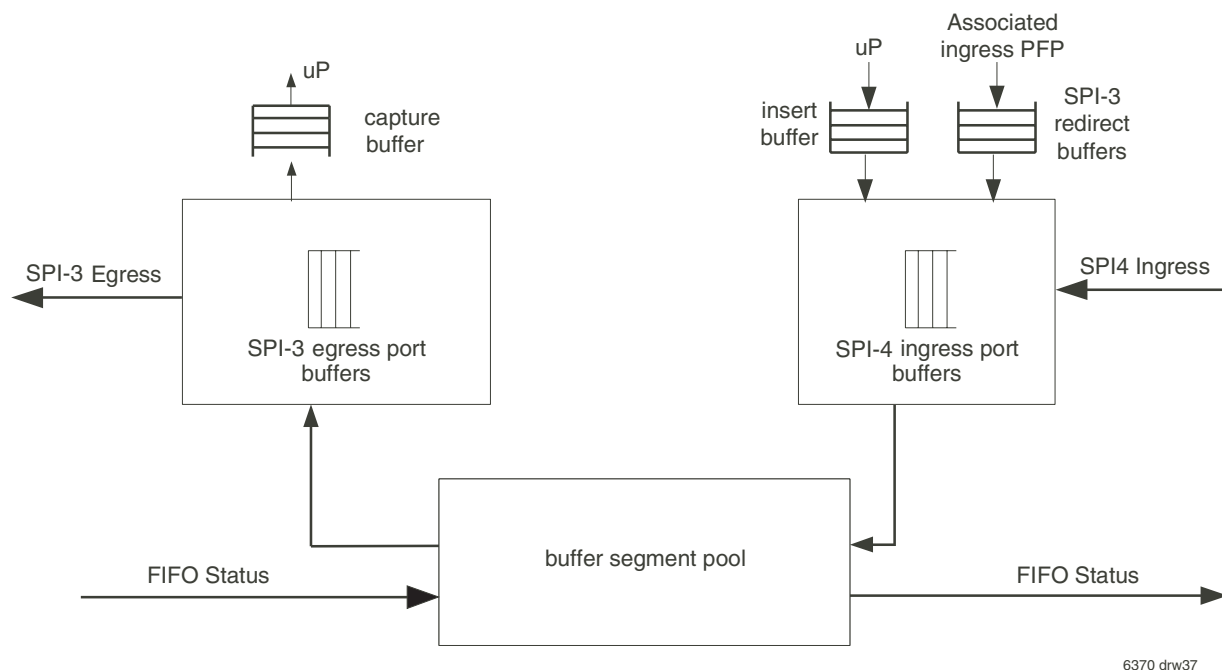


Figure 18. SPI-4 ingress to SPI-3 egress packet fragment processor

**SPI-4 ingress interface configurable parameters:**

The IDT88P8344 can interface to either a Link or a PHY layer device. The SPI-4 port can be enabled or disabled.

The SPI-4 ingress bits are aligned with the ingress clock. In addition, the SPI-4 words are then aligned among each other to produce valid words. This is performed both on the data channel and the status channel. The bit alignment algorithm runs as long as the interface is active. The word alignment algorithm is run during training intervals.

**SPI-4 ingress per-LID configurable parameters**

**SPI-4 to SPI-3 LID map**

- 256 entries, one per SPI-4 LP
- SPI-3 physical interface identifier
- SPI-3 LID
- Enable bit per LID

**SPI-4 ingress packet length check**

Each LID on the SPI-4 ingress interface has the ability to be programmed for minimum and maximum packet length. The minimum packet length can be set from 0 to 255 bytes in one byte increments. The maximum packet length can be set from 0 to 16,383 bytes in one byte increments. Packets shorter or longer than set by these parameters will be optionally counted in the short or long packet counter for that LID.

**SPI-3 egress configurable parameters**

**Length of SPI-3 packet fragment**

All packet fragments from a particular SPI-3 physical interface are programmable to an equal length with the possible exception of an EOP fragment which may be shorter.

**SPI-3 egress poll length**

Applies when the SPI-3 interface is acting as a Link layer device when using the packet level polling mode

Causes polling of the PHY for the logical ports associated with LIDs ranging from [0 up to POLL\_LENGTH] to find logical ports that can accept data

Poll range is 0-63 LPs.

**SPI-3 egress per-LID configurable parameters**

Many parameters to control the flow of data are programmable per LID. The following paragraphs describe these parameters.

**SPI-3 egress LID to LP map**

- one map per SPI-3 physical port
- 64 entries per map, one per LID
- LP enable bit per LP
- Bit reversal enable per LP

**SPI-3 egress multiple burst enable**

Multiple Burst Enable allows more than one burst to be sent to an LP. This feature is included to relieve systems with long latency between updates. When this feature is not enabled, only one burst per LP is allowed into the round robin SPI-3 egress buffers at a time.

**SPI-4 ingress to SPI-3 egress data memory**

**SPI-3 egress control**

There is a SPI-3 egress port descriptor table for the paths out of the data memory. The function a SPI-3 egress port descriptor table is to define where data goes after leaving the main data memory. There are three configurable options:

- SPI-3 egress
- Microprocessor Interface Capture
- Discard

**Maximum number of memory segments**

- Defines the largest Buffer available to an LP / LID
- Each segment is 256 bytes
- Range 1 – 508 in increments of one segment

The figure below shows the datapath through the device from the SPI-4 interface to the SPI-3 interface.

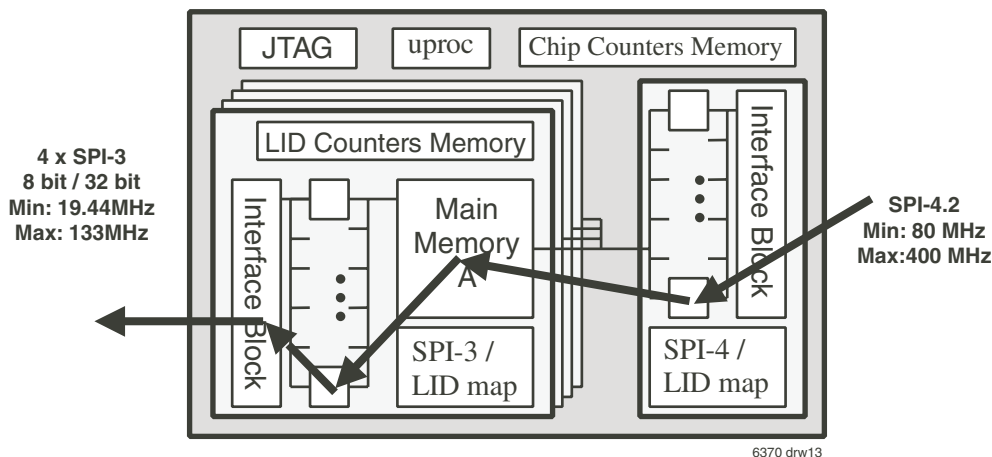


Figure 19. SPI-4 ingress to SPI-3 egress datapath

**SPI-4 ingress to SPI-3 egress flow control**

The SPI-4 control information is transmitted to the adjacent device. The adjacent device determines which LP to service next according to the status information it receives from the IDT88P8344. The SPI-4 ingress data arrive in bursts that are of equal length except for the last burst of a packet which may be shorter.

The SPI-4 burst data is transferred to the per LID allocated buffer segments. The addition of the data may cause an update of the SPI-4 status information (starving, hungry, satisfied), and may change the SPI-3 STPA, DTPA, or PTPA signals when in PHY mode.

For the control information there are two separate cases to consider: the case that the SPI-3 is configured to Link mode, and the case that the SPI-3 is configured

to PHY mode. Note that since the SPI-3 physical interfaces are configured separately, the IDT88P8344 device is able to deal with the case that some of the LP fragments are to be transmitted on a Link SPI-3 interface and some are to be transmitted on a PHY SPI-3 interface

When in PHY mode, the data is sent according to the availability of the data in the buffer segment pool. In the Link mode an extra consideration is taken to account – that of the fill level of the ingress FIFOs in the adjacent device.

**Backpressure threshold**

- Number of free segments allocated to trigger backpressure for the LP

The diagram below shows the SPI-4 to SPI-3 flow control path through the IDT88P8344 device.

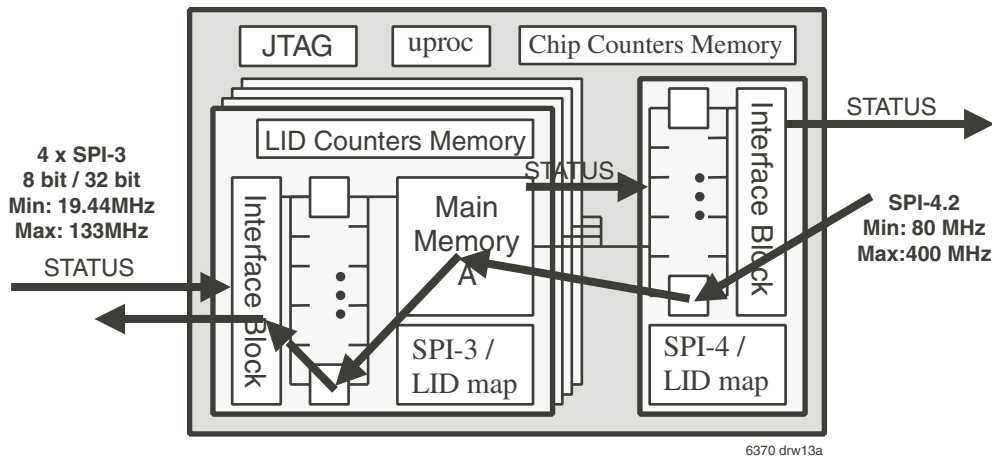


Figure 20. SPI-4 ingress to SPI-3 egress flow control



### 4.3 SPI-3 ingress to SPI-3 egress datapath

The SPI-3 redirect buffer can store SPI-3 packet fragments. The status of the packet fragment buffers is forwarded to the associated packet fragment processor. The purpose of the SPI-3 redirect is to enable per-LP flows between physical interfaces SPI-3 A and SPI-3 B; as well as between SPI-3 C and SPI-3 D. Other flows between SPI-3 ports are not allowed; *i.e.*, between A and A, A and C, A and D, B and B, B and C, B and D, C and C, and D and D.

The following is a description of the path taken by a fragment of data through the device.

SPI-3 modules are implemented in pairs (ports A & B or C & D). A SPI-3 to SPI-3 path is between an LP on one SPI-3 to the paired SPI-3. Data enters on the SPI-3 interface in fragments. Fragments are of equal length except the last fragment of a packet which may be shorter. The LP address is in-band with the data. The packet fragment enters an ingress buffer. SPI-3 LP address, error information, SOP, and EOP information is stored with the fragment. The LP address is mapped to a LID. The fragment is stored in buffer segment pool per-LID-allocated memory space.

The Table 80, SPI-3 egress port descriptor table (64 entries) is consulted, and the PFP decides to send a LID to the associated SPI-3 egress port. The SPI-3 packet fragment processor chooses the next LP. The choice of LP is dependent on status of the LP and availability of a complete fragment. Data is moved to an egress buffer along with the SPI-3 LP address, error information, SOP, and EOP information. Data is transmitted in packet fragments over a SPI-3 interface.

The paths to and from the microprocessor interface can be used to perform mappings from a SPI-3 LP to a SPI-3 LP where not provided, and from a SPI-4 to a SPI-4 LP. However these paths are limited by the bandwidth of the microprocessor interface.

The diagram below shows the datapath through the device from a SPI-3 interface to its paired SPI-3 interface. For the SPI-3 redirect, the LID connecting associated port pairs must be the same in both directions.

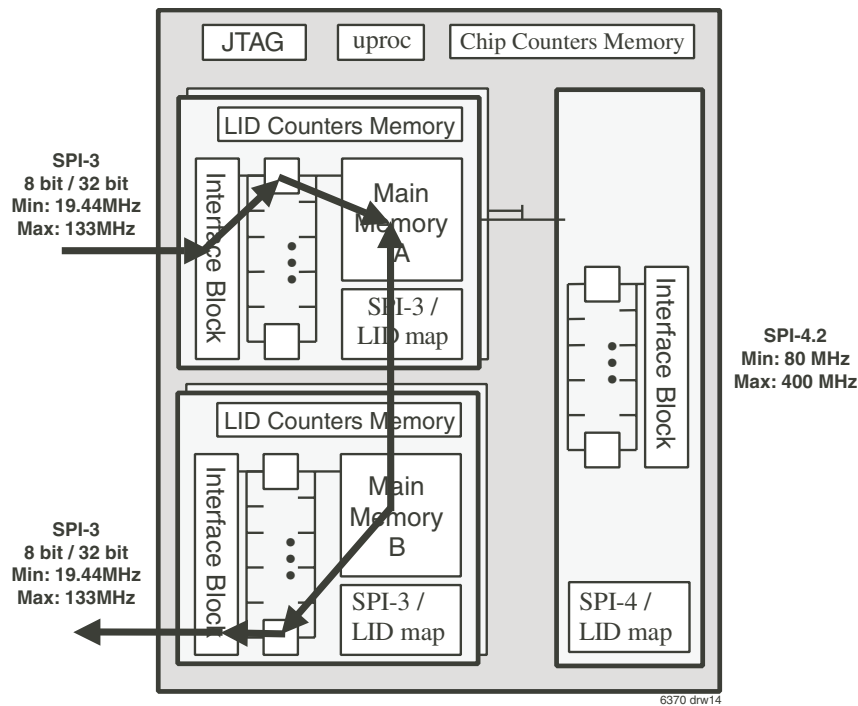


Figure 21. SPI-3 ingress to SPI-3 egress datapath

### 4.4 Microprocessor interface to SPI-3 datapath capture/insert configurable parameters

Enable insertion / capture of data to the SPI-3 or SPI-4 data stream (which is dependent on the egress control register). For each direction, the following are to be used:

- Data for insertion or data captured
- Data available: set when data is available. Asserted by device for capture, asserted by microprocessor for insertion.
- LID: Logical Identifier of capture / insertion channel
- Length: length of data for insertion or capture
- Flags: SOP, EOP, address parity error, data parity error, packet error

There are separate instantiations of microprocessor insert capture buffers for SPI-3 and SPI-4.

#### Capture data fragment

Packets can be captured from the SPI-3-4 stream and directed towards the microprocessor. The capture buffer can store only one 256 byte packet fragment. When the buffer is full the DATA\_AVAILABLE flag is set and a SPI-3 capture event is generated. The event is directed towards the interrupt module.

#### Read packet data fragment

The microprocessor needs to read a buffer to capture a packet fragment. It verifies the DATA\_AVAILABLE flag in the SPI-3 capture control register.

Microprocessor reads the packet fragment and EOP, SOP, ERROR, LID and LENGTH fields from the SPI-3 data capture buffer. Microprocessor hands over control of the capture buffer when it clears the DATA\_AVAILABLE flag in the SPI-3 data capture control register (Table 31 - SPI-3 data capture control register).

#### 4.4.1 SPI-3 to ingress microprocessor interface datapath

The diagram below shows the datapath through the device from the SPI-3 interface to the microprocessor capture interface.

The following is a description of the path taken by a fragment of data through the device.

Data enters on a SPI-3 interface in fragments. Fragments are of equal length except the last fragment of a packet which may be shorter. The LP address is in-band with the data. The fragment enters a SPI-3 ingress buffer. SPI-3 LP address, error information, SOP, and EOP are stored with the fragment. The LP address is mapped to a LID. The fragment is stored in LID allocated buffer segments.

The Table 80, SPI-3 egress port descriptor table (64 entries) is consulted, and the PFP decides to send this LID to the microprocessor capture port. Data is moved to the capture buffer along with the LP address. LID, error information, SOP, and EOP. The data available bit is set. Data and control information are read from the relevant register space through the microprocessor interface.

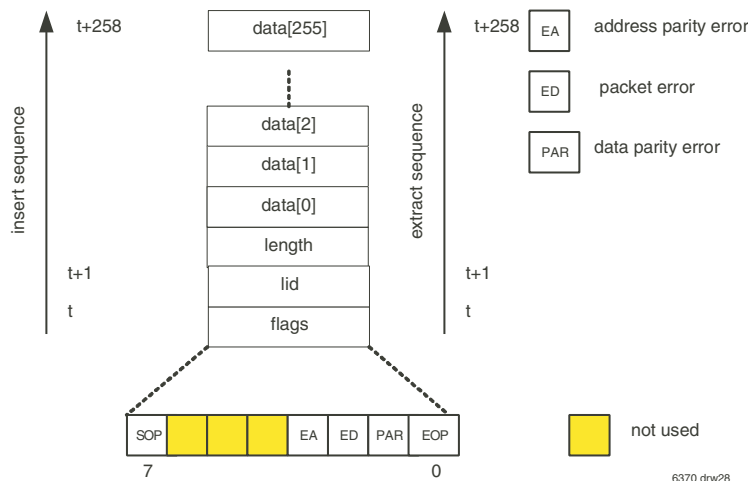


Figure 22 . Microprocessor data capture buffer

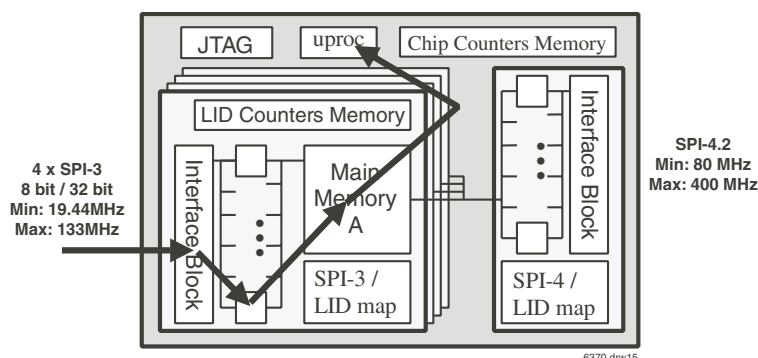


Figure 23. SPI-3 ingress to microprocessor capture interface datapath

### 4.4.2 Microprocessor insert to SPI-3 egress datapath

The diagram below shows the datapath through the device from the microprocessor data insert interface to a SPI-3 egress port.

The following is a description of the path taken by a fragment of data through the device.

Data and control information are written to the insert buffer through the microprocessor interface. The data available bit is set. Data is stored along with its LP address, LID (including SPI-3 choice), error information, SOP, and EOP.

Data is stored in LID-allocated buffer segments. The Table 80, SPI-3 egress port descriptor table (64 entries) is consulted and the PFP decides to move the data to the SPI-3 egress port. The SPI-3 packet fragment processor chooses the next LP. The choice of LP is dependent on the status of the LP and the availability of a complete fragment. Data is moved to a SPI-3 egress buffer along with its LP address. SPI-3 LP address, error information, SOP, and EOP.

Data is transmitted in packet fragments over the selected SPI-3 egress interface.

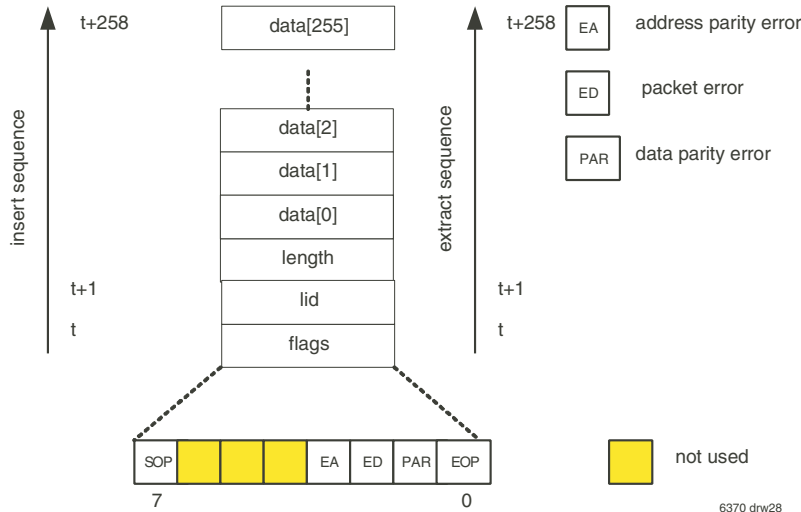


Figure 24 . Microprocessor data insert buffer

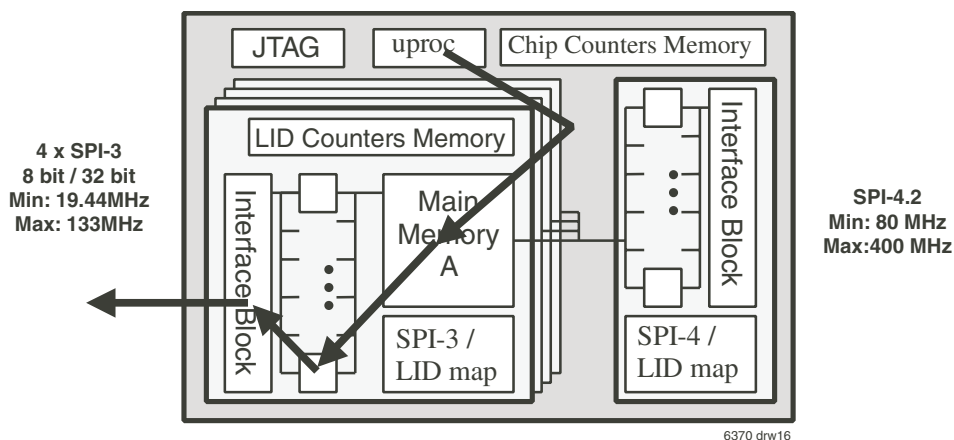


Figure 25. Microprocessor interface to SPI-3 egress detailed datapath diagram

### 4.4.3 Microprocessor interface to SPI-4 egress datapath

Packets can be inserted into the SPI-3-4 datapath by the microprocessor. The following is a description of the path taken by a burst of data through the device.

Data and control information are written to the insert buffer through the microprocessor interface. The data available bit is set. Data is stored in the insert buffer along with the LP address, LID, error information, SOP, and EOP. Data is stored in per-LID allocated buffer segments. The Table 36-SPI-3 data insert control register is consulted, and determines to send this LID to the SPI-4 egress port. The SPI-4 Packet Fragment Processor chooses the next LP. Data is sent to the SPI-4 egress buffer along with the SPI-4 LP address, error information, SOP, and EOP. Data is transmitted in bursts over the SPI-4 egress interface.

The microprocessor needs to write data into a dedicated buffer to insert a packet burst. Refer to Figure 26, *Microprocessor data insert buffer* for the data format in the buffer. The microprocessor must verify the DATA\_AVAILABLE flag in the SPI-4 insert control register and waits until the flag is cleared. The microprocessor specifies the EOP, SOP, ERROR, LID and LENGTH fields and writes up to 256 bytes of packet fragment burst into the insert buffer. The packet burst insert buffer is accessed through the Table 34, SPI-4 data insert register (register\_offsets 0x03, 0x0B, 0x13, 0x1B) SPI-4 data insert register. The microprocessor hands over control of the buffer setting the DATA\_AVAILABLE flag in the SPI-4 insert control register. A SPI4\_insert\_empty event is generated when the DATA\_AVAILABLE flag is cleared. The event is directed towards the interrupt module.

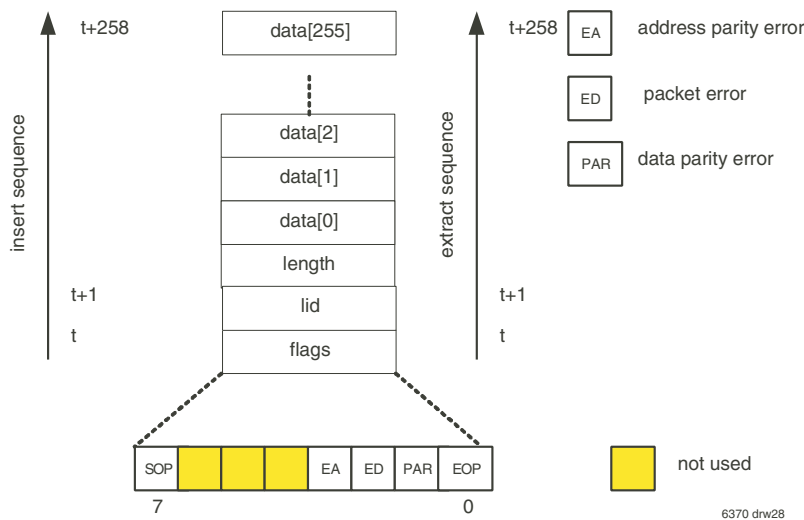


Figure 26. Microprocessor data insert buffer

The diagram below shows the datapath through the device from the microprocessor data insert interface to the SPI-4 egress interface.

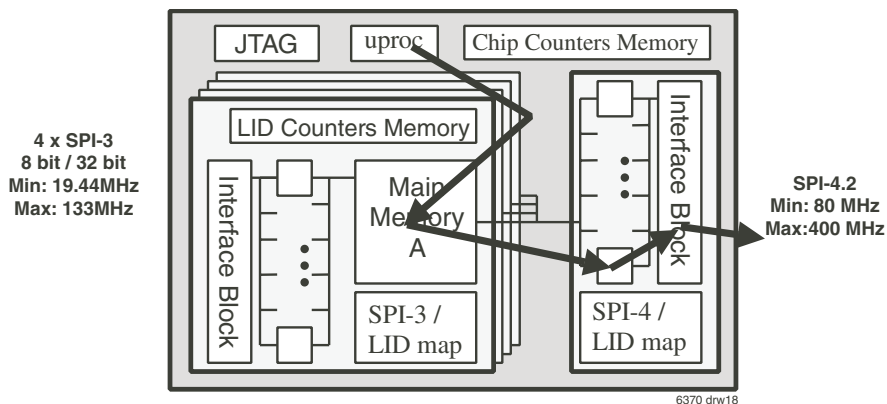


Figure 27. Microprocessor data insert interface to SPI-4 egress datapath

### 4.4.4 SPI-4 ingress to microprocessor interface datapath

The diagram below shows the datapath through the device from the SPI-4 interface to the microprocessor data capture interface.

The following is a description of the path taken by a burst of data through the device.

Data enters on the SPI-4 interface in bursts. Bursts are normally of equal length except the last burst of a packet which may be shorter. The control word is in-band with the data. The burst data enters a SPI-4 ingress buffer. SPI-4 LP

address, error information, SOP, and EOP are stored along with the burst data. The SPI-4 LP address is mapped to a LID. Data is stored in per-LID allocated buffer segments. The DIRECTION field of the SPI-3 egress port descriptor (Block\_base 0x1700 + Register\_offset 0x00 - 0xFF) is used to send this LID data to the microprocessor port. Data is moved to the capture buffer along with the LP address, LID, error information, SOP, and EOP.

The data available bit is set by the PFP. Data and control information are read from the capture buffer through the microprocessor interface.

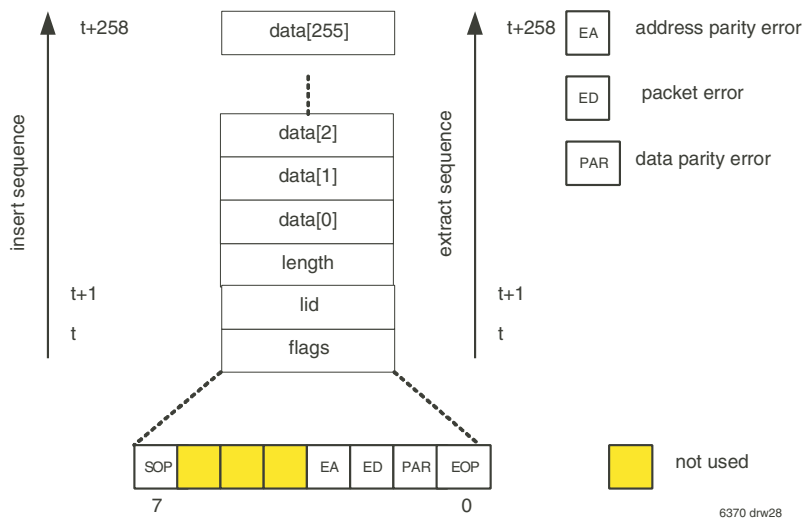


Figure 28. Microprocessor data capture buffer

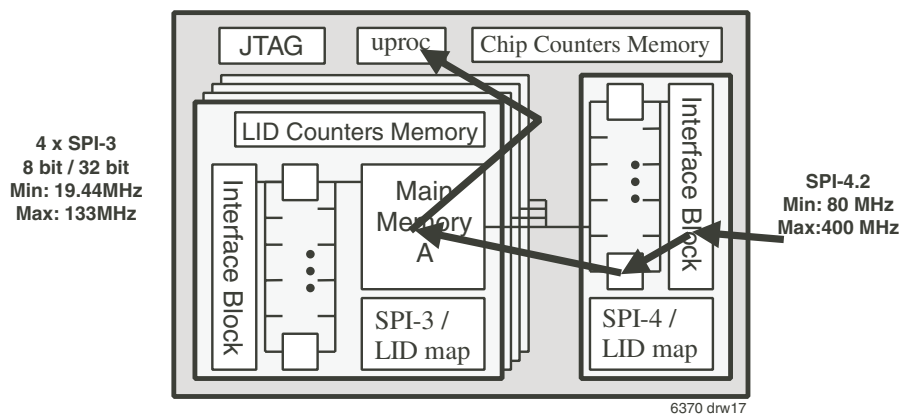


Figure 29. SPI-4 ingress to microprocessor data capture interface path

## 5. PERFORMANCE MONITOR AND DIAGNOSTICS

### 5.1 Mode of operation

A performance monitor & diagnostics module is available in modules A, B, C, and D. The performance monitor captures events and accumulates error events and diagnostics data. Some performance monitor accumulators are associated to a physical port, some to a LID.

### 5.2 Counters

All events and diagnostics data are accumulated during an interval defined by the timebase event. The data accumulated during the previous time period can be accessed by the indirect access scheme. The counters are cleared when the timebase expires. All counters are saturating, and will not overflow.

#### 5.2.1 LID associated event counters

A set of event counters is provided for each of the 64 LPs on each SPI-3 interface and for each LID to/from the SPI-4 module.

A packet is delineated by an SOP and EOP on the SPI-3/SPI-4 logical port. It is defined as "bad" when the packet is tagged with an error.

All packets that are not "bad" are considered "good".

For more information refer to Table 60 - LID Associated Event Counters (0x000-0x17F).

#### 5.2.2 Non - LID associated event counters

A set of event counters is provided for each of the SPI-3 and SPI-4 physical interfaces.

Refer to Table 61, Non LID associated event counters (0x00-0x0B) for the offset in the indirect access space, and for the events recorded.

### 5.3 Captured events

Two categories of events are captured: LID and non LID associated events. If at least one event is captured in one of the interrupt indication registers, an active PMON service request is directed towards the interrupt module.

#### 5.3.1 Non LID associated events

Non LID associated events are captured into the Table 62 - Non LID associated interrupt indication register (Block\_base 0x0C00 + Register\_offset 0x00 to 0x0B). An interrupt is generated if the event is enabled by its enable flag in the Table 63 - Non LID associated interrupt enable register (Block\_base 0x0C00 + Register\_offset 0x0D). The interrupt is cleared by writing a logical one to the Table 62 - Non LID associated interrupt indication register (Block\_base 0x0C00 + Register\_offset 0x00 to 0x0B).

#### 5.3.2 LID associated events

Two types of LID associated events are captured. Non critical events are defined in Table 64 - LID-associated interrupt indication register (0x0E) and are associated with the physical interface. Critical events are defined as buffer overflows within the IDT88P8344 device in Table 67, SPI-3 to SPI-4 critical LID interrupt indication registers (register\_offset 0x16-0x17).

#### 5.3.2.1 Non critical events

LID associated non critical events are captured in the Table 64, LID-associated interrupt indication register (0x0E). An interrupt is generated if the interrupt is enabled by its enable flag in the Table 65, LID-associated interrupt enable register (0x0F). The interrupt indication is cleared by writing a logical one to the Table 64, LID-Associated Interrupt Indication Register (0x0E).

When the event is captured, the LID or LP associated with the event is captured in Table 66, Non-Critical LID-Associated Capture Table (0x10-0x15). The table records the latest captured LID or LP.

#### 5.3.2.2 Critical events

Critical events are captured per LID in Table 67, SPI-3 to SPI-4 Critical LID interrupt indication registers (Block\_base 0x0C00 + Register\_offset 0x16-0x17) and Table 69, SPI-4 to SPI-3 critical LID interrupt indication registers (0x1A-0x1B). An interrupt is generated if enabled by the corresponding enable flag in the Table 68, SPI-3 to SPI-4 critical LID interrupt enable registers (0x18-0x19) and Table 70, SPI-4 to SPI-3 critical LID interrupt enable registers (0x1C-0x1D). The indication is cleared by writing a logical one to the Table 67, SPI-3 to SPI-4 critical LID interrupt indication registers (0x16-0x17) or Table 69, SPI-4 to SPI-3 critical LID interrupt indication registers (0x1A-0x1B). Only one kind of critical event is defined, buffer overflow. Since there are  $64 \times 2 = 128$  critical LID associated event sources, two source indication bits are contained in Table 71, Critical events source indication register (0x1E). The bits are read only. Bit SPI34\_OVR reflects the OR result of all bits in Table 67, SPI-3 to SPI-4 critical LID interrupt indication registers (0x16-0x17). Bit SPI43\_OVR reflects the OR result of all bits in Table 69, SPI-4 to SPI-3 critical LID interrupt indication registers (0x1A-0x1B).

### 5.3.3 Timebase

A single timebase module is provided in the device. The timebase period can be configured to be internally or externally generated. A snapshot of the counters is taken when the timebase expires and the counters are cleared. The snapshot registers are accessed by an indirect access scheme.

#### 5.3.3.1 Internally generated timebase

The period of the timebase is configured for the device using the register defined in Table 120, Timebase register (Register\_offset 0x01). The configuration specifies the number of master clock (MCLK) cycles required for each period. For a description of MCLK refer to Chapter 6 Clock generator. The timebase event is captured by the timebase status in Table 45, Secondary interrupt status register (0x2D in the direct accessed space).

The internal timebase is generated either by the microprocessor or by a free running timer input. The selection is made by the TIMER flag in the Table 119, PMON update control register (Register\_offset 0x00). When the time interval expires, the TIMEBASE pin is asserted for sixteen MCLK cycles.

#### 5.3.3.2 Externally generated timebase

The externally generated timebase signal is applied on the TIMEBASE pin. A positive edge detector generates the timebase event. The timebase event is captured by the timebase status in the Table 45 - Secondary Interrupt Status Register (0x2D in the direct accessed space).

## 6. CLOCK GENERATOR

The device generates clocks from the SPI-4 ingress clock (I\_DCLK) or from the REF\_CLK input pin. The clock so selected is used for core functions of the device, and must be present during reset and thereafter. The selection and frequency divisors are defined by CK\_SEL[3:0] pins as defined in the following Table 14, CK\_SEL[3:0] input pin encoding.

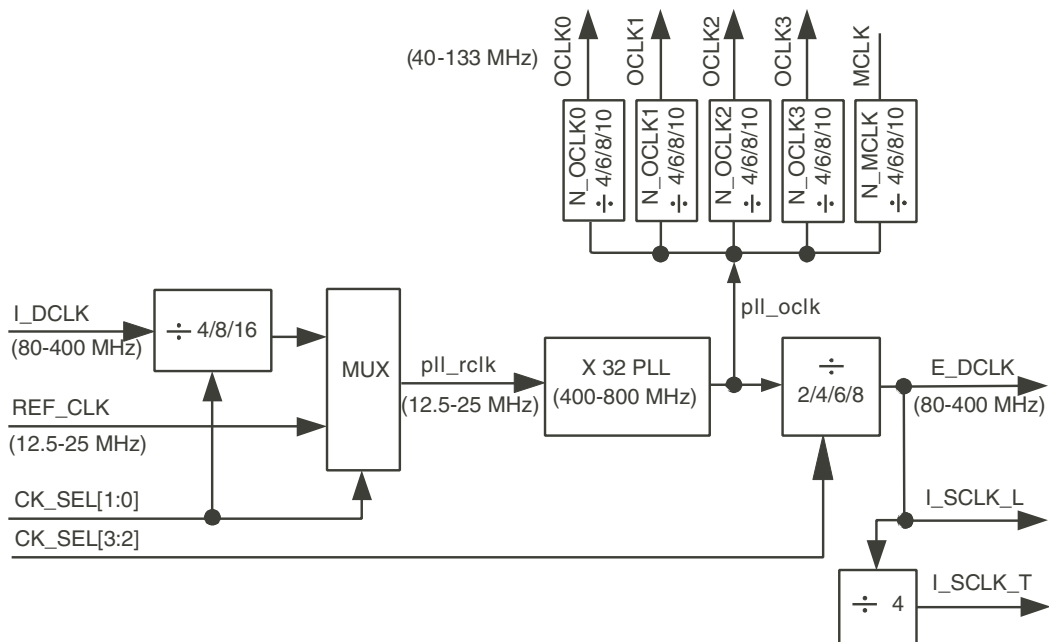
The clock generator provides four clock outputs on the OCLK[3:0] pins, MCLK for internal use, and SPI-4 data and FIFO status channel egress clocks. The OCLK[3:0] clock frequencies can be selected independently of each other. OCLK[3:0] outputs always have a relative output skew of one pll\_ock (refer to Figure 30 Clock generator) to prevent simultaneous switching when used as SPI-3 clock sources. Use of the OCLK[3:0] outputs is encouraged for the SPI-3 clock inputs to reduce system jitter. The frequency is divided according to the value selected in the clock generator control register shown below. The

OCLK[3:0] pins are separately enabled by setting each associated enable flag in Table 121, Clock generator control register (Register\_offset 0x10). When an OCLK[3:0] output is not enabled, it is in a logic low state. MCLK is the internal processing clock, and is always enabled. Divide options should be selected to keep the internal PLL output pll\_ock within its operating frequency range of 400 to 800 MHz. Refer to Table 122, OCLK and MCLK frequency select encoding for selecting the frequencies of MCLK and OCLKs. Note that divider values should be chosen so that OCLK[3:0] and MCLK are within their specified operating range provided in Table 136, OCLK[3:0] clock outputs and MCLK internal clock.

During either a hardware or a software reset, the OCLK[3:0] pins are all logic low. Immediately following reset, all OCLK[3:0] outputs are active with the output frequency defined by pll\_ock divided by the initial value in the Table 121, Clock generator control register (Register\_offset 0x10).

**TABLE 14 – CK\_SEL[3:0] INPUT PIN ENCODING**

CK_SEL[1:0]	Function
00	pll_rclk = REF_CLK
01	pll_rclk = I_DCLK/16
10	pll_rclk = I_DCLK/8
11	pll_rclk = I_DCLK/4
CK_SEL[3:2]	Function
00	E_DCLK = pll_ock/2
01	E_DCLK = pll_ock/4
10	E_DCLK = pll_ock/6
11	E_DCLK = pll_ock/8



6370 drw21

**Figure 30. Clock generator**

## 7. LOOPBACKS

Local loopbacks are supported on each of the SPI-3 physical ports. They are described below.

### 7.1 SPI-3 Loopback

A SPI-3 physical port loop back is supported on each of the SPI-3 interfaces. In this mode, the contents of the SPI-3 ingress buffers are directly transferred to the SPI-3 egress buffers. All data and error information received on an ingress interface of a SPI-3 physical port is transmitted on the egress interface of the same SPI-3 physical port.

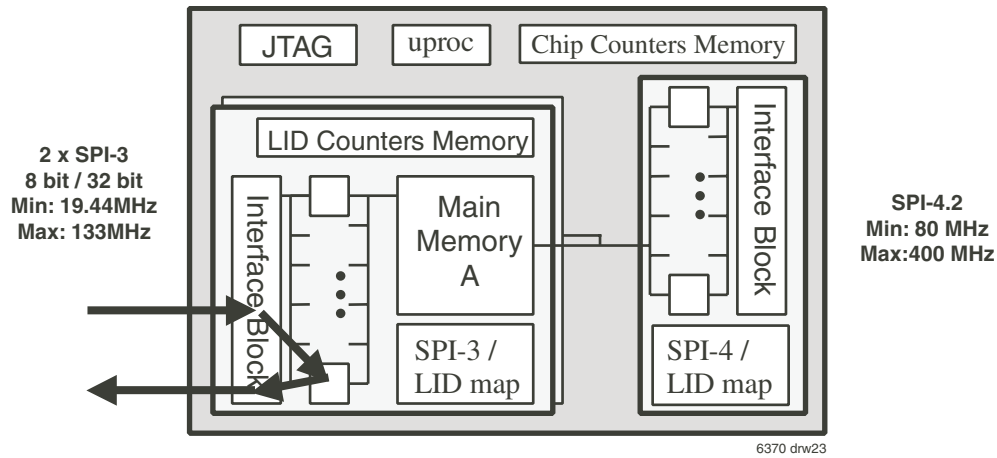


Figure 31. SPI-3 Loopback diagram



## 8. OPERATION GUIDE

### 8.1 Hardware operation

#### 8.1.1 System reset

There are two methods for resetting the device: hardware reset & software reset. During reset the output clocks are not toggled.

##### Hardware reset

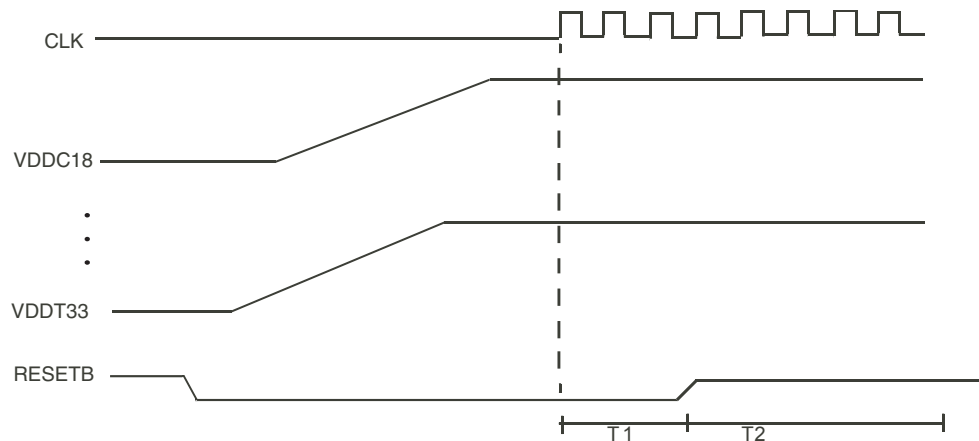
The RESETB input requires an active low pulse to reset the internal logic.

##### Software reset

The software reset is triggered by setting to 1 the SW\_RESET field in direct register Software Reset Register (p.51). The response to a software reset is identical to a hardware reset except that software reset does not change the N\_OCLK[3:0] fields in the Clock Generator Control Register (p.77), so it does not impact the clock generators. The SW\_RESET field is self-clear to 0 after the device initialized itself. After software reset the external microprocessor should have delay of at least 1ms before accessing the device, and then. After the 1ms delay, the user should poll the INIT\_DONE field in the Software Reset Register (p.51), and wait till it is 1. When the INIT\_DONE field is 1, the user should download a boot code from the external microprocessor flash to the device embedded processor RAM.

#### 8.1.2 Power on sequence

A correct power-on-reset sequence is crucial for the normal behavior of the device. The power-on-reset sequence includes the following signals:



6370 drw23a

Figure 32. Power-on-Reset Sequence

#### 8.1.3 Clock domains

The chip has several clock domains. The related registers can not be configured without each clock. It is necessary to supply the clocks that are pertinent to the registers being initialized for the initialization to succeed. In order to access the microprocessor interface, MCLK must be active, either by selecting and providing a stable REF\_CLK input, or by selecting and ensuring that a stable clock is always present on the I\_DCLK input. The selection of either the REF\_CLK or the I\_DCLK clock inputs is described in Table 14 CK\_SEL[3:0] input pin decoding.

CLK (REF\_CLK or I\_DCLK - depends on which of these pins are selected), VDDT33, VDDC12 and RESETB. Figure 32, Power-on-Reset Sequence illustrates the recommended implementation for the power-on-reset sequence for the device. IDT recommends powering up the VDD33 power supply first, and the VDDC18 power supply last. The power supplies can be also powered up in the same time. There is no requirement for the minimum or maximum delay between the power-up of the power supplies. The power supplies should be powered off in the reverse order. The power ramp should not be fast than 100us, but also not too slow.

When the power supplies are powered up, the RESETB signal should be at low level. During power-on-reset, after the VDDT33, VDDC18, CLK (REF\_CLK or I\_DCLK - depends on which of these pins are selected) and the configuration signals are stable, the RESETB signal should remain at a low level at least 10ms (symbol "T1") to reset the internal logic. After the RESETB pulse ends, the device starts generating the SPI-4 / SPI-3 external output clocks & the MCLK internal clock.

After the RESETB pulse ends, a delay of 1ms should be added (symbols "T2") before accessing the device for initialization and configuration. This allows the internal logic to be stable. During T2 (at least 1ms delay) the device performs internal memories initialization.

After T2, the user should poll the INIT\_DONE field in the in the Software Reset Register (p.51), and wait till it is 1. When the INIT\_DONE field is 1, the user should download a boot code from the external microprocessor flash to the device embedded processor RAM.

## 8.2 Software operation

### 8.2.1 Chip configuration sequence

For proper device operation, it is important to initialize the IDT88P8344 in the correct sequence following reset. This sequence is outlined in the following paragraphs.

1) Reset the IDT88P8344 chip. After reset, the chip will perform auto initialization. Wait for the chip initialization to complete. The INIT\_DONE flag will go high when initialization has been completed.

- 2) Configure the clock generator as follows:
    - a) Configure the value for the MCLK divider, OCLK dividers and enables in the Clock Generator Control Register (refer to Table 121, Clock generator control register (Register\_offset 0x10)).
    - b) Configure the values for I\_LOW and E\_LOW (refer to Table 89, SPI-4 ingress configuration register (0x00) and Table 104, SPI-4 egress configuration register\_0 (Register\_offset 0x00)).
  - 3) Load status channel firmware. See section 8.2.5 for details.
  - 4) Configure all PFPs as follows:
    - a) The NR\_LID fields must be configured first. Do not change the NR\_LID fields once configured after reset. The NR\_LID fields are in the Table 76, SPI-3 to SPI-4 PFP register (register\_offset 0x00) and the Table 83, SPI-4 to SPI-3 PFP register (0x00).
    - b) There are four sets of port descriptor tables: The Table 73, SPI-4 egress port descriptor table (64 entries), the Table 75, SPI-3 ingress port descriptor table (Block\_base 0x1200), the Table 80, SPI-3 egress port descriptor table (64 entries), and the Table 82, SPI-4 ingress port descriptor tables (64 entries). Configure the M (SPI-4 egress), MAX\_BURST\_S, MAX\_BURST\_H, DIRECTION (SPI-4 egress), FREE\_SEGMENT, MAX\_BURST, DIRECTION (SPI-3 egress), M (SPI-4 ingress), FREE\_SEGMENT\_S, and FREE\_SEGMENT\_H parameters of the port descriptor tables for each LID to be activated. The total buffer segment assignment should not exceed the available buffer segment pool capacity of 508 segments per PFP.
  - 5) Configure the Table 54- SPI-3 egress LID to LP map.
  - 6) Configure the Table 49 - SPI-3 ingress LP to LID map.
  - 7) The SPI-4 Calendar tables must be configured before the SPI-4 mapping tables are configured. Do not change the SPI-4 calendar tables once they are configured. In LVDS status mode, ensure that the value of  $(CAL\_LEN+1) * (M+1)$  is at least 4.
    - a) Configure the SPI-4 ingress calendar or calendars (refer to Table 87, SPI-4 ingress calendar\_0 (256 entries) and Table 88, SPI-4 ingress calendar\_1 (256 entries)).
    - b) Configure the SPI-4 egress calendar or calendars (refer to Table 102, SPI-4 egress calendar\_0 (256 locations) and Table 103, SPI-4 egress calendar\_1 (256 locations)).
  - 8) Configure the Table 101, SPI-4 egress LID to LP map (256 entries).
  - 9) Configure the Table 86, SPI-4 ingress LP to LID (256 entries, one per LP).
  - 10) Configure the SPI-4 physical interface. The Table 89, SPI-4 ingress configuration register (0x00), Table 104, SPI-4 egress configuration register\_0 (Register\_offset 0x00), and Table 105, SPI-4 egress configuration register\_1 (Register\_offset 0x01) must be configured before enabling the SPI-4 physical interface. Once the SPI-4 interface is enabled, the SPI-4 interface configuration registers can not be changed unless the chip is reset. Individual LIDs can still be enabled or disabled, within the bounds set by NR\_LID.
    - 11) Configure the SPI-3 physical interfaces. The Table 75, SPI-3 ingress port descriptor table (Block\_base 0x1200) must be configured before enabling the SPI-3 physical interfaces. Once the SPI-3 interfaces are enabled, Table 80, SPI-3 egress port descriptor table (64 entries) can not be changed unless the chip is reset. Individual LIDs can still be enabled or disabled without a chip reset, within the bounds set by NR\_LID.
    - 12) Enable the SPI-3 physical interfaces. Set the enable bit per LID in Table 46, SPI-3 ingress LP to LID map and Table 54, SPI-3 egress LID to LP map.
    - 13) Enable the SPI-4 physical interface. Set the enable bit per LID in Table 82, SPI-4 ingress LP to LID map (256 entries, one per LP) and Table 101, SPI-4 egress LID to LP map (256 entries).
- Note: Sufficient edge transitions on the bus are required to cause a change

in the TAP value. Therefore, the adjacent device should send training for at least 100ms in the end of the initialization sequence & before starting to send data. The bit alignment will select the best tap for each lane.

## 8.2.2 Logical Port activation and deactivation

### ***Dynamically deactivate a logical port***

The procedure for deactivating a logical port is outlined as follows:

- 1) Configure the enable bit of the ingress LP to LID map to "disabled".
- 2) Configure the egress data DIRECTION field in Table 73, SPI-4 egress port descriptor table (64 entries) or Table 80, SPI-3 egress port descriptor table (64 entries) to "discard".
- 3) Wait at least 0.1 ms for the flush of the remaining data in the queue.
- 4) Change M to 0 for the LP.

### ***Dynamically activate a logical port***

The procedure for activating a logical port is outlined as follows:

- 1) Make sure the LID is inactive.
- 2) Configure M for the desired LID.
- 3) Configure the egress LID to LP map, then enable the LID.
- 4) Configure the egress data direction for the LID.
- 5) Configure the ingress LP to LID map, then enable the LP.

## 8.2.3 Buffer segment modification

### ***Modification of the buffer segment allocation for a LID***

The buffer segment allocation can be changed while the corresponding LP is disabled. The amount of buffering available for a LID can be decreased or it can be increased if more buffer segments are available for use. The procedure for changing the buffer segment allocation for a LID is outlined as follows:

- 1) Disable the LP corresponding to the LID to undergo buffer segment modification.
- 2) Discard the fragments, by configuring the DIRECTION field for DISCARD for the LID.
- 3) Wait at least 0.1 ms for the buffer to empty.
- 4) Change M to 0 for the LID.
- 5) Configure the new M value for the LID.
- 6) Configure the DIRECTION field for the LID to restore data flow.
- 7) Enable the LP corresponding to the LID that underwent buffer segment modification.

## 8.2.4 Manual SPI-4 ingress LVDS bit alignment

The procedure for manually adjusting the SPI-4 ingress LVDS bit alignment is outlined. It is recommended to use automatic alignment in most cases.

- 1) Configure the FORCE bit from Table 99, SPI-4 ingress bit alignment control register (register\_offset 0x11). This puts the SPI-4 ingress under manual control.
- 2) Configure the SPI-4 ingress data lane to be measured for clock-data alignment in Table 111, SPI-4 ingress lane measure register (register\_offset 0x01), LANE field.
- 3) Wait for the eye measurement to complete in Table 111, SPI-4 ingress lane measure register (register\_offset 0x01), MEASURE\_busy field.
- 4) Read the eye pattern counter in Table 112, SPI-4 ingress bit alignment counter register (0x02 to 0x0B).
- 5) Calculate the proper value.
- 6) Configure the appropriate phase tap in Table 113, SPI-4 ingress manual alignment phase/result register (0x0C to 0x1F).

### 8.2.5 SPI-4 status channel software

The SPI-4 status channel may be configured to either LVTTTL or LVDS by loading the appropriate status channel binary file to activate the firmware. Download LVTTTL.bin when using LVTTTL status mode. Download LVDS.bin when using LVDS status mode. This step should be performed as the third step in the chip configuration sequence after reset in section 8.2.1.

The download process is described.

```
Direct write (0x20, 0x01); /* Write register 0x20 with 0x01 to reset */
Delay at least 5ms
Direct write (0x36, 0x07);
ind_write(0x00c8, 0xdc8b0);
```

Open LVTTTL.bin or LVDS.bin file

```
number = file length
addr = 0x0e00;
if ( number % 2 == 0 )
    number /= 2;
else
    number = number/2 + 1;

for ( i = 0; i < number; i ++ )
{
    scr_fp.Read(ch, 2);
    data = (ch[1] << 8) | ch[0];
    ind_write(addr, data);
    addr ++;
    addr ++;
}
close file
ind_write(0x00c6, 0x0e00);
ind_write(0x00c8, 0xc860);
Direct write (0x36, 0x00);
```

### 8.2.6 IDT88P8344 layout guidelines SPI-3 LAYOUT GUIDELINES

1) Series terminate SPI-3 traces that are greater than 1/2 inch in end-to-end length. Place the series resistor as close as possible to the driver, but no more than 1/2 inch away from the driving end. SPI-3 inputs must have ringing controlled to prevent the SPI-3 inputs from going more than 0.5 Volts below ground. Use the IBIS models for more accurate results with the specific devices being used.

2) Minimize all SPI-3 data and control trace lengths to not exceed the  $T_{D-MAX} - T_{SETUP}$  requirement. For example, if the SPI-3 clock is 104 MHz,  $T_{D-$

$T_{D-MAX}$  of a device is 5.65 ns, and  $T_{SETUP}$  of the attached device is 1 ns, the maximum PCB trace delay Table 18 permitted is 3 ns ( $Unit\ Interval - T_{D-MAX} - T_{SETUP}$ ). This translates to a maximum PCB trace length for data and control lanes of 13.5 inches, if the loaded PCB trace delay is 220 picoseconds per inch. This is for zero  $T_{SETUP}$  margin, and does not include any margin for clock driver skew. Clock driver or clock trace skew could reduce the  $T_{SETUP}$  margin in this example.

3) Match all SPI-3 clock lengths to within the  $T_{D-MIN} - T_{HOLD}$  requirement. For example, if  $T_{D-MIN}$  for the device is 1.5 ns, and  $T_{HOLD}$  for the attached device is 0.5 ns, the worst case PCB clock trace skew for zero  $T_{HOLD}$  margin (defined in this example as the maximum PCB trace delay that the SPI-3 ingress clock of the attached device can exceed the trace delay of the SPI-3 egress clock of the device and still meet the  $T_{HOLD}$  requirement of the attached device with zero margin, assuming the fastest device [ $T_{D-MIN}$ ] and the worst case  $T_{HOLD}$  for the attached device and no trace delay on the data and control lanes) is 1.7 ns ( $Table\ 15\ (T_{D-MIN} - T_{HOLD})$ ), for a maximum PCB clock trace difference of 7.6 inches. Trace delay on the data and control lanes would improve the  $T_{HOLD}$  margin in this example. This example does not include any margin for SPI-3 clock buffer skew.

4) Ensure a few nanoseconds of clock delay between one SPI-3 clock net and other SPI-3 clock nets of the same frequency to minimize simultaneous switching noise. The IDT88P8344 OCLK[3:0] outputs have skew between each output already built in, and so are useful in lowering simultaneous switching noise. A SPI-3 clock net is defined to be the SPI-3 egress clock for a device and the SPI-3 ingress clock for the attached device.

5) Route all SPI-3 traces as 50 Ohm embedded stripline (inner layer referencing ground planes). For example, 8 mil wide 1/2 oz copper traces sandwiched between ground planes with 10 mil dielectric spacing between ground planes and signal planes yields 52 Ohms single-ended, using FR-4 with a relative dielectric constant ( $\epsilon_r$  or  $D_k$ ) of 4.2. If the edge to edge spacing between adjacent SPI-3 series terminated signals is 20 mils in this example, crosstalk between adjacent signals can be kept to 2%. Use a field solver for more accurate results.

An example timing budget Table 15, Zero Margin SPI-3 Timing budget, and example trace lengths to achieve timing margin Table 16, Margin check for SPI-3 timing, are shown. These timing budget tables do not include clock driver relative skew incurred if different drivers are used for a SPI-3 egress and its attached SP-3 ingress. These tables are based on timing only and do not include such effects as crosstalk and rise time degradation.

### SPI-4 LAYOUT GUIDELINES

1) Match the P and N trace lengths within an LVDS differential signal pair to within 100 mils or less.

2) Match the group of all differential data, control, and clock signal lengths to within 1/2 unit interval (DDR), or less, of each other (1/4 clock period). For

**TABLE 15 - ZERO MARGIN SPI-3 TIMING BUDGET**

SPI-3Clock	Tsetup	Thold	Td, minimum	Td, maximum	Unit Interval	Maximum data trace delay	Maximum data trace length	Maximum clock skew	Maximum Clock Δtrace length
104 MHz	1 ns	0.65 ns	2.33 ns	5.65 ns	9.6 ns	3 ns	13.5 in	1.7 ns	7.6 in

**TABLE 16 - MARGIN CHECK FOR SPI-3 TIMING**

SPI-3Clock	Tsetup	Thold	Td, minimum	Td, maximum	Egress clock trace	Ingress clock trace	Longest data trace	Shortest data trace	Tsetup margin	Thold margin
104 MHz	1 ns	0.65 ns	2.33 ns	5.65 ns	4 inches	8 inches	6 inches	4 inches	2.33 ns	1.48 ns

example, a SPI-4 clock of 400MHz gives a data unit interval of 1.25ns, so match the lengths within the entire signal group to within 625 ps, or 3 inches.

3) Keep P and N signals within a differential pair on the same layer with the minimum trace spacing possible while still being able to get 100 ohms differential impedance (tightly edge-coupled pair routing).

4) Route all differential pairs as 100 Ohm embedded differential stripline (on an inner layer, referencing ground planes). For example, 7 mil wide 1/2 oz copper traces separated by 10 mils, with 10 mil dielectric spacing to ground planes above and below the traces gives 100 Ohms of differential impedance for FR-4 with a relative dielectric constant ( $\epsilon_R$  or  $D_R$ ) of 4.2. If the edge to edge spacing between adjacent differential pair traces is 20 mils, crosstalk is 0.6% for signals terminated to within a 10% impedance match. If the edge to edge spacing between a differential pair and an LVTTTL signal is 30 mils within the parameters of this example, crosstalk is 0.8% (with the LVTTTL signals series terminated). Use a field solver for more accurate results.

5) Follow the SPI-3 layout guidelines for any routed SPI-4 LVTTTL status signals.

**GENERAL LAYOUT GUIDELINES**

1) Keep LVDS signals far from LVTTTL signals: at least three times the dielectric thickness to the reference plane (or three times the trace separation, whichever is greater) in separation width, to minimize the crosstalk contribution of noise on the LVDS signals from the noisy LVTTTL environment.

2) Separate signals of the same type by at least twice the dielectric thickness (or twice the trace separation, whichever is greater) to the reference plane to reduce crosstalk.

3) The reference planes must extend at least five times the dielectric thickness from either side of the trace and be unbroken.

4) Avoid changing layers on high-speed signals. On a layer change, signals should share the same reference (such as ground), connected by reference vias close to the signal vias for good current return. If a different reference plane (such as Vcc) must be used due to a signal layer change, good high-frequency 0.01  $\mu$ F ceramic capacitors must be used to connect the references together as close to the signal vias as possible to ensure good transmission line properties and current return.

5) Use of a low-jitter (100 picoseconds peak-peak maximum jitter) frequency source for REF\_CLK is important. If I\_DCLK is used instead of REF\_CLK, ensure that I\_DCLK is low in jitter and always available.

6) Keep the power decoupling capacitors as close as possible to the power pins, using at least 15 mil traces and double vias for reduced inductance where possible.

7) Distribute some large-valued capacitors around the board for low-frequency decoupling and to lower the power-supply impedance.

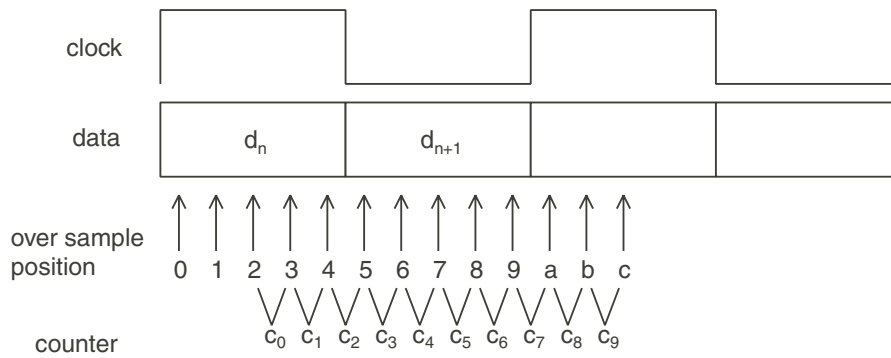
8) TRSTB (JTAG reset) must have a pull down resistor or be connected to RESETB for normal operation.

9) Filter the 1.8 Volt and 3.3 Volt analog power pins to isolate them from the noisy digital environment. Use ferrite beads and capacitors (Pi filters) for VDDA18\_x and VDDA33.

10) Suppress non-functional inner layer pads.

**8.2.7 Software Eye-Opening Check on SPI-4 Interface**

Since the SPI-4 interface is a DDR interface, both rising and falling edges are used to update or sink data.



6370 drw23b

**Figure 33. DDR interface and eye opening check through over sampling**

Refer to the IDT88P8344 uses an internal sampling clock cycle which has a frequency of 10 times SPI-4 clock to over-sample the data on a lane. For each sampling clock cycle t position n data are sampled and labeled as  $R_t.d_n$ . The following operation is then performed:

$$\begin{aligned}
 CNT_0 &= R_t.d_2 \wedge R_t.d_3 \\
 CNT_1 &= R_t.d_3 \wedge R_t.d_4 \\
 CNT_2 &= R_t.d_4 \wedge R_t.d_5 \\
 CNT_3 &= R_t.d_5 \wedge R_t.d_6 \\
 CNT_4 &= R_t.d_6 \wedge R_t.d_7 \\
 CNT_5 &= R_t.d_7 \wedge R_t.d_8 \\
 CNT_6 &= R_t.d_8 \wedge R_t.d_9 \\
 CNT_7 &= R_t.d_9 \wedge R_{t+1}.d_0 \\
 CNT_9 &= R_{t+1}.d_0 \wedge R_{t+1}.d_1
 \end{aligned}$$

For an ideal case, there is zero jitter on clock and data, zero skew, the clock high and low level phase are symmetrical. For random input data on each lane, the counters  $C_n = CNT_n(t) + CNT_n(t+1) + \dots + CNT_n(t+T)$ , where T is a time window to do the statistics computation, will increment as follows:

Counter	C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>8</sub>	C <sub>9</sub>
Value	0	0	P	0	0	0	0	K	0	0

Where P and K are non-zero and need to be a large enough value mark the transition position of a clock and define the position.

### Software for implementing the Eye-Opening Check

In the IDT88P8344, a set of diagnostic registers are provided for implementing an eye-opening check. The SPI-4 interface has 16 data lanes and one control lane on ingress, 2 status lanes on egress, making a total of 19 lanes.

The SPI-4 ingress bit alignment window register defines the window T stated above, based on which the signal statistics are computed. It is recommended to use the default value. The MEASURE\_BUSY bit indicates the status of the internal measurement operation.

The SPI-4 ingress lane measure register selects the lane statistics counters to be read, a write to this register triggers the eye-opening check process to the selected lanes and the MEASURE\_BUSY bit will be set accordingly indicating the measuring process is active. The MEASURE\_BUSY bit is cleared internally which indicates that the measuring process is complete. The measured result of counters C<sub>0</sub> through C<sub>9</sub> will be available in the SPI-4 ingress bit alignment counter registers.

Note that there is one SPI-4 ingress lane measure register and 19 SPI-4 ingress bit alignment counter registers.

The following pseudo code shows how to check the eye opening:

```
#define SPI-4_ingress_lane_measure_register 0x8801 /* register address in SPI exchange device*/
#define SPI-4_ingress_bit_alignment_counter_register(0) 0x8802
#define SPI-4_ingress_bit_alignment_counter_register(1) 0x8803
#define SPI-4_ingress_bit_alignment_counter_register(2) 0x8804
#define SPI-4_ingress_bit_alignment_counter_register(3) 0x8805
#define SPI-4_ingress_bit_alignment_counter_register(4) 0x8806
#define SPI-4_ingress_bit_alignment_counter_register(5) 0x8807
#define SPI-4_ingress_bit_alignment_counter_register(6) 0x8808
#define SPI-4_ingress_bit_alignment_counter_register(7) 0x8809
#define SPI-4_ingress_bit_alignment_counter_register(8) 0x880a
#define SPI-4_ingress_bit_alignment_counter_register(9) 0x880b

For lane=0 to K step 1 /*the number K depend on status mode: K=18 in LVDS status mode*/
{ /* K=16 otherwise */
    write #lane, SPI-4_ingress_lane_measure_register
    wait until BUSY=0 /* BUSY: bit 8 of SPI-4_ingress_lane_measure_register, at address
    0x8801*/
    for i=0 to 9 step 1
    {
        read C(i), SPI-4_ingress_bit_alignment_counter_register(i)
    }

    print C(0), C(1), C(2), C(3), C(4), C(5), C(6), C(7), C(8), C(9)
}
}
```

Because the SPI-4 ingress bit alignment counter register has a 10-bit width, the maximum counter value is 0x3ff.

If the counter values of a lane are:

0, 0, 0x3ff, 0, 0, 0, 0, 0x3ff, 0, 0

The eye open is perfect; there is very good signal integrity on input signals of the SPI-4 interface.

In each sample position represents a "tap". Depending on the delay in a lane, even a small jitter value of 1ps on the lane or the clock, may cause eye closing that can be detected by observing the counter values. In an ideal case with zero

delay, if the jitter on a data lane or clock is less than one tap interval (peak to peak), the jitter will not be reflected in counters. While the eye open check can indicate excessive jitter there are limitations in providing an accurate measurement using this method.

Theoretically as long as one tap accumulates enough non-zero samples for each 2 bits within a clock cycle the sampled signal position will be correct and the interface will function correctly. The more counters that have zero values, the better the eye opening.

## 9. REGISTER DESCRIPTION

There are two distinctly different types of register access in the IDT88P8344. Direct access registers are used for interrupts and other high-priority registers and for access to the indirect access registers. Direct access registers can be accessed more quickly than indirect access registers, and are used where this access speed advantage is required. There are only a limited number of direct access registers due to the six address lines used on the IDT88P8344. All direct access registers are one byte wide. Most registers within the IDT88P8344 are of the indirect access type. Indirect access registers are used for configuration, maps, etc., that may not need to be accessed as often as the direct registers.

Indirect registers are accessed through special direct access registers designed for the purpose of allowing indirect access to the large set of registers and maps that are needed to configure the IDT88P8344.

### 9.1 Register access summary

The SPI Exchange device uses an indirect addressing scheme for most of the configuration registers. The indirect registers are accessed through a protocol where interface pins A[5:0], D[7:0], and control pins are mapped into internal register space A[15:0], D[31:0], and Control[7:0]. The full address of any indirectly addressed register = Module\_base + Block\_base + Register\_offset.

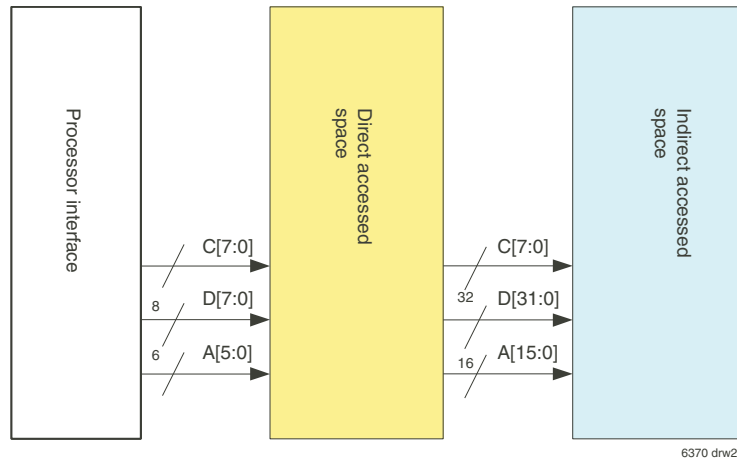


Figure 34. Direct & indirect access

#### 9.1.1 Direct register format

All direct register accesses are one byte. The bit ordering for the direct access registers is shown.

**TABLE 17 - BIT ORDER WITHIN AN 8-BIT DATA REGISTER**

<b>Direct Register Format</b>
bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

Bit 7 is the most significant data bit.

#### 9.1.2 Indirect register format

The internal format for 32 and 8 bit registers is shown below. The registers are accessed from the external processor interface as successive bytes of indirect data. The indirect register space includes 32-bit data registers and 8-bit data registers. The directly-addressed register space includes directly-addressable 8-bit data registers, four 8-bit data registers for indirect data access, two 8-bit address registers for indirect data access, and an 8-bit control register for indirect data access.

**TABLE 18 - BIT ORDER WITHIN A 32-BIT DATA REGISTER**

<b>Indirect Data (register 0x33)</b>	<b>Indirect Data (register 0x32)</b>	<b>Indirect Data (register 0x31)</b>	<b>Indirect Data (register 0x30)</b>
bit 31...bit 25	bit 24...bit 16	bit 15...bit 8	bit 7...bit 0

Bit 31 is the most significant data bit.

**TABLE 19 - BIT ORDER WITHIN AN 8-BIT DATA REGISTER**

<b>Indirect Data (register 0x30)</b>
bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

Bit 7 is the most significant data bit.

**TABLE 20 - BIT ORDER WITHIN A 16-BIT ADDRESS REGISTER**

Indirect High Address (register 0x35)	Indirect Low Address (register 0x34)
bit 15 bit 14 bit 13 bit 12 bit 11 bit 10 bit 9 bit 8	bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

Bit 15 is the most significant data bit.

**TABLE 21 - BIT ORDER WITHIN AN 8-BIT CONTROL REGISTER**

Indirect Control (register 0x3F)
bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

Bit 7 is the most significant data bit.

**Module base address (Module\_base)**

There are five modules defined for indirect data access.

**TABLE 22 - MODULE BASE ADDRESS (MODULE\_BASE)**

MODULE	Module_base
Module A (SPI3-A, PFP-A, PMON-A)	0x0000
Module B (SPI3-B, PFP-B, PMON-B)	0x2000
Module C (SPI3-C, PFP-C, PMON-C)	0x4000
Module D (SPI3-D, PFP-D, PMON-D)	0x6000
Common (SPI-4, timing, PMON, clock, GPIO, and version number)	0x8000

**Block base**

There are block bases defined for SPI-3 modules and also for Common, as shown in the following tables.

**TABLE 23 - INDIRECT ACCESS BLOCK BASES FOR MODULE A, MODULE B, MODULE C, AND MODULE D**

Block_base	Function
0x0000	SPI-3 ingress LP to LID registers
0x0200	SPI-3 ingress general configuration register
0x0500	SPI-3 egress LID to LP registers
0x0700	SPI-3 egress configuration registers
0x0A00	LID associated event counters
0x0C00	Non LID associated event counters
0x1000	SPI-3 ingress packet length configuration register
0x1100	SPI-4 egress port descriptor table
0x1200	SPI-3 ingress port descriptor tables
0x1300	SPI-3 to SPI-4 PFP and flow control registers
0x1600	SPI-4 ingress control registers
0x1700	SPI-3 egress port descriptor table
0x1800	SPI-4 ingress port descriptor table
0x1900	SPI-4 to SPI-3 PFP register

**TABLE 24 - INDIRECT ACCESS BLOCK BASES FOR COMMON MODULE**

Block_base	Function
0x0000	SPI-4 ingress LP to LID tables
0x0100	SPI-4 ingress calendar_0
0x0200	SPI-4 ingress calendar_1
0x0300	SPI-4 ingress registers
0x0400	SPI-4 egress LID to LP map
0x0500	SPI-4 egress calendar_0
0x0600	SPI-4 egress calendar_1
0x0700	SPI-4 egress configuration and status registers
0x0800	SPI-4 ingress timing block registers
0x0900	PMON timebase control, clock generator control, GPIO register, and version number

### Register offset

The register offset is shown in the section where the register is defined. The register offset is referred to as, "Register\_offset", in this document. A register reference takes the form of, "[Register\_offset 0xHH]", where HH is the hexadecimal value of the register offset.

### Indirect register access

#### Indirect register write access

An indirect write access is initiated by first checking for IND\_BUSY=0 in the indirect access control register, and then writing data into the indirect access data registers. Next, the address is written into the indirect access address registers. Then, 0x00 is written into the indirect access control register. The status of the IND\_BUSY flag in the indirect access control register is checked to ensure the process has completed before another indirect access can be initiated.

### Indirect register read access

An indirect read access is initiated by first checking for IND\_BUSY=0 in the indirect access control register, and then writing the address into the indirect access address registers. Then, 0x40 is written into the indirect access control register. The status of the IND\_BUSY flag in the indirect access control register is checked to ensure the process has completed, and then data is read out from the indirect access data registers.

The registers for controlling indirect register access are shown below. The registers for controlling indirect register access are directly accessible with read and write access.

**TABLE 25 - INDIRECT ACCESS DATA REGISTERS (DIRECT ACCESSED SPACE) AT 0x30 to 0x33**

Field	Bits	Length	Function
DATA[7:0]	7:0	8	Indirect Data Register 0x30
DATA[15:8]	15:8	8	Indirect Data Register 0x31
DATA[23:16]	23:16	8	Indirect Data Register 0x32
DATA[31:24]	31:24	8	Indirect Data Register 0x33

**TABLE 26 - INDIRECT ACCESS ADDRESS REGISTER (DIRECT ACCESSED SPACE) AT 0x34 to 0x35**

Field	Bits	Length	Function
ADDRESS[7:0]	7:0	8	Indirect Low Address Register 0x34
ADDRESS[15:8]	15:8	8	Indirect High Address Register 0x35

**TABLE 27 - INDIRECT ACCESS CONTROL REGISTER (DIRECT ACCESSED SPACE) AT 0x3F**

Field	Bits	Length
ERROR code	5:0	6
R/WN	6	1
IND_BUSY	7	1

The fields for this register are defined below.



**ERROR code** Error code: see Error coding table. This error code pertains to the last indirect access attempted.

**TABLE 28 - ERROR CODING TABLE**

ERROR code	Error Meaning
0x00	Normal indirect access completion
0x01	Multiple LP to same LID attempted assignment
0x02	Multiple LID to same LP attempted assignment
0x03	Buffer segment overflow
0x04	Not enough Queue entries
0x05	Attempt to modify while active
0x06	Address out of bound
0x07	Calibration before the chip has finished reset
0x08	LP limited
0x09	Undefined direction
0x3E	Undefined address
0x3F	Time out

**R/WN** This bit defines the read or write access to the indirect register.

0=WRITE

1=READ

**IND\_BUSY** This bit is an indication of the ability of the indirect register access to accept a new transaction, and of the completion of the current transaction.

0=Ready for read or write operation

1=Busy indication. Wait before beginning a read or write operation

## 9.2 Direct access registers

The direct access registers are in the directly-addressed access space. Direct access registers are more quickly accessed, and serve the needs of interrupts and the indirect register access.

**TABLE 29 - DIRECT MAPPED MODULE A, MODULE B, MODULE C, AND MODULE D REGISTERS**

Module A-D Direct registers	Module A	Module B	Module C	Module D
SPI-3 data capture control register	0x00	0x08	0x10	0x18
SPI-3 data capture register	0x01	0x09	0x11	0x19
SPI-4 data insert control register	0x02	0x0A	0x12	0x1A
SPI-4 data insert register	0x03	0x0B	0x13	0x1B
SPI-4 data capture control register	0x04	0x0C	0x14	0x1C
SPI-3 data insert control register	0x05	0x0D	0x15	0x1D
SPI-4 data capture register	0x06	0x0E	0x16	0x1E
SPI-3 data insert register	0x07	0x0F	0x17	0x1F
Module status register	0x24	0x25	0x26	0x27
Module enable register	0x28	0x29	0x2A	0x2B

**TABLE 30 - DIRECT MAPPED OTHER REGISTERS**

Other Direct registers	Address
Software reset	0x20
SPI-4 status register	0x22
SPI-4 enable register	0x23
Primary interrupt status register	0x2C
Secondary interrupt status register	0x2D
Primary interrupt enable register	0x2E
Secondary interrupt enable register	0x2F
Indirect access data[7:0]	0x30
Indirect access data[15:8]	0x31
Indirect access data[23:16]	0x32
Indirect access data[31:24]	0x33
Indirect access address[7:0]	0x34
Indirect access address[15:8]	0x35
Indirect access control[7:0]	0x3F

### *SPI-3 data capture control register*

**TABLE 31 - SPI-3 DATA CAPTURE CONTROL REGISTER (REGISTERS 0x00, 0x08, 0x10, 0x18)**

Field	Bits	Length	Initial Value
DATA_AVAILABLE	0	1	0b0
Reserved	7:1	7	0x00

The SPI-3 data capture control register has read and write access in the direct register access space. Write a zero to DATA\_AVAILABLE to clear the transfer. The microprocessor uses these registers to capture data from a SPI-3 ingress.

The SPI-3 data capture control register for SPI-3 port A is register 0x00.  
 The SPI-3 data capture control register for SPI-3 port B is register 0x08.  
 The SPI-3 data capture control register for SPI-3 port C is register 0x10.  
 The SPI-3 data capture control register for SPI-3 port D is register 0x18.  
 The bit field of the SPI-3 data capture control register is described.

**DATA\_AVAILABLE** The SPI-3 capture data buffer is full and ready for reading.

### *SPI-3 data capture register*

**TABLE 32 - SPI-3 DATA CAPTURE REGISTER (REGISTERS 0x01, 0x09, 0x11, 0x19)**

Field	Bits	Length	Initial Value
DATA	7:0	8	0x00

The SPI-3 data capture register has read-only access in the direct register access space. The microprocessor uses these registers to capture data from a SPI-3 ingress.

The SPI-3 data capture register for SPI-3 port A is register 0x01.  
 The SPI-3 data capture register for SPI-3 port B is register 0x09.  
 The SPI-3 data capture register for SPI-3 port C is register 0x11.  
 The SPI-3 data capture register for SPI-3 port D is register 0x19.  
 The bit field of the SPI-3 data capture register is described.

**DATA** The SPI-3 capture data buffer is read from this field.

**SPI-4 data insert control register****TABLE 33 - SPI-4 DATA INSERT CONTROL REGISTER (REGISTERS 0x02, 0x0A, 0x12, 0x1A)**

Field	Bits	Length	Initial Value
DATA_AVAILABLE	0	1	0b0
Reserved	7:1	7	0x00

The SPI-4 data insert control register has read and write access in the direct register access space. The microprocessor uses these registers to insert data into a SPI-3 ingress.

- The SPI-4 data insert control register for SPI-3 port A is register 0x02.
- The SPI-4 data insert control register for SPI-3 port B is register 0x0A.
- The SPI-4 data insert control register for SPI-3 port C is register 0x12.
- The SPI-4 data insert control register for SPI-3 port D is register 0x1A.
- The bit field of the SPI-4 data insert control register is described.

**DATA\_AVAILABLE** The SPI-4 insert data buffer is empty and ready for writing.

**SPI-4 data insert register****TABLE 34 - SPI-4 DATA INSERT REGISTER (REGISTERS 0x03, 0x0B, 0x13, 0x1B)**

Field	Bits	Length	Initial Value
DATA	7:0	8	0x00

The SPI-4 data insert register has write-only access in the direct register access space. The microprocessor uses these registers to insert data into a SPI-3 ingress.

- The SPI-4 data insert register for SPI-3 port A is register 0x03.
- The SPI-4 data insert register for SPI-3 port B is register 0x0B.
- The SPI-4 data insert register for SPI-3 port C is register 0x13.
- The SPI-4 data insert register for SPI-3 port D is register 0x1B.
- The bit field of the SPI-4 data insert register is described.

**DATA** The SPI-4 insert data buffer is written to this field.

**SPI-4 data capture control register****TABLE 35 - SPI-4 DATA CAPTURE CONTROL REGISTER (REGISTERS 0x04, 0x0C, 0x14, 0x1C)**

Field	Bits	Length	Initial Value
DATA_AVAILABLE	0	1	0b0
Reserved	7:1	7	0x00

The SPI-4 data capture control register has read and write access in the direct register access space. The microprocessor uses these registers to capture data from a SPI-3 ingress.

- The SPI-4 data capture control register for SPI-3 port A is register 0x04.
- The SPI-4 data capture control register for SPI-3 port B is register 0x0C.
- The SPI-4 data capture control register for SPI-3 port C is register 0x14.
- The SPI-4 data capture control register for SPI-3 port D is register 0x1C.
- The bit field of the SPI-4 data capture control register is described.

**DATA\_AVAILABLE** The SPI-4 capture data buffer is full and ready for reading.

**SPI-3 data insert control register****TABLE 36 - SPI-3 DATA INSERT CONTROL REGISTER (REGISTERS 0x05, 0x0D, 0x15, 0x1D)**

Field	Bits	Length	Initial Value
DATA_AVAILABLE	0	1	0b0
Reserved	7:1	7	0x00

The SPI-3 data insert control register has read and write access in the direct register access space. The microprocessor uses these registers to insert data into a SPI-4 ingress.

- The SPI-3 data insert control register for SPI-3 port A is register 0x05.
- The SPI-3 data insert control register for SPI-3 port B is register 0x0D.
- The SPI-3 data insert control register for SPI-3 port C is register 0x15.
- The SPI-3 data insert control register for SPI-3 port D is register 0x1D.
- The bit field of the SPI-3 data insert control register is described.

**DATA\_AVAILABLE** The SPI-3 insert data buffer is empty and available for writing.

**SPI-4 data capture register****TABLE 37 - SPI-4 DATA CAPTURE REGISTER (REGISTERS 0x06, 0x0E, 0x16, 0x1E)**

Field	Bits	Length	Initial Value
DATA	7:0	8	0x00

The SPI-4 data capture register has read-only access in the direct register access space. The microprocessor uses these registers to capture data from a SPI-4 ingress.

- The SPI-4 data capture register for SPI-3 port A is register 0x06.
- The SPI-4 data capture register for SPI-3 port B is register 0x0E.
- The SPI-4 data capture register for SPI-3 port C is register 0x16.
- The SPI-4 data capture register for SPI-3 port D is register 0x1E.
- The bit field of the SPI-4 data capture register is described.

**DATA** The SPI-4 capture data buffer is read from this field.

**SPI-3 data insert register****TABLE 38 - SPI-3 DATA INSERT REGISTER (REGISTERS 0x07, 0x0F, 0x17, 0x1F)**

Field	Bits	Length	Initial Value
DATA	7:0	8	0x00

The SPI-3 data insert register has write-only access in the direct register access space. The microprocessor uses these registers to insert data into a SPI-4 ingress.

- The SPI-3 data insert register for SPI-3 port A is register 0x07.
- The SPI-3 data insert register for SPI-3 port B is register 0x0F.
- The SPI-3 data insert register for SPI-3 port C is register 0x17.
- The SPI-3 data insert register for SPI-3 port D is register 0x1F.
- The bit field of the SPI-3 data insert register is described.

**DATA** The SPI-3 insert data buffer is written from this field.

## Software reset

**TABLE 39 - SOFTWARE RESET REGISTER (0x20 in the direct accessed space)**

Field	Bits	Length	Initial Value
SW_RESET	0	1	0
INIT_DONE	1	1	0
Reserved	7:2	6	0

The software reset bit is writable from the direct accessed memory space. Write a "1" to the SW\_RESET bit to initiate the software reset. The SW\_RESET bit will clear to a "0" after the chip has initialized itself. The INIT\_DONE bit is set to a "1" when the initialization following reset has completed. The software reset is the same as the hardware. The Reserved field must be set to 0.

The Initial Value column in this document is the value of the register after reset has completed.

**SW\_RESET** Setting the SW\_RESET bit initiates a software reset of the chip. The SW\_RESET bit is self-clearing.  
 0=No operation is performed  
 1=Initiate a software reset

**INIT\_DONE** Status indication bit following a reset.  
 0=Chip has not completed initialization following reset  
 1=Chip has completed initialization following reset

## SPI-4 status register (0x22 in the direct accessed space)

**TABLE 40 - SPI-4 STATUS REGISTER (0x22 IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Val
I_DIP_ERR_I	0	1	0
I_SYNCH_I	1	1	0
I_BUS_ERR_I	2	1	0
SPI4_INACTIVE_TRANSFER_I	3	1	0
DCLK_UN_I	4	1	0
E_DIP_ERR_I	5	1	0
E_SYNCH_I	6	1	0
SCLK_UN_I	7	1	0

The SPI-4 Status Register (0x22 in the direct accessed space) has read access, and interrupt status fields are cleared by a microprocessor write cycle, where a logical one must be written to clear the field(s) targeted.

The SPI-4 Status Register is a secondary interrupt status register and can only be active if the SPI-4 field is active in the Primary Interrupt Status Register (Direct 0x2C).

**I\_DIP\_ERR\_I** SPI-4 ingress DIP-4 error interrupt indication.  
 0=No errors  
 1=One or more DIP-4 errors have been registered on the SPI-4 ingress

**I\_SYNCH\_I** SPI-4 ingress data path has transitioned from out of synchronization to in synchronization condition interrupt indication.  
 0=No detection, still not in synchronization

1=Transition from out of synchronization to in synchronization, or transition from in synchronization to out of synchronization

**I\_BUS\_ERR\_I** SPI-4 ingress bus error interrupt indication.  
 0=No errors  
 1=One or more bus errors have been registered on the SPI-4 ingress

**SPI4\_INACTIVE\_TRANSFER\_I** SPI-4 ingress inactive transfer interrupt indication.  
 0=No indication  
 1=One or more inactive transfers have been registered on the SPI-4 ingress

**DCLK\_UN\_I** SPI-4 ingress data clock has transitioned from available to an unavailable condition interrupt indication.  
 0=No detection, I\_DCLK is available  
 1=I\_DCLK transitioned from available to an unavailable state

**E\_DIP\_ERR\_I** SPI-4 egress DIP-2 error interrupt indication on the SPI-4 egress status channel.  
 0=No errors  
 1=One or more DIP-2 errors have been registered

**E\_SYNCH\_I** SPI-4 egress status channel has transitioned from out of synchronization to an in synchronization condition interrupt indication.  
 0=No detection, still not in synchronization  
 1=Transition from out of synchronization to in synchronization, or transition from in synchronization to out of synchronization

**SCLK\_UN\_I** SPI-4 egress status clock has transitioned from available to an unavailable condition interrupt indication.

In LVTTTL mode the Bridgeport does not detect the SPI-4 egress status clock (E\_SCLK\_T). Therefore, for LVTTTL mode the software should ignore the SCLK\_UN field in the SPI-4 Status Register.

0=No detection, E\_SCLK is available  
 1=E\_SCLK transitioned from available to an unavailable state

## SPI-4 enable register (0x23 in the direct accessed space)

**TABLE 41 - SPI-4 ENABLE REGISTER (0x23 IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Val
I_DIP4_ERR_EN	0	1	0
I_SYNCH_EN	1	1	0
I_BUS_ERR_EN	2	1	0
SPI4_INACTIVE_TRANSFER_EN	3	1	0
DCLK_UN_EN	4	1	0
E_DIP_ERR_EN	5	1	0
E_SYNCH_EN	6	1	0
SCLK_UN_EN	7	1	0

The SPI-4 Enable Register (0x23 in the direct accessed space) has read and write access. SPI-4 Enable Register is used to bitwise enable the interrupts in the SPI-4 Status Register.

**I\_DIP4\_ERR\_EN** SPI-4 ingress DIP-4 error interrupt indication enable.  
 0=Disable DIP-4 error interrupt  
 1=Enable DIP-4 error interrupt

**I\_SYNCH\_EN** SPI-4 ingress data path has transitioned from out of synchronization to in synchronization condition interrupt indication enable.  
 0=Disable synchronization interrupt  
 1=Enable synchronization interrupt

**I\_BUS\_ERR\_EN** SPI-4 ingress bus error interrupt indication enable.  
 0=Disable bus error interrupt  
 1=Enable bus error interrupt

**SPI4\_INACTIVE\_TRANSFER\_EN** SPI-4 ingress inactive transfer interrupt indication enable.  
 0=Disable inactive transfer interrupt  
 1=Enable inactive transfer interrupt

**DCLK\_UN\_EN** SPI-4 ingress data clock has transitioned from available to an unavailable condition interrupt indication enable.  
 0=Disable unavailable interrupt  
 1=Enable unavailable interrupt

**E\_DIP\_ERR\_EN** SPI-4 egress DIP-2 error interrupt indication enable on the SPI-4 egress status channel.  
 0=Disable DIP-2 error interrupt  
 1=Enable DIP-2 error interrupt

**E\_SYNCH\_EN** SPI-4 egress status channel has transitioned from out of synchronization to in synchronization condition interrupt indication enable.  
 0=Disable synchronization interrupt  
 1=Enable synchronization interrupt

**SCLK\_UN\_EN** SPI-4 egress status clock has transitioned from available to an unavailable condition interrupt indication enable. SCLK\_UN\_EN should be written as a zero if using a SPI-4 egress LVTTTL status clock that is less than one-half of the MCLK frequency. The SCLK\_UN\_I interrupt indication is not usable in this case.  
 0=Disable unavailable interrupt  
 1=Enable unavailable interrupt

**Module status register (0x24 to 0x27 in the direct accessed space)**

**TABLE 42 - MODULE STATUS REGISTER (0x24 TO 0x27 IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
SPI-34_CAPTURE	0	1	0
SPI-43_CAPTURE	1	1	0
SPI-34_INSERT	2	1	0
SPI-43_INSERT	3	1	0
PMON	4	1	0
Reserved	7:5	3	0

The Module Status registers (0x24 to 0x27 in the direct accessed space) have read and write access, and interrupt status fields are cleared by a microprocessor write cycle, where a logical one must be written to clear the field(s) targeted, except for the PMON field, which can not be cleared.

The Module Status registers are secondary interrupt status registers and can only be active if the corresponding field is active in the Primary Interrupt Status Register (Direct 0x2C):

Module Status Register 0x24 is for module A, and can only be active if the MODULE\_A field is active in the Primary Interrupt Status Register.

Module Status Register 0x25 is for module B, and can only be active if the MODULE\_B field is active in the Primary Interrupt Status Register.

Module Status Register 0x26 is for module C, and can only be active if the MODULE\_C field is active in the Primary Interrupt Status Register.

Module Status Register 0x27 is for module D, and can only be active if the MODULE\_D field is active in the Primary Interrupt Status Register.

**SPI-34\_CAPTURE** SPI-3 ingress to SPI-4 egress capture event interrupt indication.  
 0=No capture event  
 1=Buffer is ready for reading by the microprocessor

**SPI-43\_CAPTURE** SPI-4 ingress to SPI-3 egress capture event interrupt indication.  
 0=No capture event  
 1=Buffer is ready for reading by the microprocessor

**SPI-34\_INSERT** SPI-3 ingress to SPI-4 egress insert event interrupt indication.  
 0=No insert event  
 1=Buffer is ready for writing by the microprocessor

**SPI-43\_INSERT** SPI-4 ingress to SPI-3 egress insert event interrupt indication.  
 0=No insert event  
 1=Buffer is ready for writing by the microprocessor

**PMON** Performance Monitor event interrupt indication. Writing to this field has no effect.  
 0=No PMON event  
 1=PMON event is ready for reading by the microprocessor

**Module enable register (0x28 to 0x2B in the direct accessed space)**

**TABLE 43 - MODULE ENABLE REGISTER (0x28 TO 0x2B IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
SPI-34_CAPTURE_EN	0	1	0
SPI-43_CAPTURE_EN	1	1	0
SPI-34_INSERT_EN	2	1	0
SPI-43_INSERT_EN	3	1	0
PMON_EN	4	1	0
Reserved	7:5	3	0

The Module Enable registers (0x28 to 0x2B in the direct accessed space) have read and write access.

Module Enable Register 0x28 is for module A.  
 Module Enable Register 0x29 is for module B.  
 Module Enable Register 0x2A is for module C.  
 Module Enable Register 0x2B is for module D.

**SPI-34\_CAPTURE\_EN** SPI-3 ingress to SPI-4 egress capture event interrupt enable.

0=Disable capture interrupt  
1=Enable capture interrupt

**SPI-43\_CAPTURE\_EN** SPI-4 ingress to SPI-3 egress capture event interrupt enable.

0=Disable capture interrupt  
1=Enable capture interrupt

**SPI-34\_INSERT\_EN** SPI-3 ingress to SPI-4 egress insert event interrupt enable.

0=Disable insert interrupt  
1=Enable insert interrupt

**SPI-43\_INSERT\_EN** SPI-4 ingress to SPI-3 egress insert event interrupt enable.

0=Disable insert interrupt  
1=Enable insert interrupt

**PMON\_EN** Performance Monitor event interrupt enable.

0=Disable PMON interrupt  
1=Enable PMON interrupt

### Primary interrupt status register (0x2C in the direct accessed space)

**TABLE 44 - PRIMARY INTERRUPT STATUS REGISTER (0x2C IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
MODULE_A	0	1	0
MODULE_B	1	1	0
MODULE_C	2	1	0
MODULE_D	3	1	0
SPI-4	4	1	0
SECONDARY	5	1	0
Reserved	7:6	2	0

The primary interrupt status register (0x2C in the direct accessed space) has read-only access. The interrupts for the primary interrupt status register must be acknowledged by servicing the corresponding secondary interrupt status registers.

**MODULE\_A** When active, the MODULE\_A field is responsible for allowing an interrupt in the Module Status Register (secondary interrupt register 0x24).

0=No MODULE\_A interrupt active  
1=MODULE\_A interrupt is active

**MODULE\_B** When active, the MODULE\_B field is responsible for allowing an interrupt in the module status register (secondary interrupt register 0x25).

0=No MODULE\_B interrupt active  
1=MODULE\_B interrupt is active

**MODULE\_C** When active, the MODULE\_C field is responsible for allowing an interrupt in the module status register (secondary interrupt register 0x26).

0=No MODULE\_C interrupt active  
1=MODULE\_C interrupt is active

**MODULE\_D** When active, the MODULE\_D field is responsible for allowing an interrupt in the module status register (secondary interrupt register 0x27).

0=No MODULE\_D interrupt active  
1=MODULE\_D interrupt is active

**SPI-4** When active, the SPI-4 field is responsible for allowing an interrupt in the SPI-4 status register (secondary interrupt register 0x22).

0=No SPI-4 interrupt active  
1=SPI-4 interrupt is active

**SECONDARY** When active, the SECONDARY field is responsible for allowing an interrupt in the Secondary Interrupt Status Register (secondary interrupt register 0x2D).

0=No SECONDARY interrupt active  
1=SECONDARY interrupt is active

### Secondary interrupt status register (0x2D in the direct accessed space)

**TABLE 45 - SECONDARY INTERRUPT STATUS REGISTER (0x2D IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
TIME_BASE	0	1	0
INDIRECT_ACCESS	1	1	0
Reserved	7:2	6	0

The secondary interrupt status register (0x2D in the direct accessed space) has read and write access.

The secondary interrupt status register has read access, and Interrupt status fields are cleared by a microprocessor write cycle, where a logical one must be written to clear the field(s) targeted.

The secondary interrupt status register is a secondary interrupt status register and can only be active if the SECONDARY\_EN field is active in the primary interrupt enable register (Direct 0x2C).

**TIME\_BASE** Time base expiration interrupt indication.

0=No time base event  
1=Time base has expired

**INDIRECT\_ACCESS** Indirect access completion interrupt indication.

0=No indirect access event  
1=Indirect access has completed

**Primary interrupt enable register (0x2E in the direct accessed space)**

**TABLE 46 - PRIMARY INTERRUPT ENABLE REGISTER (0x2E IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
MODULE_A_EN	0	1	0
MODULE_B_EN	1	1	0
MODULE_C_EN	2	1	0
MODULE_D_EN	3	1	0
SPI4_EN	4	1	0
SECONDARY_EN	5	1	0
Reserved	7:6	2	0

The Primary Interrupt Enable Register (0x2E in the direct accessed space) has read and write access.

The Primary Interrupt Enable Register is used to bitwise enable the interrupts in the Primary Interrupt Status Register.

**MODULE\_A\_EN** MODULE\_A interrupt enable.  
0=Disable MODULE\_A interrupt  
1=Enable MODULE\_A interrupt

**MODULE\_B\_EN** MODULE\_B interrupt enable.  
0=Disable MODULE\_B interrupt  
1=Enable MODULE\_B interrupt

**MODULE\_C\_EN** MODULE\_C interrupt enable.  
0=Disable MODULE\_C interrupt  
1=Enable MODULE\_C interrupt

**MODULE\_D\_EN** MODULE\_D interrupt enable.  
0=Disable MODULE\_D interrupt  
1=Enable MODULE\_D interrupt

**SPI4\_EN** SPI-4 interrupt enable.  
0=Disable SPI-4 interrupt  
1=Enable SPI-4 interrupt

**SECONDARY\_EN** SECONDARY interrupt enable.  
0=Disable SECONDARY interrupt  
1=Enable SECONDARY interrupt

**Secondary interrupt enable register (0x2F in the direct accessed space)**

**TABLE 47 - SECONDARY INTERRUPT ENABLE REGISTER (0x2F IN THE DIRECT ACCESSED SPACE)**

Field	Bits	Length	Initial Value
TIME_BASE_EN	0	1	0
INDIRECT_ACCESS_EN	1	1	0
Reserved	7:2	6	0

The secondary interrupt enable register (0x2F in the direct accessed space) has read and write access.

The secondary interrupt enable register is used to bitwise enable the interrupts in the secondary interrupt enable register.

**TIME\_BASE\_EN** Time base expiration interrupt enable.  
0=Disable time base event interrupt  
1=Enable time base event interrupt

**INDIRECT\_ACCESS\_EN** Indirect access completion interrupt enable.  
0=Disable indirect access completion event interrupt  
1=Enable indirect access completion event interrupt

### 9.3 Indirect registers for SPI-3A, SPI-3B, SPI-3C, SPI-3D modules

*Module A is at Module\_base 0x0000*

*Module B is at Module\_base 0x2000*

*Module C is at Module\_base 0x4000*

*Module D is at Module\_base 0x6000*

**TABLE 48 - MODULE A/B/C/D INDIRECT REGISTER**

Table Number, Page	Block_base, Register_offset	Title of Register
49, page 57	0x0000, 0x00-0xFF	SPI-3 ingress LP to LID map
50, page 57	0x0200, 0x00	SPI-3 general configuration register
51, page 58	0x0200, 0x01	SPI-3 ingress configuration register
52, page 58	0x0200, 0x02	SPI-3 ingress fill level register
53, page 58	0x0200, 0x03	SPI-3 ingress max fill level register
54, page 58	0x0500, 0x00-0x3F	SPI-3 egress LID to LP map
55, page 59	0x0700, 0x00	SPI-3 egress configuration register
56, page 59	0x0700, 0x01	SPI-4 ingress to SPI-4 egress flow control register
57, page 59	0x0700, 0x02	SPI-3 egress test register
58, page 60	0x0700, 0x03	SPI-3 egress fill level register
59, page 60	0x0700, 0x04	SPI-3 egress max fill level register
60, page 61	0x0A00, 0x00-0x17F	LID associated event counters
61, page 61	0x0C00, 0x00-0x0B	Non LID associated event counters
62, page 62	0x0C00, 0x0C	Non LID associated interrupt indication register
63, page 62	0x0C00, 0x0D	Non LID associated interrupt enable register
64, page 62	0x0C00, 0x0E	LID associated interrupt indication register
65, page 62	0x0C00, 0x0F	LID associated interrupt enable register
66, page 63	0x0C00, 0x10-0x15	Non critical LID associated capture table
67, page 63	0x0C00, 0x16-0x17	SPI-3 to SPI-4 critical LID interrupt indication registers
68, page 63	0x0C00, 0x18-0x19	SPI-3 to SPI-4 critical LID interrupt enable registers
69, page 63	0x0C00, 0x1A-0x1B	SPI-4 to SPI-3 critical LID interrupt indication registers
70, page 63	0x0C00, 0x1C-0x1D	SPI-4 to SPI-3 critical LID interrupt enable registers
71, page 63	0x0C00, 0x1E	Critical events source indication register
72, page 64	0x1000, 0x00-0x3F	SPI-3 ingress packet length configuration register
73, page 64	0x1100, 0x00-0x3F	SPI-4 egress port descriptor table
74, page 64	—————	SPI-4 egress direction code assignment
75, page 64	0x1200, 0x00-0x3F	SPI-3 ingress port descriptor table
76, page 65	0x1300, 0x00	SPI-3 to SPI-4 PFP register
77, page 65	—————	NR_LID field encoding
78, page 65	0x1300, 0x01	SPI-3 to SPI-4 flow control register
79, page 66	0x1600, 0x00-0x3F	SPI-4 ingress packet length register
80, page 66	0x1700, 0x00-0x3F	SPI-3 egress port descriptor table
81, page 66	—————	SPI-3 egress direction code assignment
82, page 66	0x1800, 0x00-0x3F	SPI-4 ingress port descriptor table
83, page 67	0x1900, 0x00	SPI-4 to SPI-3 PFP register
84, page 67	—————	NR_LID field encoding



### 9.3.1 Block base 0x0000 registers

**SPI-3 ingress LP to LID map (Block\_base 0x0000 + Register\_offset 0x00 to 0xFF)**

**TABLE 49 - SPI-3 INGRESS LP TO LID MAP**

Field	Bits	Length	Initial Value
LID	5:0	6	0x00
ENABLE	6	1	0b0
BIT_REVERSAL	7	1	0b0

There are 256 SPI-3 ingress Logical Port (LP) to Logical Identifier (LID) registers, one per potential SPI-3 LP. Only 64 LPs per SPI-3 physical interface can be enabled. An attempt to enable more than 64 LPs per SPI-3 physical interface or to assign an identical LID to more than one LP will be discarded and an error code will be returned. The ENABLE bit is used to enable SPI-3 logical ports. All data from non-enabled SPI-logical ports is discarded and an inactive SPI-3 logical port event is generated. This event is directed towards the PMON & DIAG module. Disabled ports always generate available status.

The Table 49 - SPI-3 ingress LP to LID Map assigns a LID to a SPI-3 logical port. LID mapping for 64 out of 256 SPI-3 logical ports is supported on each SPI-3 physical port. LPs in the SPI Exchange are 8 bits wide[7:0] and range from 0 to 255. An example of mapping SPI-3 physical interface "A", LP 0x08 to LID 0x05, activating the LID, and not using bit reversal is outlined.

Perform an indirect write of 0x45 to register address Module\_base 0x0000 + Block\_base 0x0000 + Register\_offset 0x08 = 0x0008. Another example of SPI-3 LP to LID mapping is to map SPI-3 "B", LP 0xFF to LID 0x3F, write 0x7F to register Module\_base 0x2000 + Block\_base 0x0000 + Register\_offset 0xFF = 0x20FF.

The Initial Value column is the value of the register after reset.

**LID** The LID programmed is associated to the LP with the same number as the register address. Six bits support the 64 simultaneously active LIDs per SPI-3 physical interface.

**ENABLE** This bit is used to enable or disable the connection of this LP to this LID.

0=LID disabled  
1=LID enabled

**BIT\_REVERSAL** This bit is used to reverse the bit ordering of each byte of the SPI-3 interface on a per-LID basis.

0=Disable bit reversal for this LID  
1=Enable bit reversal for this LID

### 9.3.2 Block base 0x0200 registers

**SPI-3 general configuration register (Block\_base 0x0200 + Register\_offset 0x00)**

**TABLE 50 - SPI-3 GENERAL CONFIGURATION REGISTER (REGISTER\_OFFSET=0x00)**

Field	Bits	Length	Initial Value
LINK	0	1	0b0
PACKET	1	1	0b1
SPI3_ENABLE	2	1	0b0
BUSWIDTH	3	1	0b0
EVEN_PARITY	4	1	0b0
PARITY_EN	5	1	0b1
Reserved	6	1	0b0
Reserved	7	1	0b0
WATERMARK	12:8	5	0x0F
Reserved	31:9	19	0x000

There is one register for SPI-3 general configuration per SPI-3 interface. Each register has read and write access. The address for module A is 0x0200, module B is 0x2200, module C is 0x4200, and module D is 0x6200. The bit fields of the SPI-3 general configuration register are described in the following paragraphs.

**LINK** A SPI-3 interface can be used either in Link or PHY modes. For connecting to a transmission line-interface PHY, program the SPI Exchange for Link mode. For connecting the SPI-3 interface to an NPU or other Link-mode device, program the SPI-3 interface for PHY mode. Note that the four SPI-3 interfaces can be independently configured into either Link or PHY modes. The SPI-3 ingress and egress of a given SPI-3 physical port will always be in the same mode.

0=SPI-3 interface in PHY mode  
1=SPI-3 interface in Link mode

**PACKET** A SPI-3 interface can be used either in BYTE or PACKET modes. A SPI-3 interface acting as a Link layer device can poll the attached PHY device for up to 64 LPs if the attached PHY device supports the polling interface. When attached to a PHY device that only supports byte mode, the four direct status indicators can be used. Note that the four SPI-3 interfaces can be independently configured into either BYTE or PACKET modes. When the SPI Exchange is in PHY mode, the PACKET bit is used to select either a polled or direct status response to the attached Link device.

0 = BYTE mode with direct status indication for up to 4 LPs [3:0]  
1 = PACKET mode with polled status for up to 64 LPs

**SPI3\_ENABLE** A SPI-3 interface can be enabled or disabled according to the state programmed into this bit. A port should be disabled to save power if it is not used.

0=SPI-3 Physical port disabled, outputs are in tristate  
1=SPI-3 Physical port enabled

**BUSWIDTH** A SPI-3 interface can be used as either a single 8-bit or 32-bit interface, according to the needs of the attached device. The SPI-3 ingress and egress of a given SPI-3 physical port will always be of the same bus width.

0=32 bit SPI-3 interface  
1=8 bit SPI-3 interface

**EVEN\_PARITY** A SPI-3 interface is provisioned to generate and to check for odd or even parity. The PARITY\_EN bit must be set for this to become effective. Odd parity is standard for SPI-3 interfaces.

- 0=Odd parity on this port
- 1=Even parity on this port

**PARITY\_EN** A SPI-3 interface is provisioned to enable or disable parity generation and checking, according to the state of the EVEN\_PARITY bit.

- 0=Disable parity on this SPI-3 port
- 1=Enable parity on this SPI-3 port

**WATERMARK** A SPI-3 interface can be set to a SPI-3 ingress port watermark value. 0x10 is the highest watermark that can be set, meaning all ingress buffers will be full before backpressure will be initiated on a SPI-3 ingress interface. The WATERMARK field value of 0x08 is used to set the watermark for a half-full ingress buffer before tripping backpressure. The units of WATERMARK are one-sixteenth of the available ingress buffering per unit. Each unit is equal to 128 bytes. BACKPRESSURE\_EN must be set [Register\_offset 0x01] for the watermark to become effective. The watermark field is usually set to 0x10, and the FREE\_SEGMENT field of Table 75, SPI-3 ingress port descriptor tables (Block\_base 0x1200) is used for per LID backpressure.

### **SPI-3 ingress configuration register (Block\_base 0x0200 + Register\_offset 0x01)**

**TABLE 51 - SPI-3 INGRESS CONFIGURATION REGISTER (REGISTER\_OFFSET=0x01)**

Field	Bits	Length	Initial Value
BACKPRESSURE_EN	0	1	0b1
FIX_LP	1	1	0b0
Reserved	31:2	30	0x0000

There is one register for SPI-3 ingress configuration per SPI-3 interface. Each register has read and write access.

The bit fields for a SPI-3 ingress configuration register are described in the following paragraphs.

**BACKPRESSURE\_EN** A SPI-3 interface can have backpressure enabled or disabled. Disabling backpressure means that data coming into the ingress may be lost if the SPI-3 interface ingress buffers overflow. The SPI-3 interface can run at full-rate, however, since there will be no backpressure. Attached devices that do not respond properly to backpressure should be interfaced by disabling backpressure.

Enabling backpressure will cause the I\_ENB signal to be asserted when the ingress buffer fill level is equal to the WATERMARK value [Register\_offset 0x00], or the free segment buffer threshold Table 75, SPI-3 ingress port descriptor table (Block\_base 0x1200) has been reached for any active LID.

- 0=Disable backpressure on this SPI-3 ingress.
- 1=Enable backpressure on this SPI-3 ingress interface.

**FIX\_LP** A SPI-3 interface can fix the logical port address to 0x00. This is useful when there is only one LP on an interface, such as with some single-PHY devices.

- 0=Do not fix logical port address to 0x00, but use the actual LP found in the packet fragments.
- 1= Fix logical port address to 0x00

### **SPI-3 ingress fill level register (Block\_base 0x0200 + Register\_offset 0x02)**

**TABLE 52 - SPI-3 INGRESS FILL LEVEL REGISTER (REGISTER\_OFFSET=0x02)**

Field	Bits	Length	Initial Value
FILL_CUR	4:0	5	0x00
I_FCLK_AV	5	1	0b1

There is one register for SPI-3 ingress fill level register per SPI-3 interface. Each register has read-only access. The bit fields of a SPI-3 ingress fill level register are described.

**FILL\_CUR** Current SPI-3 ingress buffer fill level. Since this is a real-time register, the value read from it will change rapidly and is used for internal diagnostics only.

- I\_FCLK\_AV** Current SPI-3 ingress clock availability is checked here.
  - 0=SPI-3 ingress clock not detected on a SPI-3 port
  - 1=SPI-3 ingress clock transitions detected on a SPI-3 port

### **SPI-3 ingress max fill register (Block\_base 0x0200 + Register\_offset 0x03)**

**TABLE 53 - SPI-3 INGRESS MAX FILL LEVEL REGISTER (REGISTER\_OFFSET=0x03)**

Field	Bits	Length	Initial Value
FILL_MAX	4:0	5	0x00

There is one register for SPI-3 ingress max fill level register per SPI-3 interface. Each register has read-only access, and is cleared after reading. 0x10 is the highest filling level, meaning all ingress buffers had been full at some time since the last read of the FILL\_MAX field. The units of FILL\_MAX are one-sixteenth of the available ingress buffering. Each unit is equal to 128 bytes. The bit field of a SPI-3 ingress max fill level register is described. The Table 53 - SPI-3 ingress max fill level register (Register\_offset=0x03) is for diagnostics only.

**FILL\_MAX** Maximum SPI-3 ingress buffer fill level since the last read of the SPI-3 ingress max fill level register.

### **9.3.3 Block base 0x0500 registers**

#### **SPI-3 egress LID to LP map (Block\_base 0x0500 + Register\_offset 0x00-0x3F)**

**TABLE 54 - SPI-3 EGRESS LID TO LP MAP**

Field	Bits	Length	Initial Value
LP	7:0	8	0x00
ENABLE	8	1	0b0
BIT_REVERSAL	9	1	0b0

There are 64 SPI-3 egress LID to LP maps per SPI-3 interface, one per potential SPI-3 LID.

The SPI-3 egress LID to LP maps have read and write access. The SPI-3 egress LID to LP maps are used to map SPI-3 egress logical identifiers to SPI-3 logical port addresses that are in-band with the SPI-3 egress packet fragments.

**LP** The LP programmed is associated to the LID with the same number as the register address.

**ENABLE** This bit is used to enable or disable the connection of a LID to an LP.

- 0=LP disabled
- 1=LP enabled

**BIT\_REVERSAL** This bit is used to reverse the bit ordering of each byte of the SPI-3 interface on a per-LP basis.

- 0=Disable bit reversal for an LP
- 1=Enable bit reversal for an LP

### 9.3.4 Block base 0x0700 registers

#### *SPI-3 egress configuration register (Block\_base 0x0700 + Register\_offset 0x00)*

**TABLE 55 - SPI-3 EGRESS CONFIGURATION REGISTER (REGISTER\_OFFSET=0x00)**

Field	Bits	Length	Initial Value
POLL_LENGTH	5:0	6	0x0F
Reserved	7:6	2	0b00
STX_SPACING	8	1	0b0
EOP_SPACING	9	1	0b0
Reserved	31:10	22	0x00

There is one SPI-3 egress configuration register per SPI-3 interface. The SPI-3 egress configuration registers have read and write access. A SPI-3 egress configuration registers is used to control the poll sequence length of a SPI-3 egress interface when the SPI-3 interface is in Link mode. The SPI-3 egress configuration register is used to add two cycles to STX or EOP as required to interface to the attached device.

**POLL\_LENGTH** Poll sequence length when in Link mode. The poll sequence is from the LP associated with LID0 to the LP associated with the LID for POLL\_LENGTH - 1.

**STX\_SPACING** This bit is used to enable or disable the adding of two dummy STX cycles to a SPI-3 egress interface to meet the needs of an attached device.

- 0= No dummy STX cycles are added to a SPI-3 egress.
- 1= Two dummy STX cycles are added to a SPI-3 egress

**EOP\_SPACING** This bit is used to enable or disable the adding of two dummy EOP cycles to a SPI-3 egress interface to meet the needs of an attached device.

- 0= No dummy EOP cycles are added to a SPI-3 egress.
- 1= Two dummy EOP cycles are added to a SPI-3 egress

#### *SPI-4 ingress to SPI-3 egress flow control register (Block\_base 0x0700 + Register\_offset 0x01)*

**TABLE 56 - SPI-4 INGRESS TO SPI-3 EGRESS FLOW CONTROL REGISTER (REGISTER\_OFFSET=0x01)**

Field	Bits	Length	Initial Value
CREDIT_EN	0	1	0b0
BURST_EN	1	1	0b0
LOOP_BACK	2	1	0b0
Reserved	31:3	29	0x00

The SPI-4 ingress to SPI-3 egress flow control register has read and write access. The bit fields of the SPI-4 ingress to SPI-3 egress flow control register are described.

**CREDIT\_EN** CREDIT\_EN The flow control information received from the attached SPI-3 device is interpreted as status or credit information as selected by the CREDIT\_EN bit in the SPI-4 ingress to SPI-3 egress flow control Register. If the status mode is used, data will be egressed until the status is changed by the attached SPI-3 device. If the credit mode is used, the SPI-3 egress will transmit only one packet fragment and then wait for an update in the internal buffer segment pool status before sending another packet fragment.

- 0=Status mode
- 1=Credit mode

**BURST\_EN** Multiple Burst Enable allows more than one burst to be sent to an LP. When this feature is not enabled, only one burst per LP is allowed into the SPI-3 egress buffers.

- 0=Disable burst enable
- 1=Enable burst enable

**LOOP\_BACK** In this mode the contents of a SPI-3 ingress are directly transferred to a SPI-3 egress buffers of the same port. This mode is useful for off-line diagnostics.

- 0=Disable loopback
- 1=Enable loopback

#### *SPI-3 egress test register (Block\_base 0x0700 + Register\_offset 0x02)*

**TABLE 57 - SPI-3 EGRESS TEST REGISTER (REGISTER\_OFFSET=0x02)**

Field	Bits	Length	Initial Value
ADD_PAR_ERR	0	1	0b0
DAT_PAR_ERR	1	1	0b0
Reserved	7:2	6	0x00
PORT_ADDRESS	15:8	8	0x0F

The SPI-3 egress test register has read and write access. A single address parity error is introduced on a SPI-3 egress LP through the ADD\_PAR\_ERR bit field. A single data parity error is introduced on a SPI-3 egress LP through the DAT\_PAR\_ERR bit field. The LP affected by these two parity error bit fields is enumerated in the PORT\_ADDRESS field. The bit fields of SPI-3 egress test register are described. The bit fields are automatically cleared following the generation of the associated error.

**ADD\_PAR\_ERR** A single address parity error is introduced on a SPI-3 egress LP through the ADD\_PAR\_ERR bit field. The LP affected by the ADD\_PAR\_ERR bit field is enumerated in the PORT\_ADDRESS field.

0=No parity error introduced

1=Introduce a single address parity error on a SPI-3 egress LP

**DAT\_PAR\_ERR** A single data parity error is introduced on a SPI-3 egress LP through the DAT\_PAR\_ERR bit field. The LP affected by the DAT\_PAR\_ERR bit field is enumerated in the PORT\_ADDRESS field.

0=No parity error introduced

1=Introduce a single data parity error on a SPI-3 egress LP

**PORT\_ADDRESS** The LP affected by both the ADD\_PAR\_ERR and the DAT\_PAR\_ERR bit fields is enumerated in the PORT\_ADDRESS field. The value of the PORT\_ADDRESS is set from 0x00 to 0xFF.

**SPI-3 egress fill level register (Block\_base 0x0700 + Register\_offset 0x03)**

**TABLE 58 - SPI-3 EGRESS MAX FILL LEVEL REGISTER (REGISTER\_OFFSET=0x03)**

Field	Bits	Length	Initial Value
FILL_CUR	3:0	4	0x0
Reserved	4	1	0x0
E_FCLK_AV	5	1	0b0

There is one register for SPI-3 egress fill level register per SPI-3 interface. Each register has read-only access. The bit fields of the SPI-3 egress fill level register are described.

**FILL\_CUR** Current SPI-3 egress buffer fill level. Since this is a real-time register, the value read from it will change rapidly and is used for internal diagnostics only.

**I\_FCLK\_AV** Current SPI-3 egress clock availability is checked here.

0=SPI-3 egress clock transitions were not detected on a SPI-3 port

1=SPI-3 egress clock transitions were detected on a SPI-3 port

**SPI-3 egress max fill level register (Block\_base 0x0700 + Register\_offset 0x04)**

**TABLE 59 - SPI-3 EGRESS MAX FILL LEVEL REGISTER (REGISTER\_OFFSET=0x04)**

Field	Bits	Length	Initial Value
FILL_MAX	3:0	4	0x0

There is one register for SPI-3 egress max fill Level per SPI-3 interface. Each register has read-only access, and is cleared after reading. 0xF is the highest filling level, meaning all egress buffers had been full at some time since the last read of the FILL\_MAX field. The units of FILL\_MAX are one-sixteenth of the available egress buffering. Each unit is equal to 128 bytes. The bit field of the SPI-3 egress max fill Level register is described.

**FILL\_MAX** Maximum SPI-3 egress buffer fill level since the last read of the SPI-3 egress Max Fill Level Register

## Performance monitor counters

Two categories of events are captured: LID and non LID associated events. If at least one event is captured in one of the interrupt indication registers, an active PMON service request is directed towards the interrupt module.

All events and diagnostics data are accumulated during an interval defined by the *time base event*. The data accumulated during the previous time period can be accessed by the indirect access scheme. The counters are cleared when the time base expires. All counters are saturating, and will not overflow.

### 9.3.5 Block base 0x0A00 registers

#### *LID associated event counters (Block\_base 0x0A00 + Register\_offset 0x000 to 0x17F)*

**TABLE 60 - LID ASSOCIATED EVENT COUNTERS (0x000-0x17F)**

Offset	Counter	Counter length (bits)
LID*6+0	SPI-3 ingress good packet counter	24
LID*6+1	SPI-3 ingress bad packet counter(error tagged)	24
LID*6+2	SPI-4 ingress good packet counter	24
LID*6+3	SPI-4 ingress bad packet counter(abort)	24
LID*6+4	SPI-3 egress packet counter	24
LID*6+5	SPI-4 egress packet counter	24

A set of event counters is provided for each of the 64 LIDs on each SPI-3 interface and for each LID to or from a SPI-4 module. LID associated event counters keep track of packets, packets not delineated by an SOP and an EOP, or error-tagged packets.

### 9.3.6 Block base 0x0C00 registers

#### *Non LID associated event counters (Block\_base 0x0C00 + Register\_offset 0x00 to 0x0B)*

**TABLE 61 - NON LID ASSOCIATED EVENT COUNTERS (0x00 - 0x0B)**

Offset	Counters	Subject to accumulation	Width
0x00	SPI-3 ingress bytes	All bytes of transfers for active SPI-3 (excluding address parity error) individual for A/B/C/D	29
0x01	SPI-3 ingress transfers	All transfers for active SPI-3	26
0x02	PFP3-4 ingress too long packet	SPI-3 ingress packets longer than MAX_LENGTH	8
0x03	PFP3-4 ingress too short packet	SPI-3 ingress packets shorter than MIN_LENGTH	8
0x04	SPI-3 ingress error tagged packet	Parity error on SPI-3 ingress	8
0x05	Reserved		8
0x06	SPI-4 ingress bytes	All bytes of transfers for SPI-4, individual for modules A/B/C/D	29
0x07	SPI-4 ingress transfers	All transfers for active SPI-4 for A and B and C and D, same value in A, B, C, D	26
0x08	PFP4-3 ingress too long packet	SPI-4 ingress packets longer than MAX_LENGTH	8
0x09	PFP4-3 ingress too short packet	SPI-4 ingress packets shorter than MIN_LENGTH	8
0x0A	SPI-4 ingress error tagged packet	DIP-4 error on ingress	8
0x0B	Reserved		8

Non LID Associated Event Counters are associated with SPI-3 and SPI-4 physical interfaces. The register offset is shown in the Offset column.

**Non LID associated interrupt indication register  
(Block\_base 0x0C00 + Register\_offset 0x0C)**

**TABLE 62 - NON LID ASSOCIATED INTERRUPT INDICATION REGISTER (REGISTER\_OFFSET 0x0C)**

Field	Bits	Length	Initial Value
SPI4_LOCK_UN	0	1	0b0
SPI3_LOCK_UN	1	1	0b0
SPI3_ICLK_UN	2	1	0b0
SPI3_ECLK_UN	3	1	0b0
SPI3_FLUSH	4	1	0b0
Reserved	31:5	27	0x0

The Non LID associated interrupt indication register is at Block\_Base 0x0C00. The Non LID interrupt indication register is used to determine the status of SPI-3 and SPI-4 port interrupts. The Non LID associated interrupt indication register is read and subsequently a "1" is written to acknowledge individual interrupts in this register. An interrupt is generated when enabled by the corresponding enable flag in the Non LID associated interrupt indication register. The bit fields in the Non LID associated interrupt indication register are described.

**SPI4\_LOCK\_UN** The SPI-4 interface can create an event indicating that the SPI-4 ingress has dropped data due the unavailability of ingress buffering.  
0=No operation  
1=The SPI-4 interface has dropped data due the unavailability of ingress buffering.

**SPI3\_LOCK\_UN** A SPI-3 interface can create an event indicating that a SPI-3 ingress has dropped data due the unavailability of ingress buffering.  
0=No operation  
1=A SPI-3 interface has dropped data due the unavailability of ingress buffering.

**SPI3\_ICLK\_UN** A SPI-3 interface can create an event indicating that a SPI-3 ingress clock has failed. No transitions were detected on a SPI-3 ingress clock (I\_FCLK)  
0=No operation  
1=A SPI-3 ingress clock has failed.

**SPI3\_ECLK\_UN** A SPI-3 interface can create an event indicating that a SPI-3 egress clock has failed. No transitions were detected on a SPI-3 egress clock (E\_FCLK)  
0=No operation  
1=A SPI-3 egress clock has failed.

**SPI3\_FLUSH** A SPI-3 interface can create an event indicating that a SPI-3 buffer has been flushed and data has been lost. A buffer is flushed if an address parity error is detected, or if an ingress buffer is not available at the time it is requested.

0=No operation  
1=A SPI-3 buffer has been flushed.

**Non LID associated interrupt enable register  
(Block\_base 0x0C00 + Register\_offset 0x0D)**

**TABLE 63 - NON LID ASSOCIATED INTERRUPT ENABLE REGISTER (REGISTER\_OFFSET 0x0D)**

Field	Bits	Length	Initial Value
SPI4_LOCK_UN	0	1	0b0
SPI3_LOCK_UN	1	1	0b0
SPI3_ICLK_UN	2	1	0b0
SPI3_ECLK_UN	3	1	0b0
SPI3_FLUSH	4	1	0b0
Reserved	31:5	27	0x0000

The Non LID associated interrupt enable register is at Block\_Base 0x0C00 + Register\_offset 0x0D. The Non LID associated interrupt enable register is used to mask the status of SPI-3 and SPI-4 port interrupts in the Non LID associated interrupt indication register. The Non LID associated interrupt enable register has read and write access. The bit fields in the Non LID associated interrupt enable register are active "1" interrupt enables for the corresponding bit fields in the Non LID associated interrupt indication register.

**LID associated interrupt indication register  
(Block\_base 0x0C00 + Register\_offset 0x0E)**

**TABLE 64 - LID ASSOCIATED INTERRUPT INDICATION REGISTER (REGISTER\_OFFSET 0x0E)**

Field	Bits	Length	Initial Value
EVENT_TYPE	5:0	6	0x00
Reserved	31:6	26	0x0

The LID associated interrupt indication register is at Block\_Base 0x0C00 + Register\_offset 0x0E. The LID associated interrupt indication register is used to determine the EVENT\_TYPE of SPI-3 and SPI-4 interrupts. The EVENT\_TYPE coding is given in the Table 66 - Non critical LID associated capture table (0x10-0x15). The LID associated interrupt indication register is read and subsequently a 0xFF must be written for interrupt acknowledge. An EVENT\_TYPE interrupt is generated when enabled by the EVENT\_TYPE enable flag in the LID associated interrupt enable register.

**LID associated interrupt enable register  
(Block\_base 0x0C00 + Register\_offset 0x0F)**

**TABLE 65 - LID ASSOCIATED INTERRUPT ENABLE REGISTER (REGISTER\_OFFSET 0x0F)**

Field	Bits	Length	Initial Value
EVENT_TYPE	5:0	6	0x00
Reserved	31:6	26	0x0

The LID associated interrupt enable register is at Block\_Base 0x0C00 + Register\_offset 0x0F. The LID associated interrupt enable register is used to mask the EVENT\_TYPE of SPI-3 and SPI-4 per-LID interrupts. The LID associated interrupt enable register has read and write access.

**Non critical LID associated capture table  
(Block\_base 0x0C00 + Register\_offset 0x10-0x15)**

**TABLE 66 - NON CRITICAL LID ASSOCIATED CAPTURE TABLE (REGISTER\_OFFSET 0x10-0x15)**

Register	EVENT_TYPE	Associated field
0x00	Inactive ingress SPI-3 logical port event	LP (8 bits)
0x01	SPI-3 ingress data parity error	LID (6 bits)
0x02	SPI-4 illegal SOP sequence event	LID (6 bits)
0x03	SPI-4 illegal EOP sequence event	LID (6 bits)
0x04	SPI-3 illegal SOP sequence event	LID (6 bits)
0x05	SPI-3 illegal EOP sequence event	LID (6 bits)

The Non critical LID associated capture table is at Block\_Base 0x0C00 + Register\_Offset 0x10-0x15. The Non critical LID associated capture table is used to determine the EVENT\_TYPE of SPI-3 and SPI-4 per-LID or per-LP interrupts. The EVENT\_TYPE coding is used to indicate which event or events are pertinent to the interrupt in the Table 64 - LID associated interrupt indication register(0x0E). The Non critical LID associated capture table is used to determine the EVENT, and multiple bits can be active at the same time. The Non critical LID associated capture table is read-only.

**SPI-3 to SPI-4 critical LID interrupt indication registers (Block\_base 0x0C00 + Register\_offset 0x16-0x17)**

**TABLE 67 - SPI-3 TO SPI-4 CRITICAL LID INTERRUPT INDICATION REGISTERS (REGISTER\_OFFSET 0x16-0x17)**

Register	Field	Bits	Length	Initial Value
0x16	LID[31:0]	31:0	32	0x00
0x17	LID[63:32]	31:0	32	0x00

The SPI-3 to SPI-4 critical LID interrupt indication registers are at Block\_Base 0x0C00 + Register\_offset 0x16-0x17.

Critical events are captured per LID in the SPI-3 to SPI-4 critical LID interrupt indication registers. An interrupt is generated when enabled by the enable flag in the SPI-3 to SPI-4 critical LID interrupt enable registers. A SPI-3 to SPI-4 critical LID interrupt indication register has read and write access. An interrupt indication is cleared by writing a logical one to the appropriate bit of a SPI-3 to SPI-4 critical LID interrupt indication register. Only one kind of critical event is defined-buffer overflow. Each bit of the LID field set to logical one indicates the presence of a buffer overflow event. A summary indication of as to which of the two sources, SPI-3 to SPI-4 or SPI-4 to SPI-3, is responsible for the critical interrupt is indicated in the Table 71 Critical events source indication register (0x1E).

**SPI-3 to SPI-4 critical LID interrupt enable registers (Block\_base 0x0C00 + Register\_offset 0x18-0x19)**

**TABLE 68 - SPI-3 TO SPI-4 CRITICAL LID INTERRUPT ENABLE REGISTERS (REGISTER\_OFFSET 0x18-0x19)**

Register	Field	Bits	Length	Initial Value
0x18	LID[31:0]	31:0	32	0x00
0x19	LID[63:32]	31:0	32	0x00

The SPI-3 to SPI-4 critical LID interrupt enable registers have read and write access. A SPI-3 to SPI-4 critical LID interrupt enable register bits enable the corresponding bits in a SPI-3 to SPI-4 critical LID interrupt indication register.

**SPI-4 to SPI-3 critical LID interrupt indication registers (Block\_base 0x0C00 + Register\_offset 0x1A-0x1B)**

**TABLE 69 - SPI-4 TO SPI-3 CRITICAL LID INTERRUPT INDICATION REGISTERS (REGISTER\_OFFSET 0x1A-0x1B)**

Register	Field	Bits	Length	Initial Value
0x1A	LID[31:0]	31:0	32	0x00
0x1B	LID[63:32]	31:0	32	0x00

The SPI-4 to SPI-3 critical LID interrupt indication registers are at Block\_Base 0x0C00 + Register\_offset 0x1A-0x1B.

Critical events are captured per LID in a SPI-4 to SPI-3 critical LID interrupt indication register. An interrupt is generated when enabled by the enable flag in the SPI-4 to SPI-3 critical LID interrupt enable register. The SPI-4 to SPI-3 critical LID interrupt indication registers have read and write access. An interrupt indication is cleared by writing a logical one to the appropriate bit of a SPI-4 to SPI-3 critical LID interrupt indication register. Only one kind of critical event is defined-buffer overflow. Each bit of a LID field set to logical one indicates the presence of a buffer overflow event. A summary indication of as to which of the two sources, SPI-3 to SPI-4 or SPI-4 to SPI-3, is responsible for the critical interrupt is indicated in the Table 71 Critical events source indication register (0x1E).

**SPI-4 to SPI-3 critical LID interrupt enable registers (Block\_base 0x0C00 + Register\_offset 0x1C-0x1D)**

**TABLE 70 - SPI-4 TO SPI-3 CRITICAL LID INTERRUPT ENABLE REGISTERS (REGISTER\_OFFSET 0x1C-0x1D)**

Register	Field	Bits	Length	Initial Value
0x1C	LID[31:0]	31:0	32	0x00
0x1D	LID[63:32]	31:0	32	0x00

The SPI-4 to SPI-3 critical LID interrupt enable registers have read and write access. The SPI-4 to SPI-3 critical LID interrupt enable register bits enable the corresponding bits in the SPI-4 to SPI-3 critical LID interrupt indication registers.

**Critical events source indication register (Block\_base 0x0C00 + Register\_offset 0x1E)**

**TABLE 71 - CRITICAL EVENTS SOURCE INDICATION REGISTER (REGISTER\_OFFSET 0x1E)**

Field	Bits	Length	Initial Value
SPI34_OVR	0	1	0b0
SPI43_OVR	1	1	0b0
Reserved	31:2	30	0x0

The bits in the Critical events source indication register are read only. Bit SPI34\_OVR reflects the logical OR result of all bits in the SPI-3 to SPI-4 critical LID associated interrupt indication registers. Bit SPI43\_OVR reflects the logical OR result of all bits in the SPI-4 to SPI-3 critical LID interrupt indication registers.

### 9.3.7 Block base 0x1000 registers

#### *SPI-3 ingress packet length configuration register (Block\_base 0x1000 + Register\_offset 0x00-0x3F)*

**TABLE 72 - SPI-3 INGRESS PACKET LENGTH CONFIGURATION REGISTER**

Field	Bits	Length	Initial Value
MIN_LENGTH	7:0	8	0x40
Reserved	15:8	8	0x00
MAX_LENGTH	29:16	14	0x5EE
Reserved	30:31	2	0x0

There is one set of SPI-3 ingress packet length configuration registers per SPI-3 ingress interface. Each SPI-3 ingress interface has 64 registers, one for each of the allowed LIDs supported per SPI-3 interface. Each register has read and write access. The minimum and maximum packet lengths per LID are provisioned using the SPI-3 ingress packet length configuration register. The bit fields of a SPI-3 ingress packet length configuration register are described.

**MIN\_LENGTH** SPI-3 ingress minimum packet length. The minimum packet length is programmed from 0 to 255 bytes. The resolution of the minimum packet length is one byte.

**MAX\_LENGTH** SPI-3 ingress maximum packet length. The maximum packet length is programmed from 0 to 16,383 bytes. The resolution of the maximum packet length is one byte.

### 9.3.8 Block base 0x1100 registers

#### *SPI-4 egress port descriptor table (Block\_base 0x1100 + Register\_offset 0x00-0x3F)*

**TABLE 73 - SPI-4 EGRESS PORT DESCRIPTOR TABLE (64 ENTRIES)**

Field	Bits	Length	Initial Value
MAX_BURST_H	3:0	4	0xF
MAX_BURST_S	7:4	4	0xF
DIRECTION	9:8	2	0x3
Reserved	31:10	22	0x000

There are four sets of SPI-4 egress port descriptor tables, one per SPI-3 interface. The minimum and maximum SPI-4 egress burst lengths per LID are provisioned using a SPI-4 egress port descriptor table. Each SPI-4 egress port descriptor table has read and write access. The bit fields of a SPI-4 egress port descriptor table are described. These fields need to be programmed only for SPI-4 egress (DIRECTION=0 in Table 74-SPI-4 egress direction code assignment).

**MAX\_BURST\_H** SPI-4 egress per-LID burst length when the attached device has declared hungry through the FIFO status channel. The number in the MAX\_BURST\_H field is taken to mean that one more than that number multiplied by 16 is the maximum hungry burst length. For example, programming the number 3 into the MAX\_BURST\_H field results in a maximum hungry burst size of  $(3 + 1) \times 16 = 64$  bytes.

**MAX\_BURST\_S** SPI-4 egress per-LID burst length when the attached device has declared starving through the FIFO status channel. The number in the MAX\_BURST\_S field is taken to mean that one more than that number

multiplied by 16 is the maximum starving burst length. For example, programming the number 7 into the MAX\_BURST\_S field results in a maximum starving burst size of  $(7 + 1) \times 16 = 128$  bytes. The MAX\_BURST\_S field should not be set to less than the MAX\_BURST\_H field.

**DIRECTION** The SPI-4 egress traffic can be captured by the microprocessor, directed to a SPI-3 egress port, to the SPI-4 egress port, or discarded. The Path selection is defined for each of the 64 LIDs by the associated DIRECTION field as shown in the following table.

**TABLE 74 - SPI-4 EGRESS DIRECTION CODE ASSIGNMENT**

DIRECTION	Path
00	SPI-4
01	Associated SPI-3
10	Capture to microprocessor
11	Discard

### 9.3.9 Block base 0x1200 registers

#### *SPI-3 ingress port descriptor tables (Block\_base 0x1200 + Register\_offset 0x00-0x3F)*

**TABLE 75 - SPI-3 INGRESS PORT DESCRIPTOR TABLE (BLOCK\_BASE 0x1200 + REGISTER\_OFFSET 0x00-0x3F)**

Field	Bits	Length	Initial Value
M	8:0	9	0x000
Reserved	15:9	7	0x00
Reserved	20:16	5	0x00
Reserved	23:21	3	0x0
FREE_SEGMENT	28:24	5	0x00
Reserved	31:29	3	0x0

There is one set of 64 SPI-3 ingress port descriptor tables per SPI-3 ingress interface. The SPI-3 ingress port descriptor tables are at Block\_base 0x1200 + Register\_offset 0x00-0x3F and have read and write access. Each SPI-3 ingress interface has 64 table entries for per-LID provisioning of M and FREE\_SEGMENT fields. The SPI-3 ingress port descriptor tables are used to control the amount of buffering and the free segment backpressure threshold of the available buffer segment pool for a SPI-3 ingress on a per-LID basis.

Each SPI-3 buffer segment pool is 128 Kbytes, divided into 508 segments of 256 bytes per segment. These 508 segments are shared among the LIDs initially programmed into the NR\_LID fields. A SPI-3 ingress LID can be allocated the maximum number of segments out of the available buffer segments, or can be programmed to fewer segments by decreasing the M field.

The FREE\_SEGMENT field is used, along with the M field, to set the free segment backpressure threshold for a LID on a SPI-3 ingress.

**M** The number of 256-byte buffer pool segments on a SPI-3 ingress port allocated to a LID. The range of M is 0x000 to 0x1FC (508 base 10), but can not exceed the number set by the choice of NR\_LID [Block\_base 0x1300 + Register\_offset 0x00].



**FREE\_SEGMENT** The FREE\_SEGMENT field is used to define the SPI-3 ingress per-LID free segment backpressure threshold based on the number of free buffer segments (M) available, as follows:

$$\text{THRESHOLD} = N * \text{FREE\_SEGMENT},$$

Where the value of N is defined as a function of the domain of M:

M[8:0]	N (base 10)
0x100 to 0x1FC	16
0x080 to 0x0FF	8
0x040 to 0x07F	4
0x020 to 0x03F	2
0x000 to 0x01F	1

The THRESHOLD thus defined is the number of free segments available for a LID at the time of backpressure initiation.

### 9.3.10 Block base 0x1300 registers

The SPI-3 ingress to SPI-4 egress Packet Fragment Processor and flow control registers are at Block\_Base 0x1300.

#### *SPI-3 to SPI-4 PFP register (Block\_base 0x1300 + Register\_offset 0x00)*

**TABLE 76 - SPI-3 TO SPI-4 PFP REGISTER (REGISTER\_OFFSET 0x00)**

Field	Bits	Length	Initial Value
NR_LID	2:0	3	0b011
Reserved	7:3	5	0x0

A SPI-3 ingress to SPI-4 egress PFP (Packet Fragment Processor) Register has read and write access. There is one SPI-3 to SPI-4 PFP Register per SPI-3 ingress. The bit fields of an 8-bit SPI-3 to SPI-4 PFP Register are described.

**NR\_LID** The maximum number of LIDs per SPI-3 physical ingress interface that will ever be used is programmed into the NR\_LID field. Once configured after reset, this value can not be changed. Fewer LIDs can be used by not activating some of the LIDs, but more LIDs than the value in NR\_LID are not allowed and will generate an error. The NR\_LID field is important, as the buffer segment pool is divided among the number of LIDs programmed into the NR\_LID field.

A 128 Kbyte SPI-3 to SPI-4 buffer segment pool for storing data bursts for the SPI-4 egress is available for each SPI-3 physical port. A configurable part of this buffer segment pool can be assigned to each of the possible LIDs allowed by the NR\_LID field value per SPI-3 physical interface. The buffer size for a LID can be configured in multiples (M) of 256 bytes. Modifications of the buffer size allocated to a LID are supported only when the logical port associated to the LID is disabled. Attempts to allocate more memory than available will generate an allocation error event. The indirect access module will discard the attempt.

A 128 Kbyte SPI-3 to SPI-4 buffer segment pool is divided into 508 buffer segments. Each buffer segment is equal to 256 bytes. The buffer segments are shared among the number of logical ports defined by the static NR\_LID configuration. The buffer segments do not have to be equally shared among the LIDs. One buffer segment corresponds to a data burst to be forwarded to the SPI-4 egress interface.

An example of the use of the buffer segment pool follows. For a SPI-3 ingress interface that will never have more than four LIDs, set the NR\_LID field for this interface to 0x01. This allows 256 buffer segments for a LID, with the total number of buffer segments for all 4 LIDs equal to 508. Let's say you want only 64 buffer segments for one of the LIDs. Program field M for that LID to 0x040 (64 base 10). Let's say you want to backpressure the SPI-3 ingress interface when 48 of the 64 allocated buffers for this LID are full. In other words, you want to exert SPI-3 ingress backpressure when only 16 segments remain for this LID. Since M=0x040, N=4 from the description of the M field above [Block\_base 0x1200]. Setting the FREE\_SEGMENT field to 4 then yields the desired THRESHOLD of 16.

**TABLE 77 - NR\_LID FIELD ENCODING**

NR_LID	Maximum Number of LIDs (base 10)	Maximum Buffer Segments for a LID (base 10)
0b000	1	508
0b001	4	256
0b010	8	128
0b011	16	64
0b100	32	32
0b101	64	16

#### *SPI-3 to SPI-4 flow control register (Block\_base 0x1300 + Register\_offset 0x01)*

**TABLE 78 - SPI-3 TO SPI-4 FLOW CONTROL REGISTER (REGISTER\_OFFSET 0x01)**

Field	Bits	Length	Initial Value
CREDIT_EN	0	1	0b1
BURST_EN	1	1	0b0
Reserved	7:2	6	0x00

A SPI-3 to SPI-4 flow control register has read and write access. There is one SPI-3 to SPI-4 flow control register per SPI-3 ingress. The bit fields of the SPI-3 to SPI-4 flow control register are described.

**CREDIT\_EN** The information received over the FIFO status channel is interpreted as status or credit information as selected by the CREDIT\_EN flag in the SPI-3 to SPI-4 flow control Register. If the status mode is used, data will be egressed until the status is changed by the attached device. If the credit mode is used, the SPI-4 egress will issue only one credit's worth data burst and then wait for another credit from the status channel before issuing another LID burst.

- 0=Status mode
- 1=Credit mode

**BURST\_EN** Multiple Burst Enable allows more than one burst to be sent to an LP. This feature is included to relieve systems with long latency between updates. When this feature is not enabled, only one burst per LP is allowed into the SPI-4 egress buffers.

- 0=Disable burst enable
- 1=Enable burst enable

**9.3.11 Block base 0x1600 registers**

The SPI-4 ingress registers are at Block\_base 0x1600.

**SPI-4 ingress packet length configuration (Block\_base 0x1600 + Register\_offset 0x00-0x3F)**

**TABLE 79 - SPI-4 INGRESS PACKET LENGTH CONFIGURATION (64 ENTRIES)**

Field	Bits	Length	Initial Value
MIN_LENGTH	7:0	8	0x40
Reserved	15:8	8	0x00
MAX_LENGTH	29:16	14	0x5EE
Reserved	31:30	2	0x0

There are four sets of 64 registers for SPI-4 ingress packet length configuration, one per SPI-3 interface. Each register has read and write access. The minimum and maximum packet lengths per LID are provisioned using a SPI-4 ingress packet length configuration register. The bit fields of a SPI-4 ingress packet length configuration register are described.

**MIN\_LENGTH** SPI-4 ingress minimum packet length. The minimum packet length is programmed from 0 to 255 bytes. The resolution is one byte.

**MAX\_LENGTH** SPI-4 ingress maximum packet length. The maximum packet length is programmed from 0 to 16,383 bytes. The resolution is one byte.

**9.3.12 Block base 0x1700 registers SPI-3 egress port descriptor table (Block\_base 0x1700 + Register\_offset 0x00-0x3F)**

**TABLE 80 - SPI-3 EGRESS PORT DESCRIPTOR TABLE (64 ENTRIES)**

Field	Bits	Length	Initial Value
MAX_BURST	3:0	4	0x0F
Reserved	7:4	4	0x0
DIRECTION	8:9	2	0b11
Reserved	31:10	22	0x00

There are 64 SPI-3 egress port descriptor tables per SPI-3 egress port. The SPI-3 egress port descriptor table has read and write access. The SPI-3 egress per LID packet fragment length and direction are provisioned using the SPI-3 egress port descriptor tables. The bit fields of a SPI-3 egress port descriptor table are described.

**MAX\_BURST** SPI-3 packet fragment length for a SPI-3 egress LP. One more than MAX\_BURST field multiplied by sixteen is the packet fragment length for the LP. For example, programming the number 3 into the MAX\_BURST field results in a packet fragment length of (3+1) x 16 = 64 bytes. The MAX\_BURST field is used to prioritize traffic.

**DIRECTION** The SPI-3 egress traffic is directed to a SPI-3 egress port. The Path selection is defined for each of the 64 LIDs by the associated DIRECTION field as shown in the following table.

**TABLE 81 - SPI-3 EGRESS DIRECTION CODE ASSIGNMENT**

DIRECTION	Path
00	SPI-3 physical
01	Reserved
10	Capture
11	Discard

**9.3.13 Block base 0x1800 registers SPI-4 ingress port descriptor table (Block\_base 0x1800 + Register\_offset 0x00-0x3F)**

**TABLE 82 - SPI-4 INGRESS PORT DESCRIPTOR TABLE (64 ENTRIES)**

Field	Bits	Length	Initial Value
M	8:0	9	0x000
Reserved	15:9	7	0x00
FREE_SEGMENT_S	20:16	5	0x00
Reserved	23:21	3	0x0
FREE_SEGMENT_H	28:24	5	0x00
Reserved	31:29	3	0x0

There is one set of 64 registers for SPI-4 ingress port descriptors per SPI-3 interface. The SPI-4 ingress port descriptor tables are 32 bits wide and have read and write access. Each of the SPI-4 ingress port descriptor tables is used to control the amount of buffering and the backpressure threshold of the available buffer segment pool for the SPI-4 ingress.

Each SPI-4 ingress buffer segment pool is 128 Kbytes, divided into 508 buffer segments of 256 bytes per segment. The 508 buffer segments can be shared among the LIDs initially programmed by the numerical field NR\_LID. Of the share of the buffer memory, a SPI-4 LID can be allocated the maximum number of segments permitted, or can be programmed to fewer segments by decreasing the M field. Decreasing M increases the chance of backpressure and possibly buffer overflow, but can result in lower latency.

The FREE\_SEGMENT\_S (starving threshold) and FREE\_SEGMENT\_H (hungry threshold) fields are used, along with the M field, to set the two backpressure settings per LID on the SPI-4 ingress. The FREE\_SEGMENT\_S field must always be greater than the FREE\_SEGMENT\_H field.

**M** The number of 256-byte buffer pool segments allocated to a LID. The range of M is 0x000 to 0x1FC (508 base 10), but can not exceed the number dictated by NR\_LID [Block\_base 0x1900 + Register\_offset 0x00].

**FREE\_SEGMENT\_S**

This field is used to define the SPI-4 ingress per-LID starving backpressure threshold based on the number of free buffer pool segments (M) available, as follows:

$THRESHOLD\_S = N * FREE\_SEGEMENT\_S$ , where the value of N is defined as:

M[8:0]	N
0x1FF to 0x100	16
0x0FF to 0x080	8
0x07F to 0x040	4
0x03F to 0x020	2
0x01F to 0x000	1

**FREE\_SEGMENT\_H**

This field is used to define the SPI-4 ingress per-LID hungry backpressure threshold based on the number of free buffer pool segments (M) available, as follows:

$THRESHOLD\_H = N * FREE\_SEGEMENT\_H$ , where the value of N is as defined for FREE\_SEGMENT\_S.

### 9.3.14 Block base 0x1900 registers

#### *SPI-4 to SPI-3 PFP register (Block\_base 0x1900 + Register\_Offset 0x00)*

**TABLE 83 - SPI-4 TO SPI-3 PFP REGISTER (0x00)**

Field	Bits	Length	Initial Value
NR_LID	2:0	3	0b011
Reserved	7:3	5	0x00

The SPI-4 ingress to SPI-3 egress Packet Fragment Processor registers are at Block\_Base 0x1900 + Register\_offset 0x00. A SPI-4 to SPI-3 PFP Register has read and write access. The bit fields of a SPI-4 to SPI-3 PFP Register are described.

**NR\_LID** The maximum number of LIDs per SPI-3 physical interface that will ever be used is programmed into the NR\_LID field. Once configured after reset, this value can not be changed. Fewer LIDs can be used by not activating some of the LIDs, but more LIDs than the value in NR\_LID are not allowed and will generate an error. The NR\_LID field is important, as the buffer segment pool is divided among the number of LIDs programmed into the NR\_LID field.

A 128 Kbyte SPI-4 to SPI-3 buffer segment pool for storing packet fragments for a SPI-3 egress is available per SPI-3 physical port. A configurable part of the buffer segment pool can be assigned to each of the LIDs, as determined by the NR\_LID value, per SPI-3 physical interface. The buffer size (M) for a LID can be configured in multiples of 256 bytes. Modifications of the buffer size allocated to a LID are supported only when the logical port associated to the LID is disabled. Attempts to allocate more memory than available will generate an allocation error event. The indirect access module will discard the attempt.

The 128 Kbyte SPI-4 to SPI-3 buffer segment pool is divided into 508 buffer segments. Each buffer segment is equal to 256 bytes. The buffer segments are shared among the number of logical ports defined by the static NR\_LID configuration. The buffer segments do not have to be equally shared among the allocated LIDs. One buffer segment corresponds to a packet fragment to be forwarded to a SPI-3 egress physical interface.

An example of the use of the buffer segment pool follows. For a SPI-3 egress interface that will never have more than eight LIDs, set the NR\_LID field for this interface to 0x02. This allows 128 buffer segments for a LID with the total number of buffer segments for all eight LIDs equal to 508.

Let's say you want only 24 (base 10) buffer segments for one of the LIDs. Program field M for that LID to 0x018 (24 base 10). Let's say you want to set the per-LID starving backpressure for the SPI-4 ingress interface when 20 of the 24 allocated buffers for this LID are full. In other words, you want to assert SPI-4 ingress starving when only 4 segments remain for this LID. Since M=0x018, N=1 from the description of the M field above [Block\_base 0x1800]. Setting the FREE\_SEGMENT\_S field to 4 then yields the desired THRESHOLD\_S of 4. Similarly, to set the per-LID SPI-4 ingress hungry threshold, THRESHOLD\_H, to trip when only 6 buffer segments remain for this LID, program the FREE\_SEGMENT\_H field for this LID to 6.

**TABLE 84 - NR\_LID FIELD ENCODING**

NR_LID	Maximum Number of LIDs	Maximum Buffer Segments for a LID
0b000	1	508
0b001	4	256
0b010	8	128
0b011	16	64
0b100	32	32
0b101	64	16

## 9.4 Common module indirect registers (Module\_base 0x8000)

**TABLE 85 - COMMON MODULE (MODULE\_BASE 0x8000) INDIRECT REGISTER TABLE**

Table Number, Page	Block_base, Register_offset	Title of Register
86, page 69	0x0000, 0x00-0xFF	SPI-4 ingress LP to LID map
87, page 69	0x0100, 0x00-0xFF	SPI-4 ingress calendar_0
88, page 69	0x0200, 0x00-0xFF	SPI-4 ingress calendar_1
89, page 69	0x0300, 0x00	SPI-4 ingress configuration register
90, page 70	0x0300, 0x01	SPI-4 ingress status configuration register
91, page 70	0x0300, 0x02	SPI-4 ingress status register
92, page 70	0x0300, 0x03	SPI-4 ingress inactive transfer port
93, page 71	0x0300, 0x04-0x05	SPI-4 ingress calendar configuration register
94, page 71	0x0300, 0x06	SPI-4 ingress watermark register
95, page 71	0x0300, 0x07-0x0A	SPI-4 ingress fill level register
96, page 71	0x0300, 0x0B-0x0E	SPI-4 ingress max fill level register
97, page 71	0x0300, 0x0F	SPI-4 ingress diagnostics register
98, page 72	0x0300, 0x10	SPI-4 ingress DIP-4 error counter
99, page 72	0x0300, 0x11	SPI-4 ingress bit alignment control register
100, page 72	0x0300, 0x12	SPI-4 ingress start up training threshold register
101, page 72	0x0400, 0x00-0xFF	SPI-4 egress LID to LP map
102, page 72	0x0500, 0x00-0xFF	SPI-4 egress calendar_0
103, page 73	0x0600, 0x00-0xFF	SPI-4 egress calendar_1
104, page 73	0x0700, 0x00	SPI-4 egress configuration register_0
105, page 73	0x0700, 0x01	SPI-4 egress configuration register_1
106, page 74	0x0700, 0x02	SPI-4 egress status register
107, page 74	0x0700, 0x03-0x04	SPI-4 egress calendar configuration register
108, page 74	0x0700, 0x05	SPI-4 egress diagnostics register
109, page 74	0x0700, 0x06	SPI-4 egress DIP-2 error counter
110, page 75	0x0800, 0x00	SPI-4 ingress bit alignment window register
111, page 75	0x0800, 0x01	SPI-4 ingress lane measure register
112, page 75	0x0800, 0x02-0x0B	SPI-4 ingress bit alignment counter register
113, page 75	0x0800, 0x0C-0x1F	SPI-4 ingress manual alignment phase/result register
114, page 75	0x0800, 0x2A	SPI-4 egress data lane timing register
115, page 76	0x0800, 0x2B	SPI-4 egress control lane timing register
116, page 76	0x0800, 0x2C	SPI-4 egress data clock timing register
117, page 76	0x0800, 0x2D	SPI-4 egress status timing register
118, page 76	0x0800, 0x2E	SPI-4 egress status clock timing register
119, page 77	0x0900, 0x00	PMON timebase control register
120, page 77	0x0900, 0x01	Timebase register
121, page 77	0x0900, 0x10	Clock generator control register
122, page 77	-----	OCLK and MCLK frequency select encoding
123, page 78	0x0900, 0x20	GPIO register
124, page 78	0x0900, 0x21- 0x25	GPIO monitor table
125, page 78	0x0900, 0x30	Version number register

### 9.4.1 Common module block base 0x0000 registers

#### *SPI-4 ingress LP to LID maps (Block\_base 0x0000 + Register\_offset 0x00 to 0xFF)*

**TABLE 86 - SPI-4 INGRESS LP TO LID MAP (256 ENTRIES, ONE PER LP)**

Field	Bits	Length	Initial Value
LID	5:0	6	0x00
PFP	7:6	2	0x0
ENABLE	8	1	0x0

There are 256 SPI-4 ingress LP to LID maps for the SPI-4 ingress interface at Block\_base 0x0000. The SPI-4 ingress LP to LID maps have read and write access. The SPI-4 ingress LP to LID maps are used to map SPI-4 ingress logical ports to logical identifiers used internally.

Data for an inactive LP having an entry in the calendar is forwarded to LID0. Therefore all the LPs that have entries in the calendar tables should be enabled.

**LID** The LID programmed is associated to the LP with the same number as the register address. Six bits support the 64 simultaneously active LIDs per SPI-3 physical interface.

**PFP** The PFP field is used to select among SPI-4 ingress to SPI-3 egress Packet Processing Engines. The number in the PFP field selects the PFP module to be used.

- 0x0=Select PFP Module A
- 0x1=Select PFP Module B
- 0x2=Select PFP Module C
- 0x3=Select PFP Module D

**ENABLE** The Enable is used to enable or disable the connection of an LP to a LID.

- 0=LP is disabled
- 1=LP is enabled

### 9.4.2 Common module block base 0x0100 registers

#### *SPI-4 ingress calendar\_0 (Block\_base 0x0100 + Register\_offset 0x00 to 0xFF)*

**TABLE 87 - SPI-4 INGRESS CALENDAR\_0 (256 ENTRIES)**

Field	Bits	Length	Initial Value
LP	7:0	8	0xFF

The SPI-4 ingress calendar\_0 is at Block\_base 0x0100 and has read and write access. When the SPI-4 ingress calendar\_0 is selected, SPI-4 ingress calendar\_0 is in use. There are 256 entries in the SPI-4 ingress calendar\_0 to schedule the updating of the FIFO status channel LPs to the attached device. If less than the maximum 256 LPs are needed on the SPI-4 interface, the calendar entries should be used for scheduling more frequent status updates for higher-speed LPs. The value of time-critical LPs must appear multiple times in the table. For example, a multi-PHY SPI-4 could have OC-48 channels appear in the calendar at four times the rate of OC-12 channels, since the higher data rate of the OC-48 channels would benefit from more frequent FIFO status channel updates. The LP field values range from 0x00 to 0xFF. The IDT88P8344 and the attached device must have identical calendars for ingress and the attached egress device. The ingress and egress calendars of the IDT88P8344 device do not have to match.

**LP** The LP value programmed schedules a status channel update according to the calendar sequence.

### 9.4.3 Common module block base 0x0200 registers

#### *SPI-4 ingress calendar\_1 (Block\_base 0x0200 + Register\_offset 0x00 to 0xFF)*

**TABLE 88 - SPI-4 INGRESS CALENDAR\_1 (256 ENTRIES)**

Field	Bits	Length	Initial Value
LP	7:0	8	0xFF

The SPI-4 ingress calendar\_1 is at Block\_base 0x0200 and has read and write access. When the SPI-4 ingress calendar\_1 is selected, SPI-4 ingress calendar\_1 is in use. There are 256 entries in the SPI-4 ingress calendar\_1 to schedule the updating of the FIFO status channel LPs to the attached device. If less than the maximum 256 LPs are needed on the SPI-4 interface, the calendar entries should be used for scheduling more frequent status updates for higher-speed LPs. The value of time-critical LPs must appear multiple times in the table. For example, a multi-PHY SPI-4 could have OC-48 channels appear in the calendar at four times the rate of OC-12 channels, since the higher data rate of the OC-48 channels would benefit from more frequent FIFO status channel updates. The LP field values range from 0x00 to 0xFF. The IDT88P8344 and the attached device must have identical calendars for ingress and the attached egress device. The ingress and egress of the IDT88P8344 do not have to match, however.

**LP** The LP value programmed schedules a status channel update according to the calendar sequence.

### 9.4.4 Common module block base 0x0300 registers

#### *SPI-4 ingress configuration register (Block\_base 0x0300 + Register\_offset 0x00)*

**TABLE 89 - SPI-4 INGRESS CONFIGURATION REGISTER (0x00)**

Field	Bits	Length	Initial Value
SPI-4_EN	0	1	0b0
Reserved	1	1	0x0
Reserved	2	1	0x0
I_CLK_EDGE	3	1	0x0
I_DSC	4	1	0x0
I_INSYNC_THR	9:5	5	0x1F
I_OUTSYNC_THR	13:10	4	0xF
I_CSW_EN	14	1	0x0
CAL_SEL	15	1	0x0
I_LOW	16	1	0b1

The SPI-4 ingress configuration register is at Block\_base 0x0300 and has read and write access.

The SPI-4 ingress configuration register is used to set the state of the SPI-4 ingress interface. The bit fields of the SPI-4 ingress configuration register are described.

**SPI-4\_EN** The SPI-4 ingress path is enabled using this field. The SPI-4 path is disabled during reset and while configuring the port, and then is enabled for normal use.

- 0=SPI-4 ingress is disabled
- 1= SPI-4 ingress is enabled

**I\_CLK\_EDGE** The SPI-4 ingress LVTTTL status clock active clock edge is selected using the I\_CLK\_EDGE field.

- 0=SPI-4 ingress LVTTTL status clock uses the rising edge
- 1= SPI-4 ingress LVTTTL status clock uses the falling edge

**I\_DSC** The I\_DSC bit is used to protect against a random data error during de-skew.

- 0= One de-skew result is needed for data de-skew
- 1= Two consecutive de-skew results are needed for data de-skew (recommended setting)

**I\_INSYNC\_THR** The SPI-4 ingress DIP-4 in synchronization threshold is controlled using the I\_INSYNC\_THR field. It is recommended to use the initial value.

**I\_OUTSYNC\_THR** The SPI-4 ingress DIP-4 out-of synchronization threshold is controlled using the I\_OUTSYNC\_THR field. It is recommended to use the initial value.

**I\_CSW\_EN** The ingress calendar switch enable bit is used to enable the switching of the active calendars. It is recommended to use the initial value.

- 0=Ingress calendar switch disabled. Only SPI-4 ingress calendar\_0 is used.
- 1=Ingress calendar switch enabled. Calendar\_0 or calendar\_1 can be used.

**CAL\_SEL** The calendar select bit selects between SPI-4 ingress calendar\_0 and SPI-4 ingress calendar\_1. The CAL\_SEL bit is only valid if the I\_CSW\_EN bit is set to a logic one.

- 0=SPI-4 ingress calendar\_0 is selected
- 1=SPI-4 ingress calendar\_1 is selected if the I\_CSW\_EN bit is set to a logic one

**I\_LOW** The I\_LOW field selects the SPI-4 ingress clock frequency range.

- 0=SPI-4 ingress clock is greater than or equal to 200 MHz
- 1=SPI-4 ingress clock is less than 200 MHz

#### **SPI-4 ingress status configuration register (Block\_base 0x0300 + Register\_offset 0x01)**

**TABLE 90 - SPI-4 INGRESS STATUS CONFIGURATION REGISTER (REGISTER\_OFFSET 0x01)**

Field	Bits	Length	Initial Value
FIFO_MAX_T	23:0	24	0
ALPHA_FIFO	31:24	8	0

The SPI-4 ingress status configuration register is at Block\_base 0x0300 and has read and write access.

The SPI-4 ingress status configuration register is used to set the state of the SPI-4 ingress FIFO status path interface. The bit fields of the SPI-4 ingress status configuration register are described.

**FIFO\_MAX\_T** The SPI-4 ingress FIFO\_MAX\_T field is the maximum time interval between scheduling of training sequences on the FIFO status path interface. The units are the number of times the calendar is sent before scheduling the training sequence.

**ALPHA\_FIFO** The SPI-4 ingress ALPHA\_FIFO field is the number of repetitions of the status training sequence that must be scheduled every FIFO\_MAX\_T cycles. The value for alpha used is actually one more than the ALPHA\_FIFO value programmed into the ALPHA\_FIFO field.

#### **SPI-4 ingress status register (Block\_base 0x0300 + Register\_offset 0x02)**

**TABLE 91 - SPI-4 INGRESS STATUS REGISTER (REGISTER\_OFFSET 0x02)**

Field	Bits	Length	Initial Value
I_SYNCH	0	1	0
I_DSK_OOR	1	1	0
DCLK_AV	2	1	0

The SPI-4 ingress status register is at Block\_base 0x0300 and has read-only access.

The SPI-4 ingress status register is used to set the state of the SPI-4 ingress synchronization.

The bit fields of the SPI-4 ingress status register are described.

**I\_SYNCH** The SPI-4 ingress I\_SYNCH field describes the synchronization state of the SPI-4 ingress data path.

- 0=SPI-4 ingress data path is out of synchronization
- 1=SPI-4 ingress data path is in synchronization

**I\_DSK\_OOR** The SPI-4 ingress I\_DSK\_OOR field describes the de-skew state of the SPI-4 ingress data path.

- 0=SPI-4 ingress data path de-skew is within range
- 1= SPI-4 ingress data path de-skew is out of range

**DCLK\_AV** The SPI-4 ingress DCLK\_AV field describes the availability state of the SPI-4 ingress clock.

- 0=SPI-4 ingress clock is not available
- 1= SPI-4 ingress clock is available

#### **SPI-4 ingress inactive transfer port (Block\_base 0x0300 + Register\_offset 0x03)**

**TABLE 92 - SPI-4 INGRESS INACTIVE TRANSFER PORT (REGISTER\_OFFSET 0x03)**

Field	Bits	Length	Initial Value
INACT_LP	7:0	8	0

The SPI-4 ingress inactive transfer port is at Block\_base 0x0300 and has read-only access.

The SPI-4 ingress inactive transfer port INACT\_LP field is used to monitor the LP associated with the latest inactive transfer. The INACT\_LP field can change at any time and is used for diagnostics only.

**INACT\_LP** The SPI-4 ingress INACT\_LP field contains the numeric value of the LP associated with the last inactive LP transfer, used for diagnostics only.

### SPI-4 ingress calendar configuration register (Block\_base 0x0300 + Register\_offset 0x04 - 0x05)

**TABLE 93 - SPI-4 INGRESS CALENDAR CONFIGURATION REGISTER (0x04 to 0x05)**

Field	Bits	Length	Initial Value
I_CAL_M	7:0	8	0
I_CAL_LEN	13:8	6	0x01

The SPI-4 ingress calendar configuration registers are at Block\_base 0x0300 and have read and write access. The Register\_offset for calendar\_0 is 0x04. The Register\_offset for calendar\_1 is 0x05.

The bitfields of a SPI-4 ingress calendar configuration register are described.

Some devices have a fixed calendar length and a fixed calendar M, while the Bridgeport calendar length has to be multiply of 4, and the calendar M is programmable. Therefore, the user may need to add an FPGA between the Bridgeport & the adjacent device SPI-4 status signals.

**I\_CAL\_M** The I\_CAL\_M value programmed defines the number of times the calendar sequence is repeated before a DIP-2 parity and "1 1" framing words are inserted. The actual calendar\_M value used is one more than the value programmed into the I\_CAL\_M field.

**I\_CAL\_LEN** The I\_CAL\_LEN value programmed defines the length of the SPI-4 ingress calendar. The actual length of the calendar is four times the value of one more than the I\_CAL\_LEN field:  $(I\_CAL\_LEN + 1) * 4$ . For example, if the I\_CAL\_LEN field is programmed to 0x04, the actual value used is 0x14. The calendar length must be at least as large as the number of active SPI-4 ingress LPs.

### SPI-4 ingress watermark register (Block\_base 0x0300 + Register\_offset 0x06)

**TABLE 94 - SPI-4 INGRESS WATERMARK REGISTER (REGISTER\_OFFSET 0x06)**

Field	Bits	Length	Initial Value	Function
WATERMARK	4:0	5	0x0D	Watermark for PFP A
reserved	7:5	3	0	
WATERMARK	12:6	5	0x0D	Watermark for PFP B
reserved	15:13	3	0	
WATERMARK	20:16	5	0x0D	Watermark for PFP C
reserved	23:21	3	0	
WATERMARK	28:24	5	0x0D	Watermark for PFP D
reserved	31:29	3	0	

SPI-4 ingress Watermark Register is at Block\_base 0x0300, Register\_offset 0x06. The SPI-4 ingress Watermark Register has read and write access. A SPI-4 interface can be set to a Watermark Value per PFP. 0x1F is the highest watermark that can be set, meaning all ingress buffers will be full before backpressure will be initiated on a SPI-4 ingress interface PFP. A WATERMARK field value of 0x0F is used to set a watermark for a half-full ingress buffer before tripping backpressure. The units of WATERMARK are one-thirty-second of the available ingress buffering per unit. Each unit is equal to 128 bytes.

### SPI-4 ingress fill level register (Block\_base 0x0300 + Register\_offset 0x07-0x0A)

There is one SPI-4 ingress fill level register per SPI-3 interface at Block\_base 0x0300. Each register has read-only access.

The SPI-4 ingress fill level register for PFP A is at Block\_base 0x0300 + Register\_offset 0x07.

The SPI-4 ingress fill level register for PFP B is at Block\_base 0x0300 + Register\_offset 0x08.

The SPI-4 ingress fill level register for PFP C is at Block\_base 0x0300 + Register\_offset 0x09.

The SPI-4 ingress fill level register for PFP D is at Block\_base 0x0300 + Register\_offset 0x0A.

The bit field of a SPI-4 ingress fill level register is described.

**TABLE 95 - SPI-4 INGRESS FILL LEVEL REGISTER (0x07 to 0x0A)**

Field	Bits	Length	Initial Value
FILL_CUR	5:0	6	0x0

**FILL\_CUR** Current SPI-4 ingress buffer fill level. Since this is a real-time register, the value read from it will change rapidly and is used for internal diagnostics only.

### SPI-4 ingress max fill level register (Block\_base 0x0300 + Register\_offset 0x0B to 0x0E)

**TABLE 96 - SPI-4 INGRESS MAX FILL LEVEL REGISTER (0x0B to 0x0E)**

Field	Bits	Length	Initial Value
FILL_MAX	5:0	6	0x00

There are four SPI-4 ingress max fill level registers, one per SPI-3 interface, at Block\_base 0x0300. Each register has read-only access, and is cleared after reading. The value 0x20 is the highest filling level, meaning all ingress buffers on a PFP had been full at some time since the last read of the FILL\_MAX field. The units of FILL\_MAX are one-thirty-second of the available ingress buffering per PFP. Each unit is equal to 128 bytes.

The SPI-4 ingress max fill level register for PFP A is at Block\_base 0x0300 + Register\_offset 0x0B.

The SPI-4 ingress max fill level register for PFP B is at Block\_base 0x0300 + Register\_offset 0x0C.

The SPI-4 ingress max fill level register for PFP C is at Block\_base 0x0300 + Register\_offset 0x0D.

The SPI-4 ingress max fill level register for PFP D is at Block\_base 0x0300 + Register\_offset 0x0E.

The bit field of a SPI-4 ingress max fill level register is described.

**FILL\_MAX** Maximum SPI-4 ingress buffer fill level since the last read of the SPI-4 ingress max fill level register.

### SPI-4 ingress diagnostics register (Block\_base 0x0300 + Register\_offset 0x0F)

**TABLE 97 - SPI-4 INGRESS DIAGNOSTICS REGISTER (REGISTER\_OFFSET 0x0F)**

Field	Bits	Length	Initial Value
I_FORCE_TRAIN	0	1	0
I_ERR_INS	1	1	0
I_DIP_NUM	5:2	4	0

The SPI-4 ingress diagnostics register is addressed from Block\_base 0x0300 + Register\_offset 0x0F. The SPI-4 ingress diagnostics register has read and write access. The SPI-4 ingress diagnostics register is used in port diagnostics to force continuous training on the SPI-4 ingress status interface, insert a DIP-2 error on the SPI-4 ingress status interface, and read the number of DIP-2 errors seen on the SPI-4 egress status interface.

**I\_FORCE\_TRAIN** The I\_FORCE\_TRAIN field is used to force continuous training on the SPI-4 ingress status interface.  
 0=Normal status channel operation  
 1=Force continuous training on the SPI-4 ingress status interface

**I\_ERR\_INS** The I\_ERR\_INS field is used to insert the number of DIP-2 errors on the SPI-4 ingress status interface programmed into the I\_DIP\_NUM field. After the DIP-2 errors are inserted, the I\_ERR\_INS field will clear itself.  
 0=Normal status channel operation  
 1= Insert DIP-2 errors on the SPI-4 ingress status interface

**I\_DIP\_NUM**  
 The I\_DIP\_NUM field is used to create the number of DIP-2 errors programmed into the I\_DIP\_NUM field on the SPI-4 egress status interface..

**SPI-4 ingress DIP-4 error counter (Block\_base 0x0300 + Register\_offset 0x10)**

**TABLE 98 - SPI-4 INGRESS DIP-4 ERROR COUNTER (REGISTER\_OFFSET 0x10)**

Field	Bits	Length	Initial Value
DIP_4	15:0	16	0

The SPI-4 ingress DIP-4 error counter is addressed from Block\_base 0x0300 + Register\_offset 0x10. The SPI-4 ingress DIP-4 error counter has read access, and automatically clears itself after a read. The SPI-4 ingress DIP-4 error counter is used in port diagnostics to verify the integrity of the SPI-4 ingress data path.

**DIP\_4** The DIP\_4 field is used to read the number of DIP-4 errors seen on the SPI-4 egress status interface. The DIP\_4 field saturates at the value 0xFFFF, and is automatically cleared after reading to re-start DIP-4 error counter accumulation.

**SPI-4 ingress bit alignment control register (Block\_base 0x0300 + Register\_offset 0x11)**

**TABLE 99 - SPI-4 INGRESS BIT ALIGNMENT CONTROL REGISTER (REGISTER\_OFFSET 0x11)**

Field	Bits	Length	Initial Value
FORCE	0	1	0

The SPI-4 ingress bit alignment control register is addressed from Block\_base 0x0300 + Register\_offset 0x11. The SPI-4 ingress bit alignment control register has read and write access. The SPI-4 ingress bit alignment control register is used to overrule the automatically selected bit phase alignments and go to manual mode. In manual mode, the PHASE\_ASSIGN field [Block\_base 0x0800 + Register\_offset 0x0c - 0x1F] now defines the selected phase.

**FORCE** The FORCE field is used to manually align the SPI-4 ingress data.  
 0=Normal bit alignment operation  
 1=Force to manual bit alignment mode on SPI-4 ingress data using the PHASE\_ASSIGN field.

**SPI-4 ingress start up training threshold register (Block\_base 0x0300 + Register\_offset 0x12)**

**TABLE 100 - SPI-4 INGRESS START UP TRAINING THRESHOLD REGISTER (REGISTER\_OFFSET 0x12)**

Field	Bits	Length	Initial Value
STRT_TRAIN	7:0	8	0

The SPI-4 ingress start up training threshold register is addressed from Block\_base 0x0300 + Register\_offset 0x12. The SPI-4 ingress start up training threshold register has read and write access. The SPI-4 ingress start up training threshold register is used to set the number of consecutive training patterns that will lead to OUT\_OF\_SYNC on the SPI-4 ingress data. If the STRT\_TRAIN field is set to zero, then the OUT\_OF\_SYNC feature is disabled.

**STRT\_TRAIN** The STRT\_TRAIN field is used to set the number of consecutive training patterns that will lead to OUT\_OF\_SYNC on the SPI-4 ingress data interface.

**9.4.5 Common module block base 0x0400 registers**

**SPI-4 egress LID to LP map (Block\_base 0x0400)**

**TABLE 101 - SPI-4 EGRESS LID TO LP MAP (256 ENTRIES)**

Field	Bits	Length	Initial Value
LP	7:0	8	0x00
EN	8	1	0b0

There are 256 entries in the SPI-4 egress LID to LP map for the SPI-4 egress interface. The entries are at Block\_base 0x0400 + LID. For example, LID 0x00 is at Block\_base 0x0400 + 0x00. A SPI-4 egress LID to LP map has read and write access. A SPI-4 egress LID to LP map is used to map a logical identifier used internally to a SPI-4 egress logical port.

- 0x00 - 0x3F of the LID map is for Module A LIDs 0x00 - 0x3F
- 0x40 - 0x7F of the LID map is for Module B LIDs 0x00 - 0x3F
- 0x80 - 0xBF of the LID map is for Module C LIDs 0x00 - 0x3F
- 0xC0 - 0xFF of the LID map is for Module D LIDs 0x00 - 0x3F

Data for an inactive LP having an entry in the calendar is forwarded to LID0. Therefore all the LPs that have entries in the calendar tables should be enabled.

**LP** The LP programmed is associated to the LID with the same number as the register address. Eight bits support the 256 possible LPs on the SPI-4 physical interface. 256 simultaneously active LPs are supported by the IDT88P8344 device.

**EN** The EN bit is used to enable or disable the connection of a LID to an LP.  
 0=LP is disabled  
 1=LP is enabled

**9.4.6 Common module block base 0x0500 registers**

**SPI-4 egress calendar\_0 (Block\_base 0x0500)**

**TABLE 102 - SPI-4 EGRESS CALENDAR\_0 (256 LOCATIONS)**

Field	Bits	Length	Initial Value
LP	7:0	8	0xFF

The SPI-4 egress calendar\_0 is at Block\_base 0x0500 and has read and write access. When the SPI-4 egress calendar\_0 is selected, calendar\_0 is in use. There are 256 entries in the SPI-4 egress calendar\_0 to schedule the



updating of the FIFO status channel LPs to the attached device. If less than the maximum 256 LPs are needed on the SPI-4 interface, the calendar entries should be used for scheduling more frequent status updated for higher-speed LPs. The value of time-critical LPs must appear multiple times in the table. For example, a multi-PHY SPI-4 could have OC-48 channels appear in the calendar at four times the rate of OC-12 channels, since the higher data rate of the OC-48 channels would benefit from more frequent FIFO status channel updates. The LP field values range from 0x00 to 0xFF. The IDT88P8344 and the attached devices must have identical calendars.

**LP** The LP value programmed schedules a status channel update according to the calendar sequence.

**9.4.7 Common module block base 0x0600 registers**

**SPI-4 egress calendar\_1 (Block\_base 0x0600)**

**TABLE 103 - SPI-4 EGRESS CALENDAR\_1 (256 LOCATIONS)**

Field	Bits	Length	Initial Value
LP	7:0	8	0xFF

The SPI-4 egress calendar\_1 is at Block\_base 0x0600 and had read and write access. When the SPI-4 egress calendar\_1 is selected, calendar\_1 is in use. There are 256 entries in the SPI-4 egress calendar\_1 to schedule the updating of the FIFO status channel LPs to the attached device. If less than the maximum 256 LPs are needed on the SPI-4 interface, the calendar entries should be used for scheduling more frequent status updated for higher-speed LPs. The value of time-critical LPs must appear multiple times in the table. For example, a multi-PHY SPI-4 could have OC-48 channels appear in the calendar at four times the rate of OC-12 channels, since the higher data rate of the OC-48 channels would benefit from more frequent FIFO status channel updates. The LP field values range from 0x00 to 0xFF. The IDT88P8344 and the attached devices must have identical calendars.

**LP** The LP value programmed schedules a status channel update according to the calendar sequence.

**9.4.8 Common module block base 0x0700 registers**

**SPI-4 egress configuration register\_0 (Block\_base 0x0700 + Register\_offset 0x00)**

**TABLE 104 – SPI-4 EGRESS CONFIGURATION REGISTER\_0 (REGISTER\_OFFSET 0x00)**

Field	Bits	Length	Initial Value
Reserved	2:0	3	0
E_CLK_EDGE	3	1	0
E_DSC	4	1	0
E_INSYNC_THR	9:5	5	0x1F
E_OUTSYNC_THR	13:10	4	0xF
E_CSW_EN	14	1	0
Reserved	15	1	0
E_LOW	16	1	1
NOSTAT	17	1	0

The SPI-4 egress configuration register\_0 is at Block\_base 0x0700 and has read and write access.

The SPI-4 egress configuration register\_0 is used to set the state of the SPI-4 egress interface. The bit fields of the SPI-4 egress configuration register\_0 are described.

**E\_CLK\_EDGE** The SPI-4 egress clock active clock edge is selected using the E\_CLK\_EDGE field.

- 0=SPI-4 egress clock uses the rising clock edge
- 1=SPI-4 egress clock uses the falling clock edge

**E\_DSC** The E\_DSC bit enables or disables de-skewing of the SPI-4 egress data lines.

- 0=Data de-skewing is disabled
- 1=Data de-skewing is enabled (recommended)

**E\_INSYNC\_THR** The SPI-4 egress DIP-2 in-synchronization threshold is controlled using the E\_INSYNC\_THR field. It is recommended to use the initial value.

**E\_OUTSYNC\_THR** The SPI-4 egress DIP-2 out-of-synchronization threshold is controlled using the E\_OUTSYNC\_THR field. It is recommended to use the initial value.

**E\_CSW\_EN** The ingress calendar switch enable bit is used to enable the switching of the active calendars following the reception of the calendar selection word on the status channel. It is recommended to use the initial value.

- 0=Egress calendar switch is disabled. Only SPI-4 egress calendar\_0 is used.
- 1=Egress calendar switch is enabled. Calendar\_0 or calendar\_1 will be used.

**E\_LOW** The E\_LOW field selects the SPI-4 egress clock frequency range.

- 0=SPI-4 egress clock is greater than or equal to 200 MHz
- 1=SPI-4 egress clock is less than 200 MHz

**NOSTAT** The NOSTAT bit enables the no status channel option. Once NOSTAT is set, the status channel is ignored. There is no DIP-2 error checking, and no status channel updating. The received status is fixed to starving. The data channel is put into the IN\_SYNC state.

- 0=Normal status channel operation
- 1=No status channel option is selected

**SPI-4 egress configuration register\_1 (Block\_base 0x0700 + Register\_offset 0x01)**

**TABLE 105 - SPI-4 EGRESS CONFIGURATION REGISTER\_1 (REGISTER\_OFFSET 0x01)**

Field	Bits	Length	Initial Value
DATA_MAX_T	23:0	24	0
ALPHA	31:24	8	0

The SPI-4 egress configuration register\_1 is at Block\_base 0x0700 and has read and write access.

The SPI-4 egress configuration register\_1 is used to set the state of the SPI-4 egress FIFO status path interface. The bit fields of the SPI-4 egress configuration register\_1 are described.

**DATA\_MAX\_T** The SPI-4 egress DATA\_MAX\_T field is the maximum time interval between scheduling of training sequences on the egress data path interface. The purpose of the data training interval is to allow the de-skewing of plus or minus one bit time on the egress data interface if needed. The time is set for the DATA\_MAX\_T field multiplied by 128 cycles.

**ALPHA** The SPI-4 egress ALPHA field is the number of repetitions of the data training sequence that must be scheduled every DATA\_MAX\_T cycles. The value for alpha used is actually one more than the ALPHA value programmed into the ALPHA field.

**SPI-4 egress status register (Block\_base 0x0700 + Register\_offset 0x02)**

**TABLE 106 - SPI-4 EGRESS STATUS REGISTER (REGISTER\_OFFSET 0x02)**

Field	Bits	Length	Initial Value
E_SYNCH	0	1	0
E_DSK_OOR	1	1	0
SCLK_AV	2	1	0

**SPI-4 egress status register**  
The SPI-4 egress status register is at Block\_base 0x0700 and has read-only access.

The SPI-4 egress status register is used to set the state of the SPI-4 egress synchronization.

The bit fields of the SPI-4 egress status register are described.

**E\_SYNCH** The SPI-4 egress E\_SYNCH field describes the synchronization state of the SPI-4 egress data path.

- 0=SPI-4 egress data path is out of synchronization
- 1=SPI-4 egress data path is in synchronization

**E\_DSK\_OOR** The SPI-4 egress E\_DSK\_OOR field describes the de-skew state of the SPI-4 egress data path.

- 0=SPI-4 egress data path de-skew is within range
- 1=SPI-4 egress data path de-skew is out of range

**SCLK\_AV** The SPI-4 egress SCLK\_AV field describes the availability state of the SPI-4 egress status channel clock. This function is not available if SCLK < 0.5 MCLK.

- 0=SPI-4 egress status channel clock is not available
- 1=SPI-4 egress status channel clock is available

**SPI-4 egress calendar configuration register (Block\_base 0x0700 + Register\_offset 0x03 - 0x04)**

**TABLE 107 - SPI-4 EGRESS CALENDAR CONFIGURATION REGISTER (REGISTER\_OFFSET 0x03 - 0x04)**

Field	Bits	Length	Initial Value
E_CAL_M	7:0	8	0
E_CAL_LEN	13:8	6	0x01

The SPI-4 egress calendar configuration registers are at Block\_base 0x0300 and has read and write access. The Register\_offset for calendar\_0 is 0x03. The register offset for calendar\_1 is 0x04.

The bit fields of the SPI-4 egress calendar configuration register are described.

The IDT88P8344 calendar length can be programmed to any multiple of 4 using suitable values for the calendar entries, calendar length and calendar M. If the adjacent device is unable to configure its calendar to be a multiple of 4, conversion logic may be needed between the adjacent device SPI-4 status signals and the 88P8344 signals.

**E\_CAL\_M** The E\_CAL\_M value programmed defines the number of times the calendar sequence is repeated before a DIP-2 parity and "1 1" framing words are inserted. The actual calendar\_M value used is one more than the value programmed into the E\_CAL\_M field.

**E\_CAL\_LEN** The E\_CAL\_LEN value programmed defines the length of the SPI-4 egress calendar. The actual length of the calendar is four times one more than the value programmed into the E\_CAL\_LEN field. For example, if the E\_CAL\_LEN field is programmed to 0x3F, the actual value used is 0x100. The resulting calendar length must be at least as large as the number of active SPI-4 egress LPs.

**SPI-4 egress diagnostics register (Block\_base 0x0700 + Register\_offset 0x05)**

**TABLE 108 - SPI-4 EGRESS DIAGNOSTICS REGISTER (REGISTER\_OFFSET 0x05)**

Field	Bits	Length Value	Initial
E_FORCE_TRAIN	0	1	0
E_ERR_INS	1	1	0
E_DIP_NUM	5:2	4	0
BIT_DELAY	7:6	2	0

The SPI-4 egress diagnostics register is addressed from Block\_base 0x0700 + Register\_offset 0x05. The SPI-4 egress diagnostics register has read and write access.

**E\_FORCE\_TRAIN** The E\_FORCE\_TRAIN field is used to force continuous training on the SPI-4 egress status interface.

- 0=Normal status channel operation
- 1=Force continuous training on the SPI-4 egress status interface

**E\_ERR\_INS** The E\_ERR\_INS field is used to insert the number of DIP-4 errors on the SPI-4 egress data interface that have been programmed into the E\_DIP\_NUM field. After the DIP-4 errors are inserted, the E\_ERR\_INS field will clear itself.

- 0=Normal status channel operation
- 1= Insert DIP-4 errors on the SPI-4 egress data interface

**E\_DIP\_NUM** The E\_DIP\_NUM field is used to create DPI-4 errors on the SPI-4 egress data interface. The number of errors generated is equal to the value of the E\_DIP\_NUM field.

**BIT\_DELAY** The BIT\_DELAY field is used to delay SPI-4 egress data bit line 0 by the number of bits programmed into the BIT\_DELAY field. This may be used for diagnostics.

**SPI-4 egress DIP-2 error counter (Block\_base 0x0700 + Register\_offset 0x06)**

**TABLE 109 - SPI-4 EGRESS DIP-2 ERROR COUNTER (REGISTER\_OFFSET 0x06)**

Field	Bits	Length	Initial Value
DIP_2	15:0	16	0

The SPI-4 egress DIP-2 error counter is addressed from Block\_base 0x0700 + Register\_offset 0x06. The SPI-2 egress DIP-2 error counter has read access, and automatically clears itself after a read. The SPI-4 egress DIP-2 error counter is used in port diagnostics to verify the integrity of the SPI-4 egress status channel.

**DIP\_2** The DIP\_2 field is used to read the number of DIP-2 errors seen on the SPI-4 egress status interface. The DIP\_2 field saturates at the value 0xFFFF, and is automatically cleared after reading to re-start DIP-2 error counter accumulation.

**9.4.9 Common module block base 0x0800 registers**

**SPI-4 ingress bit alignment window register (Block\_base 0x0800 + Register\_offset 0x00)**

**TABLE 110 - SPI-4 INGRESS BIT ALIGNMENT WINDOW REGISTER (REGISTER\_OFFSET 0x00)**

Field	Bits	Length	Initial Value
W	15:0	16	0xFFFF

The SPI-4 ingress bit alignment window register is addressed from Block\_base 0x0800 + Register\_offset 0x00. The SPI-4 ingress bit alignment window register has read and write access. The SPI-4 ingress bit alignment window register is used in manual bit alignment procedures and it is recommended to leave the W field at its initial value.

**W** The W field is used to define the width of the SPI-4 ingress bit alignment observation window by setting the time between bit alignment operations.

**SPI-4 ingress lane measure register (Block\_base 0x0800 + Register\_offset 0x01)**

**TABLE 111 - SPI-4 INGRESS LANE MEASURE REGISTER (REGISTER\_OFFSET 0x01)**

Field	Bits	Length	Initial Value
LANE	4:0	5	0
Reserved	7:5	3	0
MEASURE_BUSY	8	1	0

The SPI-4 ingress lane measure register is addressed from Block\_base 0x0800 + Register\_offset 0x01. The SPI-4 ingress lane measure register has read and write access. SPI-4 ingress lane measure register is used in manual bit alignment procedures and it is recommended to leave the SPI-4 ingress lane measure register at its initial value.

**LANE** The LANE field is used to manually control the measurement of SPI-4 ingress data lane alignment. The LANE field is intended for diagnostics only and is not needed in normal operation.

- 0=DATA0 lane selected for measurement
- x=DATAx lane selected for measurement
- 15=DATA15 lane selected for measurement
- 16=CTL selected for measurement
- 17=Egress status 0 selected for measurement
- 18=Egress status 1 selected for measurement
- 19=Chip test feature not available for diagnostics

**MEASURE\_BUSY** The MEASURE\_BUSY field is used to observe when the LANE process is busy for manual lane assignment procedures. The MEASURE\_BUSY field is intended for diagnostics only and is not needed for normal operation.

- 0=Normal operation
- 1=Lane process is busy

**SPI-4 ingress bit alignment counter register (Block\_base 0x0800 + Register\_offset 0x02 – 0x0B)**

**TABLE 112 - SPI-4 INGRESS BIT ALIGNMENT COUNTER REGISTER (0x02 to 0x0B)**

Field	Bits	Length	Initial Value
C[n]	9:0	10	0

The SPI-4 ingress bit alignment counter registers at Block\_base 0x0800 are read-only and contains the values of the bit alignment counters used for manual lane alignment. The registers are intended for diagnostics only and are not needed in normal operation.

**SPI-4 ingress manual alignment phase/result register (Block\_base 0x0800 + Register\_offset 0x0C – 0x1F)**

**TABLE 113 - SPI-4 INGRESS MANUAL ALIGNMENT PHASE/RESULT REGISTER (0x0C to 0x1F)**

Field	Bits	Length	Initial Value
PHASE_ASSIGN	7:0	8	0

The SPI-4 ingress manual alignment phase/result registers at Block\_base 0x0800 have read and write access. A SPI-4 ingress manual alignment phase/result register is used to manually align the phase of the data lane, control lane, status lanes, and a test lane corresponding to its register in turn and is intended for diagnostics only and is not needed in normal operation. If the FORCE field of Table 99, SPI-4 ingress bit alignment control register (register\_offset 0x11) is set to a logic one, manual phase alignment is enabled. If the FORCE field is set to a logic zero, normal automatic phase alignment is enabled, and the result can be viewed here. There are five center taps to choose from, plus two guard taps on either side of the center, per data bit sampled. The oldest data sample is at tap 8 ("right"), while the newest data sample is at tap 0 ("left"). Taps 0 and 1 are the left margin taps for tracking purposes, while taps 7 and 8 are the right margin taps. A tap between 2 to 7 is initially selected in automatic mode. See Figure 7-Data sampling diagram. Register 0x0C is used for lane DATA0.

**PHASE\_ASSIGN [3:0]**

Used for selecting the bit phase corresponding to the rising clock edge of I\_DCLK. The four bits number the phases from 0 to 8, relative to the positively-clocked bit.

**PHASE\_ASSIGN [7:4]**

Used for selecting the bit phase corresponding to the falling clock edge of I\_DCLK. The four bits number the phases from 0 to 8, relative to the negatively-clocked bit.

**SPI-4 egress data lane timing register (Block\_base 0x0800 + Register\_offset 0x2A)**

**TABLE 114 - SPI-4 EGRESS DATA LANE TIMING REGISTER (REGISTER\_OFFSET 0x2A)**

Field	Bits	Length	Initial Value
DTC0[1:0]	1:0	2	0
DTC1[1:0]	3:2	2	0
...	..	2	0
DTC15[1:0]	31:30	2	0

The SPI-4 egress data lane timing register at Block\_base 0x0800 + Register\_offset 0x2A has read and write access. The SPI-4 egress data lane

timing register is used to manually align the phase of data lane n by adding from 0.1 clock cycle to 0.3 clock cycles of delay.

**DTCn [1:0]** Used for adding 0.1 clock cycle units of output delay to SPI-4 egress data lane n.

[1:0]=0=No added delay

[1:0]=1=Add 0.1 clock cycle of delay to data lane n

[1:0]=2=Add 0.2 clock cycles of delay to data lane n

[1:0]=3=Add 0.3 clock cycles of delay to data lane n

### **SPI-4 egress control lane timing register (Block\_base 0x0800 + Register\_offset 0x2B)**

**TABLE 115 - SPI-4 EGRESS CONTROL LANE TIMING REGISTER (REGISTER\_OFFSET 0x2B)**

Field	Bits	Length	Initial Value
CTLTC[1:0]	1:0	2	0

The SPI-4 egress control lane timing register at Block\_base 0x0800 has read and write access. The SPI-4 egress control lane timing register is used to manually align the phase of the control lane by adding from 0.1 clock cycle to 0.3 clock cycles of delay.

**CTLTC [1:0]** Used for adding 0.1 clock cycle units of output delay to the SPI-4 egress control output.

[1:0]=0=No added delay

[1:0]=1=Add 0.1 clock cycle of delay to the control output

[1:0]=2=Add 0.2 clock cycles of delay to the control output

[1:0]=3=Add 0.3 clock cycles of delay to the control output

### **SPI-4 egress data clock timing register (Block\_base 0x0800 + Register\_offset 0x2C)**

**TABLE 116 - SPI-4 EGRESS DATA CLOCK TIMING REGISTER (REGISTER\_OFFSET 0x2C)**

Field	Bits	Length	Initial Value
DCTC[3:0]	3:0	4	0

The SPI-4 egress data clock timing control register at Block\_base 0x0800 has read and write access. The SPI-4 egress data clock timing control register is used to manually align the phase of the SPI-4 egress data clock to the data and control lanes by adding from 0.1 clock cycle to 0.9 clock cycles of delay to the data clock output. Note that tap selection is not monotonic with the number in bit field [3:0].

**DCTC [3:0]** Used for adding 0.1 clock cycle units of output delay to the SPI-4 egress data clock.

[3:0]=0=No added delay

[3:0]=1=Add 0.1 clock cycle of delay to the SPI-4 egress data clock

[3:0]=3=Add 0.2 clock cycles of delay to the SPI-4 egress data clock

[3:0]=2=Add 0.3 clock cycles of delay to the SPI-4 egress data clock

[3:0]=7=Add 0.4 clock cycles of delay to the SPI-4 egress data clock

[3:0]=6=Add 0.5 clock cycles of delay to the SPI-4 egress data clock

[3:0]=4=Add 0.6 clock cycles of delay to the SPI-4 egress data clock

[3:0]=5=Add 0.7 clock cycles of delay to the SPI-4 egress data clock

[3:0]=F=Add 0.8 clock cycles of delay to the SPI-4 egress data clock

[3:0]=E=Add 0.9 clock cycles of delay to the SPI-4 egress data clock

### **SPI-4 egress status timing register (Block\_base 0x0800 + Register\_offset 0x2D)**

**TABLE 117 - SPI-4 EGRESS STATUS TIMING REGISTER (REGISTER\_OFFSET 0x2D)**

Field	Bits	Length	Initial Value
STC0[1:0]	1:0	2	0
STC1[1:0]	3:2	2	0

The SPI-4 egress status timing register at Block\_base 0x0800+Register\_offset 0x2D has read and write access. The SPI-4 egress status timing register is used to manually align the phase of the status lane n by adding from 0.1 clock cycle to 0.3 clock cycles of delay. The STC0[1:0] and STC0[1:0] fields are valid only for LVDS status, not for LVTTTL status.

**STCn [1:0]** Used for adding 0.1 clock cycle units of output delay to SPI-4 egress status lane n.

[1:0]=0=No added delay

[1:0]=1=Add 0.1 clock cycle of delay to status lane n

[1:0]=2=Add 0.2 clock cycles of delay to status lane n

[1:0]=3=Add 0.3 clock cycles of delay to status lane n

### **SPI-4 egress status clock timing register (Block\_base 0x0800 + Register\_offset 0x2E)**

**TABLE 118 - SPI-4 EGRESS STATUS CLOCK TIMING REGISTER (REGISTER\_OFFSET 0x2E)**

Field	Bits	Length	Initial Value
SCTC[3:0]	3:0	4	0

The SPI-4 egress status clock timing register at Block\_base 0x0800 + Register\_offset 0x2E has read and write access. The SPI-4 egress status clock timing register is used to manually align the phase of the SPI-4 egress status clock to the status outputs by adding from 0.1 clock cycle to 0.9 clock cycles of delay to the status clock output. Note that tap selection is not monotonic with the number in bit field [3:0]. The SCTC[3:0] field is valid only for LVDS status, not for LVTTTL status.

**SCTC [3:0]** Used for adding 0.1 unit intervals of output delay to the SPI-4 egress status clock output.

[3:0]=0=No added delay

[3:0]=1=Add 0.1 clock cycle of delay to the SPI-4 egress status clock

[3:0]=3=Add 0.2 clock cycles of delay to the SPI-4 egress status clock

[3:0]=2=Add 0.3 clock cycles of delay to the SPI-4 egress status clock

[3:0]=7=Add 0.4 clock cycles of delay to the SPI-4 egress status clock

[3:0]=6=Add 0.5 clock cycles of delay to the SPI-4 egress status clock

[3:0]=4=Add 0.6 clock cycles of delay to the SPI-4 egress status clock

[3:0]=5=Add 0.7 clock cycles of delay to the SPI-4 egress status clock

[3:0]=F=Add 0.8 clock cycles of delay to the SPI-4 egress status clock

[3:0]=E=Add 0.9 clock cycles of delay to the SPI-4 egress status clock

9.4.10 Common module block base 0x0900 registers

PMON timebase control register (Block\_base 0x0900 + Register\_offset 0x00)

TABLE 119 - PMON TIMEBASE CONTROL REGISTER (REGISTER\_OFFSET 0x00)

Field	Bits	Length	Initial Value
INTERNAL	0	1	0b0
TIMER	1	1	0b0
MANUAL	2	1	0b0

A single PMON timebase module is available in the IDT88P8344. The PMON timebase module directs a timebase event to all PMON modules in the device. The timebase period can be internally or externally generated. The selection is made by the INTERNAL flag in the PMON update control register. A snapshot of the counters is taken when the timebase expires and the counters are cleared. The PMON update control register is at common module 0x8000 + Block\_base 0x0900 + Register\_offset 0x00 = 0x8900 and has read and write access.

**INTERNAL** Selects between internal or external timebases for performance monitoring. The internal timebase is either generated by the internal processor or by a free running timer. The selection is made by the TIMER flag in the PMON update control register. When the time interval expires, the TIMEBASE pin is asserted for sixteen MCLK cycles. The timebase event is captured by the timebase status in the support interrupt status register.

0= External timebase from the TIMEBASE pin is selected. The externally generated timebase signal is applied to the TIMEBASE pin. A positive edge detector generates the timebase event.

1=Internal timebase is selected. When the time interval expires, the TIMEBASE pin is driven high for sixteen MCLK cycles.

**TIMER** Selects between the internal free-running timebase or a microprocessor-controlled write to generate the timebase event. The TIMER field is valid only when the INTERNAL field is a logic one.

- 0=Selects the microprocessor generated timebase
- 1=Selects the internal free-running timebase

**MANUAL** The microprocessor generates an internal timebase event by a write access with a logical one to the MANUAL flag in the PMON Update Control Register if the microprocessor timebase is selected. The MANUAL bit is self-clearing. The MANUAL field is only valid if the TIMER field is a logic zero.

- 0=No operation
- 1=A timebase event is generated

Timebase register (Block\_base 0x0900 + Register\_offset 0x01)

TABLE 120 - TIMEBASE REGISTER (REGISTER\_OFFSET 0x01)

Field	Bits	Length	Initial Value
PERIOD	26:0	27	0x4A2 8600

The timebase register is at Block\_base 0x0900 + Register\_offset 0x01 and has read and write access.

The timebase period for free-running timers is configured by the PERIOD field in the timebase register. The PERIOD field specifies the number of MCLK clock cycles required for a single event. The PERIOD field is only valid if both the INTERNAL and TIMER fields are a logic one.

Clock generator control register (Block\_base 0x0900 + Register\_offset 0x010)

TABLE 121 - CLOCK GENERATOR CONTROL REGISTER (REGISTER\_OFFSET 0x10)

Field	Bits	Length	Initial Value
OCLK0_EN	0	1	0b01
N_OCLK0	2:1	2	0b11
Reserved	3	1	0b0
OCLK1_EN	4	1	0b1
N_OCLK1	6:5	2	0b11
Reserved	7	1	0b0
OCLK2_EN	8	1	0b1
N_OCLK2	10:9	2	0b11
Reserved	11	1	0b0
OCLK3_EN	12	1	0b1
N_OCLK3	14:13	2	0b11
Reserved	16:15	2	0b0
N_MCLK	18:17	2	0b11
Reserved	31:19	13	0

The clock generator control register is at common module Block\_base 0x0900 + Register\_offset 0x010.

The clock generator provides four clock outputs on the OCLK[3:0] pins, MCLK for internal use, and SPI-4 data and FIFO status channel egress clocks. The OCLK[3:0] clock frequencies can be selected independently of each other. OCLK[3:0] outputs can be used as SPI-3 clock sources. The OCLK[3:0] pins are separately enabled by setting each associated enable flag in Table 121 - Clock generator control register (Register\_offset 0x10). When an OCLK[3:0] output is not enabled, it is in a logic low state. MCLK is the internal processing clock, and is always enabled. Refer to Table 122 - OCLK and MCLK frequency select encoding, for selecting the frequencies of MCLK and OCLKs.

During either a hardware or a software reset, the OCLK[3:0] pins are all logic low. Immediately following reset, all OCLK[3:0] outputs are active with the output frequency defined by pll\_ockl divided by the initial value in the Table 121 - Clock generator control register (Block\_base 0x0900 + Register\_offset 0x10).

The clock generator control register at indirect address 0x8910 has read and write access. The clock generator control register is used to set the frequency of MCLK and the OCLK outputs, as well as to enable the OCLK outputs. Note that divider values should be chosen so that OCLK[3:0] and MCLK are within their specified operating range provided in Table 136, OCLK[3:0] clock outputs and MCLK internal clock.

**OCLK[k]\_EN** Used for enabling the kth OCLK output  
 0=OCLK[k] is not enabled and OCLK[k] is at a logic zero  
 1=OCLK[k] is enabled and active

**N\_OCLK[k][1:0]** Select the OCLK[k] frequency according to Table 122- OCLK and MCLK frequency select encoding.

**N\_MCLK[k]** Select the MCLK frequency according to Table 122-OCLK and MCLK frequency select encoding.

TABLE 122 - OCLK AND MCLK FREQUENCY SELECT ENCODING

N_MCLK & N_OCLK[k] Frequency Selects	Frequency
00	pll_ockl / 4
01	pll_ockl / 6
10	pll_ockl / 8
11	pll_ockl / 10

### General purpose I/O (Block\_base 0x0900 + Register\_offset 0x20)

**TABLE 123 - GPIO REGISTER (REGISTER\_OFFSET 0x20)**

Field	Bits	Length	Initial Value
DIR_OUT	4:0	5	0x00
Reserved	7:5	3	0x0
LEVEL	12:8	5	0x00
Reserved	15:13	3	0x0
MONITOR_EN	20:16	5	0x00

Five general purpose I/O pins are provided. Each pin I/O direction is controlled by the DIR\_OUT field in the GPIO register. The logical level on a GPIO pin is controlled by the LEVEL field in the GPIO register if DIR\_OUT=1 (pin=output), or sensed if DIR\_OUT=0 (pin=input). Optionally, the LEVEL bit can monitor the logic level of any bit selected from the indirect access space if MONITOR\_EN is set high. With MONITOR\_EN set high, bits in the indirect access space can be selected for monitoring by the ADDRESS and BIT fields in the GPIO monitor table.

The general purpose I/O registers are at common module Block\_base 0x0900 and have read and write access.

**DIR\_OUT[4:0]** Used for configuring each GPIO pin as either an input or an output

- 0=GPIO pin is an input
- 1=GPIO pin is an output

**LEVEL[4:0]** Used for sensing or driving each GPIO pin

- 0=GPIO pin is sensed as a logic zero if an input, or driven to a logic zero if an output
- 1=GPIO pin is sensed as a logic one if an input, or driven to a logic one if an output

**MONITOR\_EN [4:0]** Used for enabling the monitor output function for each GPIO pin. GPIO pins used as monitors must also be configured to be outputs. All GPIO pins must be used as either monitors or as normal I/O; no mixing of the monitoring function and the normal I/O function is permitted.

- 0=GPIO pin is used as an I/O pin
- 1=GPIO pin is used as a monitor pin

### GPIO monitor table (Block\_base 0x0900 + Register\_offset 0x21 - 0x25)

**TABLE 124 - GPIO MONITOR TABLE (5 ENTRIES 0x21-0x25 FOR GPIO[0] THROUGH GPIO[4])**

Field	Bits	Length	Initial Value
ADDRESS	15:0	16	0x0000
Reserved	23:16	8	0x00
BIT	28:24	5	0x00
Reserved	31:29	3	0x0

A bit in the indirect access space can be selected for monitoring by the ADDRESS and BIT fields in the GPIO monitor table.

The GPIO Monitor Table for GPIO[0] is at Common\_Module 0x8000+ Block\_base 0x0900 + Register\_offset 0x21 = 0x8921.

The GPIO Monitor Table for GPIO[1] is at Common\_Module 0x8000+ Block\_base 0x0900 + Register\_offset 0x22 = 0x8922.

The GPIO Monitor Table for GPIO[2] is at Common\_Module 0x8000+ Block\_base 0x0900 + Register\_offset 0x23 = 0x8923.

The GPIO Monitor Table for GPIO[3] is at Common\_Module 0x8000+ Block\_base 0x0900 + Register\_offset 0x24 = 0x8924.

The GPIO Monitor Table for GPIO[4] is at Common\_Module 0x8000+ Block\_base 0x0900 + Register\_offset 0x25 = 0x8925.

**ADDRESS[15:0]** Used for configuring the indirect address select when the GPIO pins are put into monitor mode.

**BIT[4:0]** Used for selecting the register bit (1 of 32) for a GPIO put into monitor mode.

BIT[4:0]=0x00 selects data bit 0.

...

BIT[4:0]=0x1F selects data bit 31.

### Version number register (common module Block\_base 0x0900 + Register\_offset 0x30)

**TABLE 125 - VERSION NUMBER REGISTER (REGISTER\_OFFSET 0x30)**

Field	Bits	Length	Initial Value
Version	7:0	8	0x01
ID	15:8	8	0xF9

The version number register is a read-only sixteen-bit register at Common\_module 0x8000 + Block\_base 0x0900 + Register\_offset 0x30 = 0x8930 in the indirect register access space. The version number register contains hard-coded values that can be read to verify the microprocessor read path is correct, and that the correct part is installed.

**VERSION** The hardware version is read from this field.

**ID** The hardware identification is read from this field.

## 10. JTAG INTERFACE

The device supports the optional TRST input signal. It supports a TCK clock frequency up to 10MHz.

**TABLE 126 – JTAG INSTRUCTIONS**

Code	Instruction	Function
000	EXTEST	JTAG function
001	IDCODE	JTAG function
010	SAMPLE	JTAG function
011	CLAMP	JTAG function
100	HIGHZ	JTAG function
101	Private	Private function
111	BYPASS	JTAG function

Version field equals 0

JTAG ID #0x044F MANUFACTURER ID 0x033 LAST BIT IS 1.

## 11. ELECTRICAL AND THERMAL SPECIFICATIONS

### 11.1 Absolute maximum ratings

**TABLE 127 – ABSOLUTE MAXIMUM RATINGS**

Parameter	Symbol	Conditions	Min. <sup>(1)</sup>	Max. <sup>(1)</sup>	Unit
Core Digital Supply Voltage	VDDC18	V <sub>SS</sub> =0, AV <sub>SS</sub> =0, T <sub>j</sub> =25°C	-0.3	2.2	V
I/O Digital Supply Voltage	VDDT33		-0.3	4.6	V
Analog Supply Voltage	VDDA18		-0.3	3.6	V
Analog Supply Voltage	VDDA33		-0.3	3.6	V
I/O Input Voltage for CMOS	V <sub>inL</sub>		-0.5	6.0	V
I/O Input Voltage for LVTTTL	V <sub>inL</sub>		-0.5	6.0	V
I/O Output Voltage	V <sub>out</sub>		-0.5	4.6	V
Latch-up Current	I <sub>o</sub>		-	100	mA
ESD Performance (HBM)			-	2000	V
Ambient Operating Temperature	T <sub>a</sub> (Industrial)		-40	+85	°C
Ambient Operating Temperature	T <sub>a</sub> (Commercial)		0	+70	°C
Storage Temperature	T <sub>s</sub>		-65	+150	°C

**NOTE:**

1. Functional and tested operating conditions are given in Table Absolute Maximum Ratings are stress ratings only, and functional operation at the maximum is not guaranteed. Stresses beyond those listed may affect device reliability or cause permanent damage to the device.

### 11.2 Recommended Operating Conditions

**TABLE 128 – RECOMMENDED OPERATING CONDITIONS**

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Core Digital Supply Voltage	VDDC18	V <sub>SS</sub> =0	1.68	1.8	1.96	V
I/O Digital Supply Voltage	VDDT33	V <sub>SS</sub> =0	3.0	3.3	3.6	V
Analog Supply Voltage	VDDA18	V <sub>SS</sub> =0	1.68	1.8	1.96	V
Analog Supply Voltage	VDDA33	V <sub>SS</sub> =0	3.0	3.3	3.6	V
Ambient Operating Temperature	T <sub>a</sub> (Industrial)		-40	25	+85	°C
Ambient Operating Temperature	T <sub>a</sub> (Commercial)		0	25	+70	°C
Junction Temperature	T <sub>J</sub>		-	-	+110	°C

## 11.3 Terminal Capacitance

**TABLE 129 – TERMINAL CAPACITANCE**

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Capacitance	C <sub>I</sub>	Measured at V <sub>in</sub> =V <sub>out</sub> =V <sub>SS</sub> T <sub>a</sub> =25°C	-	-	8	pF
Load Capacitance	C <sub>O</sub>		-	-	20	pF
Load Capacitance for OCLK [3:0] signals	C <sub>O</sub>		-	-	30	pF
Load Capacitance for microprocessor interface	C <sub>O</sub>		-	-	100	pF

## 11.4 Thermal Characteristics

**TABLE 130 – THERMAL CHARACTERISTICS**

Parameter	Symbol	Conditions	Value
Typical Power Dissipation total	P <sub>T</sub>	T <sub>a</sub> =25°C	4.0W
Typical Power Dissipation from 1.8V	P <sub>VDD18</sub>	T <sub>a</sub> =25°C	2.1W
Typical Power Dissipation from 3.3V	P <sub>VDD33</sub>	T <sub>a</sub> =25°C	1.9W
Thermal Resistance (Junction to case)	Θ <sub>JC</sub>		4.5°C/W
Thermal Resistance (Junction to board)	Θ <sub>JB</sub>		4.1°C/W
Thermal Resistance (Ambient)	Θ <sub>JA</sub>	Airflow 0.0m/s Airflow 1.0m/s Airflow 2.0m/s	15.4°C/W 11.7°C/W 10.2°C/W



## 11.5 DC Electrical characteristics

TABLE 131 – DC ELECTRICAL CHARACTERISTICS

Parameter	Description	Min.	Typ.	Max.	Unit	Test Conditions
<b>CMOS I/O</b>						
V <sub>IL</sub>	Input Low Voltage	-0.3	—	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	—	V <sub>D33</sub> +0.3	V	
V <sub>OL</sub>	Output Low Voltage	V <sub>SS</sub>	—	0.4	V	V <sub>D33</sub> =min, I <sub>OL</sub> = max_load (Note 1)
V <sub>OH</sub>	Output High Voltage	2.4	—	V <sub>D33</sub>	V	V <sub>D33</sub> =min, I <sub>OH</sub> = max_load (Note 1)
V <sub>RESETBH</sub>	RESETB Reset Input High Voltage	1.47	—	—	V	
V <sub>RESETBL</sub>	RESETB Reset Input Low Voltage	—	—	0.95	V	
V <sub>RESETBHYST</sub>	RESETB Reset Input Hysteresis Voltage	0.53	—	—	V	
I <sub>ILPU</sub>	Input Low Current (with pullups)	-100	-50	-4	μA	V <sub>IL</sub> =V <sub>SS</sub>
I <sub>IHPU</sub>	Input High Current (with pullups)	-10	0	+10	μA	V <sub>IL</sub> =V <sub>D33</sub>
I <sub>IL</sub>	Input Low Current (without pullups)	-10	0	+10	μA	V <sub>IL</sub> =V <sub>SS</sub>
I <sub>IH</sub>	Input High Current (without pullups)	-10	0	+10	μA	V <sub>IH</sub> =V <sub>D33</sub>
I <sub>OZ</sub>	Offstate output current	-10	0	+10	μA	V <sub>D33</sub> =MAX
<b>LVTTL I/O</b>						
V <sub>IL</sub>	Input Low Voltage	-0.3	—	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	—	V <sub>D33</sub> +0.3	V	
V <sub>OL</sub>	Output Low Voltage	V <sub>SS</sub>	—	0.4	V	V <sub>D33</sub> =MIN, I <sub>OL</sub> = 8mA
V <sub>OH</sub>	Output High Voltage	2.4	—	V <sub>D33</sub>	V	V <sub>D33</sub> =MIN, I <sub>OH</sub> = -8mA
I <sub>L</sub>	Input Current	-5	0	+5	μA	V <sub>D33</sub> =MAX
I <sub>OZ</sub>	Offstate output current	-10	0	+10	μA	V <sub>D33</sub> =MAX
<b>SPI-4 LVDS I/O</b>						
<b>Input Characteristics</b>						
V <sub>IN</sub>	Input Voltage Range, V <sub>P</sub> or V <sub>N</sub>	0	—	2400	mV	V <sub>GGP</sub>   < 925 mV
V <sub>IDTH</sub>	Differential Voltage Required to Toggle Input	100	—	—	mV	V <sub>GGP</sub>   < 925 mV
V <sub>HYST</sub>	Input Differential Hysteresis, V <sub>IDTHH</sub> - V <sub>IDTHL</sub>	25	—	—	mV	
R <sub>IN</sub>	Input Differential Impedance	90	100	110	Ohms	P to N input
<b>Output Characteristics</b>						
V <sub>OL</sub>	Output Low Voltage, V <sub>P</sub> or V <sub>N</sub>	925	—	—	mV	R <sub>DIFF_TERM</sub> = 100
V <sub>OH</sub>	Output High Voltage, V <sub>P</sub> or V <sub>N</sub>	—	—	1475	mV	R <sub>DIFF_TERM</sub> = 100
V <sub>OS</sub>	Output Offset Voltage	1125	—	1375	mV	R <sub>DIFF_TERM</sub> = 100
delta V <sub>OS</sub>	Change in V <sub>OS</sub> between "0" and "1"	—	—	25	mV	R <sub>DIFF_TERM</sub> = 100
V <sub>OD</sub>	Output Differential Voltage	250	—	450	mV	R <sub>DIFF_TERM</sub> = 100
delta V <sub>OD</sub>	Change in  V <sub>OD</sub>   between "0" and "1"	50 (DC)	—	150 (AC)	mV	R <sub>DIFF_TERM</sub> = 100
R <sub>o</sub>	Output Single-ended Impedance	40	100	140	Ohms	
delta R <sub>o</sub>	R <sub>o</sub> mismatch between P and N	—	—	10	%	
I <sub>SP</sub> , I <sub>SN</sub>	Output Short Circuit Current	—	—	40	mA	P or N output shorted to V <sub>SS</sub>
I <sub>SPN</sub>	Output Short Circuit Current	—	—	12	mA	P and N outputs shorted together
I <sub>xP</sub>  ,  I <sub>xN</sub>	Power-off output leakage	—	—	10	μA	V <sub>D33</sub> = V <sub>SS</sub>

**NOTE:**

1. Maximum load = 8 mA for microprocessor data bus DBUS[7:0]; maximum load= 4 mA for all other CMOS outputs.

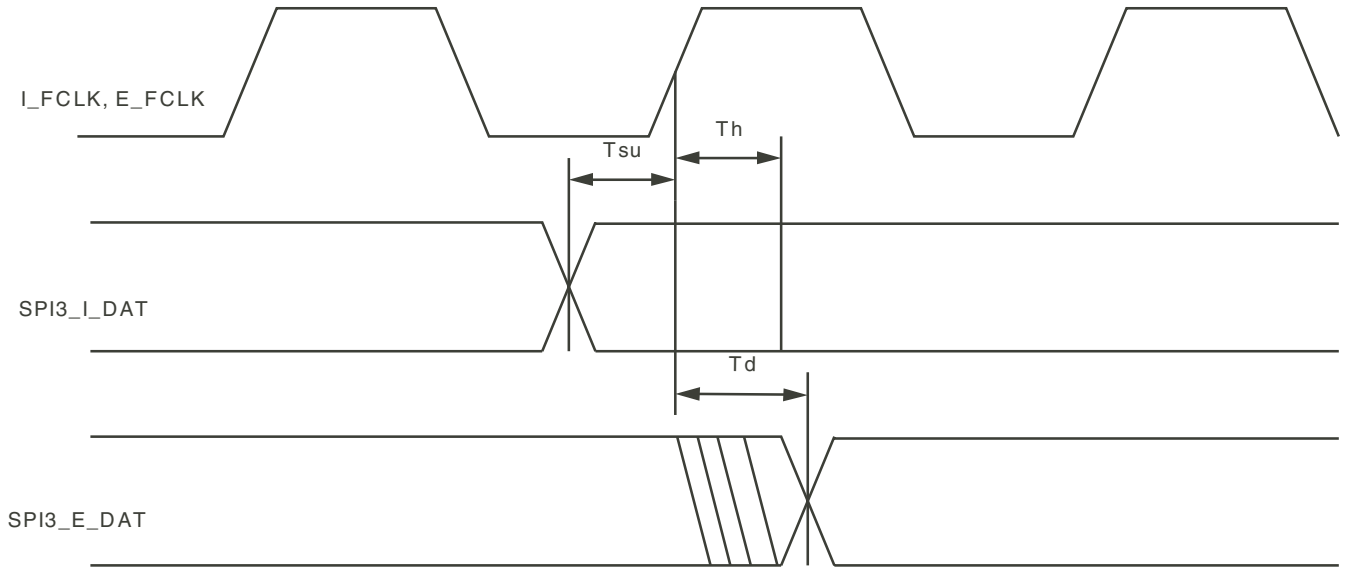
**11.6 AC characteristics**

**11.6.1 SPI-3 I/O timing**

Refer to [SPI-3 in Glossary] for logical timing diagrams of the SPI-3 and SPI-4 interfaces. Note that underclocking and overclocking for the SPI-4 and SPI-3 interfaces is supported.

**SPI-3 Input / Output**

SPI-3 input and output timing is shown in the following paragraph.



6370 drw05

Figure 35. SPI-3 I/O timing diagram

**TABLE 132 – SPI-3 AC INPUT / OUTPUT TIMING SPECIFICATIONS**

I_FCLK, E_FCLK	Unit	Min.	Typ.	Max.	Description
Duty cycle	%	30	—	70	Input SPI-3 clock duty cycle
Frequency	MHz	MCLK/4	—	133	I_FCLK, E_FCLK
TR, TF	ns	—	—	2	Rise fall time ( 20%, 80% )
<b>All outputs</b>					
TD	ns	2.33	—	5.65	Output delay after E_FCLK
TR, TF	ns	—	—	2	Rise fall time ( 20%, 80% )
<b>All inputs</b>					
TSU	ns	1	—	—	Input setup before I_FCLK
TH	ns	0.65	—	—	Input hold after I_FCLK

### 11.6.2 SPI-4 LVDS Input / Output

SPI-4 input and output timing is shown in the following paragraph. Double Data Rate protocol is used for data and status transfer. The SPI-4 LVDS signals use a dynamic data alignment at the ingress.

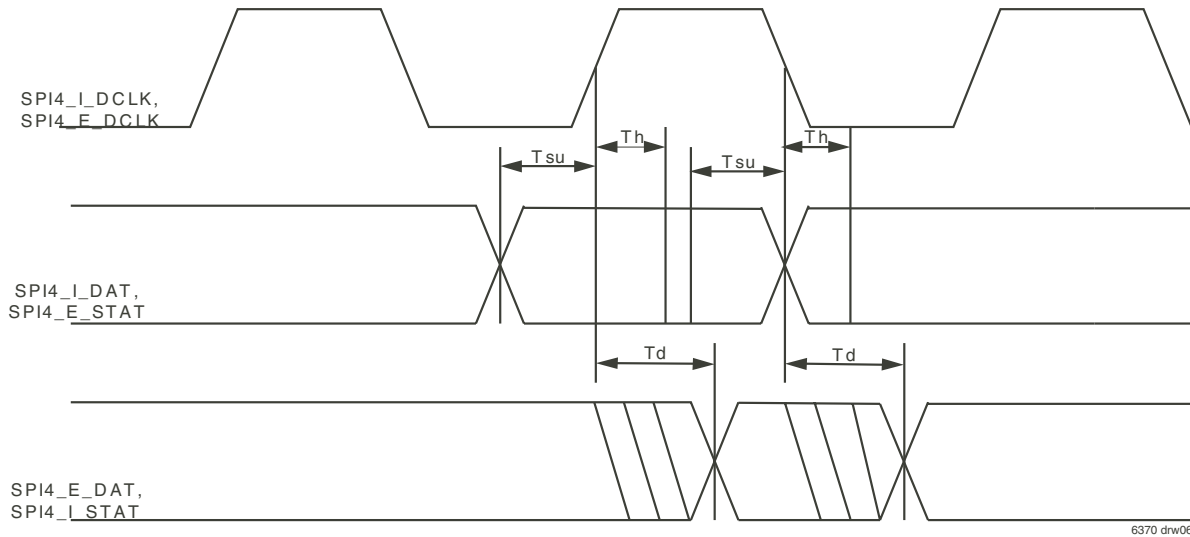


Figure 36. SPI-4 I/O timing diagram

**TABLE 133 – SPI-4.2 LVDS AC INPUT / OUTPUT TIMING SPECIFICATIONS**

Inputs	Unit	Min.	Typ.	Max.	Description
Duty cycle	%	45	50	55	I_DCLK ingress clock duty cycle
Frequency (DDR)	MHz	80	—	200	Ingress clock frequency, I_LOW=1
Frequency (DDR)	MHz	200	311	400	Ingress clock frequency, I_LOW=0
TR, TF	ps	300	—	500	Input rise or fall time ( 20%, 80% )
Deskew	UI	—	—	+/- 1	Bit line deskew
<b>Outputs</b>					
Duty cycle	%	45	50	55	E_DCLK Egress clock duty cycle
Frequency (DDR)	MHz	80	—	200	Egress clock frequency, E_LOW=1
Frequency (DDR)	MHz	200	311	400	Egress clock frequency, E_LOW=0
TR, TF	ps	300	—	500	Output rise or fall time ( 20%, 80% )
Tskew	ps	—	—	50	Output differential skew, P to N
SYNTH Jitter	UI	—	—	0.1	PLL jitter as a fraction of the clock cycle
TD	ns	—	—	—	Adjustable

**11.6.3 SPI-4 LVTTTL Status AC characteristics****TABLE 134 – SPI-4 LVTTTL STATUS AC CHARACTERISTICS**

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
<b>SPI-4 LVTTTL Status<sup>(1)</sup></b>						
STAT_T[1:0] to SCLK_T setup time	T <sub>SU</sub>		2			ns
SCLK_T to STAT_T [1:0] hold time	T <sub>H</sub>		0.5			ns
SCLK_T to STAT_T [1:0] delay	T <sub>D</sub>			1	1.2	ns

**NOTE:**

1. For the SPI-4 LVTTTL valid, hold & setup the edge is configurable. The SPI-4 ingress LVTTTL status clock active edge is configured by I\_CLK\_EDGE field in Table 89-SPI-4 Ingress Configuration Register on page 69. The SPI-4 egress LVTTTL status clock active edge is configured by E\_CLK\_EDGE field in Table 104-SPI-4 Egress Configuration Register on page 73.

**11.6.4 REF\_CLK clock input****TABLE 135 – REF\_CLK CLOCK INPUT**

REF_CLK	Unit	Min.	Typ.	Max.	
Duty cycle	%	30	50	70	REF_CLK clock input duty cycle
FREF_CLK	MHz	12.5	19.44	25	Main reference clock input
TR, TF	ns	—	—	5	Rise fall time ( 20%, 80% )

**11.6.5 MCLK internal clock and OCLK[3:0] clock outputs****TABLE 136 – OCLK[3:0] CLOCK OUTPUTS AND MCLK INTERNAL CLOCK**

OCLK[3:0]	Unit	Min.	Typ.	Max.	Description
Duty cycle	%	45	50	55	OCLK[3:0] outputs, clock duty cycle
Frequency	MHz	40	104	133	OCLK[3:0], programmable
Output skew between OCLKs					One pll_ock cycle of deliberate skew between each OCLK[3:0]
TR, TF	ns	1		2	OCLK[3:0] rise, fall time (20%,80%)
<b>MCLK</b>					
Frequency	MHz	80	—	100	Programmable

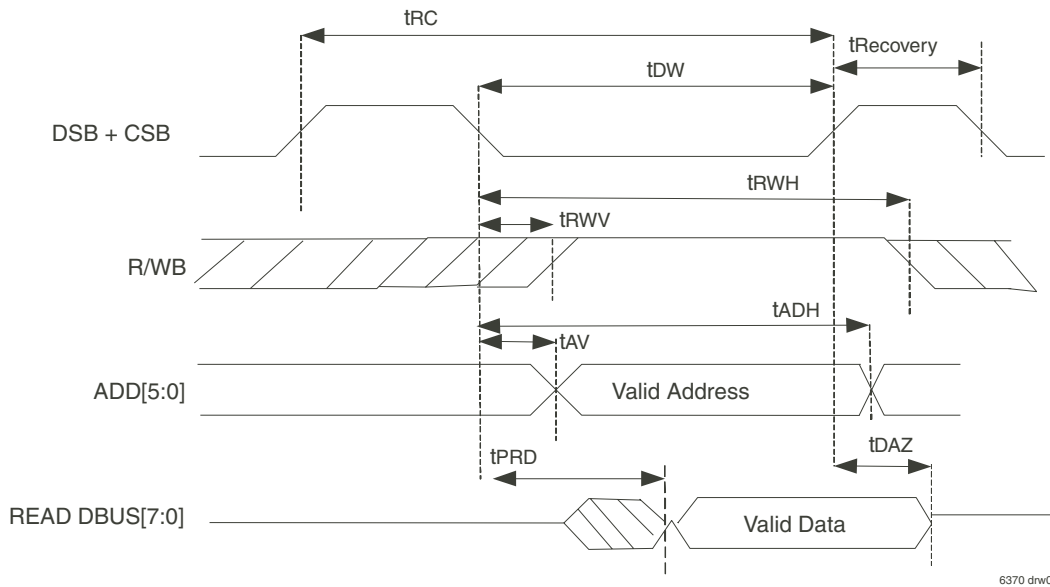
**11.6.6 Microprocessor interface****TABLE 137 – MICROPROCESSOR INTERFACE**

All outputs	Unit	Min.	Typ.	Max.	Description
TR, Tf	ns			10	Rise, fall time (20%, 80%)
<b>All inputs</b>					
TR, TF	ns			10	Rise, fall time (20%,80%)

### 11.6.6.1 Microprocessor parallel port AC timing specifications

Be sure to connect SPI\_EN to a logic low when using the parallel  $\mu$ P interface mode.

#### Read cycle specification Motorola non-multiplexed (MPM=0)

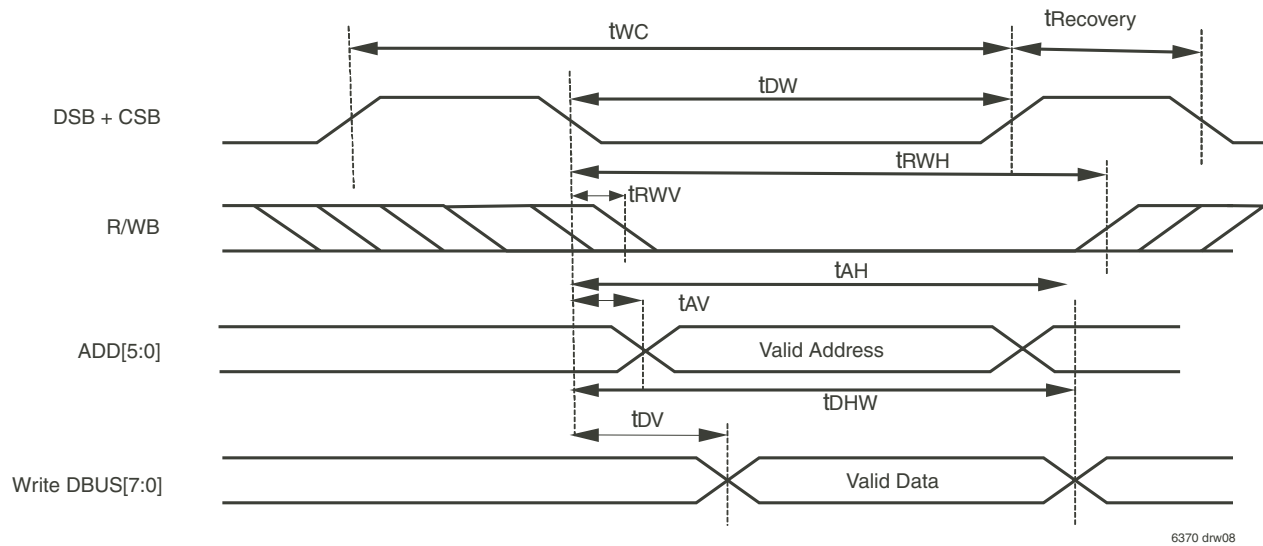


6370 drw07

Figure 37. Microprocessor parallel port Motorola read timing diagram

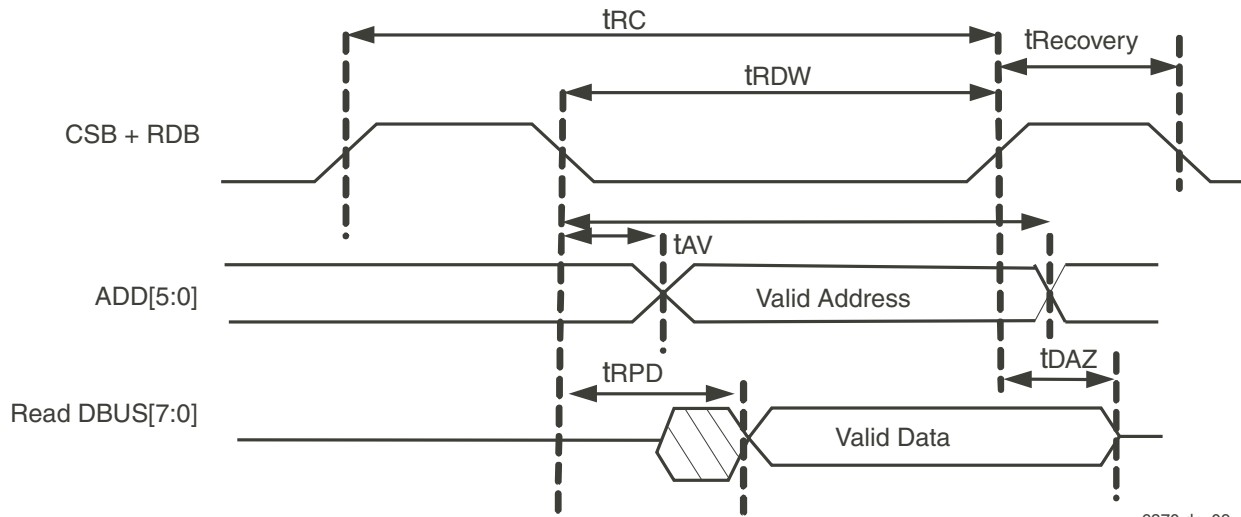
**TABLE 138 – MICROPROCESSOR PARALLEL PORT MOTOROLA READ TIMING**

Symbol	Parameter	MIN	MAX	Unit
T	Internal main clock period (MCLK)	80	100	MHz
tRC	Read cycle time	$5.5T+25$		ns
tDW	Valid DSB width	$5.5T+20$		ns
tRWV	Delay from DSB to valid read signal		$T/2-4$	ns
tRWH	R/WB to DSB hold time	$2T+10$		ns
tAV	Delay from DSB to Valid Address		$T/2-4$	ns
tADH	Address to DSB hold time	$2T+10$		ns
tRPD	DSB to valid read data propagation delay		$5.5T+20$	ns
tDAZ	Delay from read data active to high Z		12	ns
tRecovery	Recovery time from read cycle	5		ns

**Write cycle specification Motorola non-multiplexed (MPM=0)****Figure 38. Microprocessor parallel port Motorola write timing diagram****TABLE 139 – MICROPROCESSOR PARALLEL PORT MOTOROLA WRITE TIMING**

Symbol	Parameter	MIN	MAX	Unit
T	Internal main clock period (MCLK)	80	100	MHz
$t_{WC}$	Write cycle time	$2.5T+17$		ns
$t_{DW}$	Valid DSB width	$2.5T+12$		ns
$t_{RWW}$	Delay from DSB to valid write signal		$T/2-4$	ns
$t_{RWH}$	R/WB to DSB hold time	$2.5T+12$		ns
$t_{AV}$	Delay from DSB to Valid Address		$T/2-4$	ns
$t_{AH}$	Address to DSB hold time	$2.5T+12$		ns
$t_{DV}$	Delay from DSB to valid write data		$T/2-4$	ns
$t_{DHW}$	Write data to DSB hold time	$2.5T+12$		ns
$t_{Recovery}$	Recovery time from write cycle	5		ns

**Read cycle specification Intel non-multiplexed bus (MPM=1)**



6370 drw09

**NOTE:**

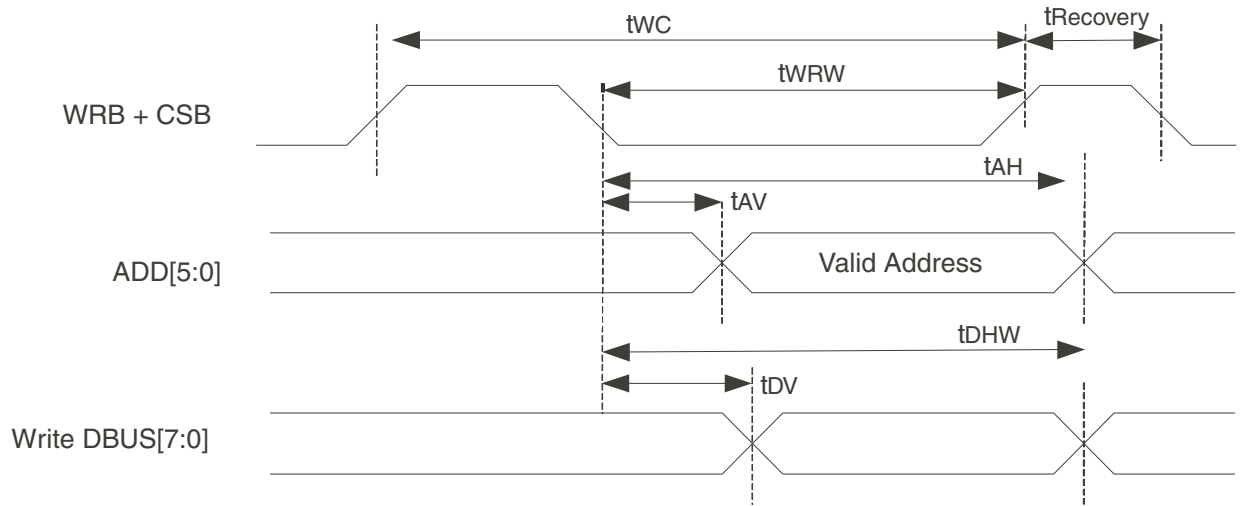
1. WRB should be tied to High.

**Figure 39. Microprocessor parallel port Intel mode read timing diagram**

**TABLE 140 – MICROPROCESSOR PARALLEL PORT INTEL MODE READ TIMING**

Symbol	Parameter	MIN	MAX	Unit
T	Internal main clock period (MCLK)	80	100	MHz
$t_{RC}$	Read cycle time	$5.5T+25$		ns
$t_{RDW}$	Valid RDB width	$5.5T+20$		ns
$t_{AV}$	Delay from RDB to Valid Address		$T/2-4$	ns
$t_{AH}$	Address to RDB hold time	$2.5T+12$		ns
$t_{RPD}$	RDB to valid read data propagation delay		$5.5T-20$	ns
$t_{DAZ}$	Delay from read data active to High-Z		12	ns
$t_{Recovery}$	Recovery time from read cycle	5		ns

**Write cycle specification Intel non-multiplexed bus (MPM=1)**



6370 dnr10

**NOTE:**

1. RDB should be tied to a logic one.

**Figure 40. Microprocessor parallel port Intel mode write timing diagram**

**TABLE 141 – MICROPROCESSOR PARALLEL PORT INTEL MODE WRITE TIMING**

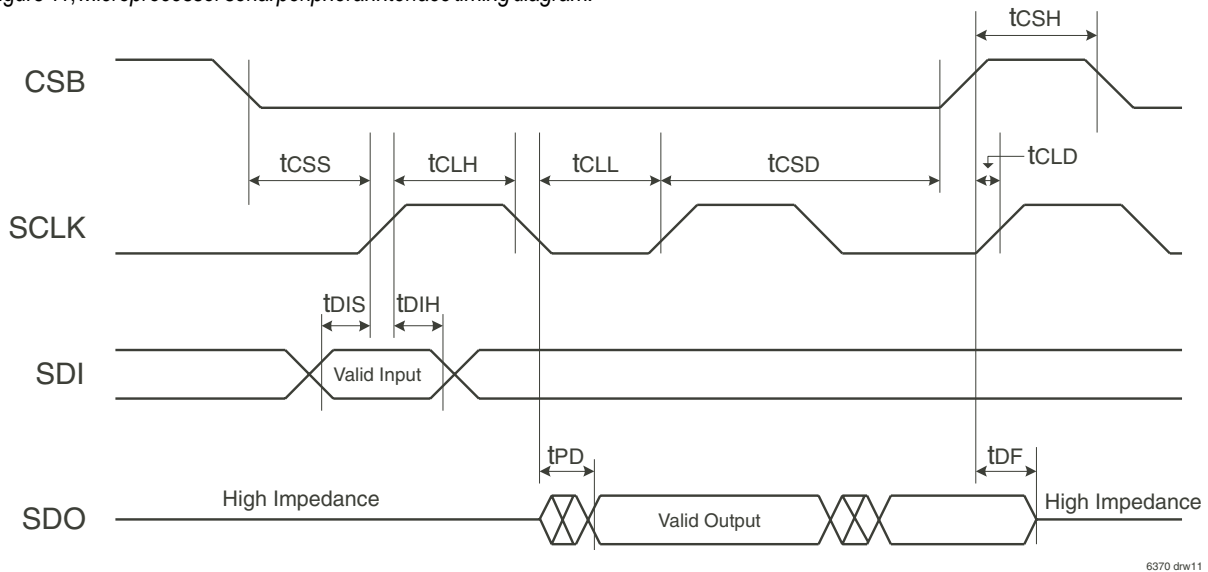
Symbol	Parameter	MIN	MAX	Unit
T	Internal main clock period (MCLK)	80	100	MHz
tWC	Write cycle time	2.5T+19		ns
tWRW	Valid WRB width	2.5T+14		ns
tAV	Delay from WRB to Valid Address		T/2-2	ns
tAH	Address to WRB hold time	2.5T+12		ns
tDV	Delay from WRB to valid write data		T/2-2	ns
tDHW	Write data to WRB hold time	2.5T+12		ns
tRecovery	Recovery time from read cycle	5		ns



### 11.6.6.2 Serial microprocessor interface (serial peripheral interface mode)

#### Timing Characteristics

The maximum SPI Data transfer clock frequency is 2 MHz. The detail information of the timing characteristics is shown in below and timing diagram is shown in Figure 41, *Microprocessor serial peripheral interface timing diagram*.



6370 drw11

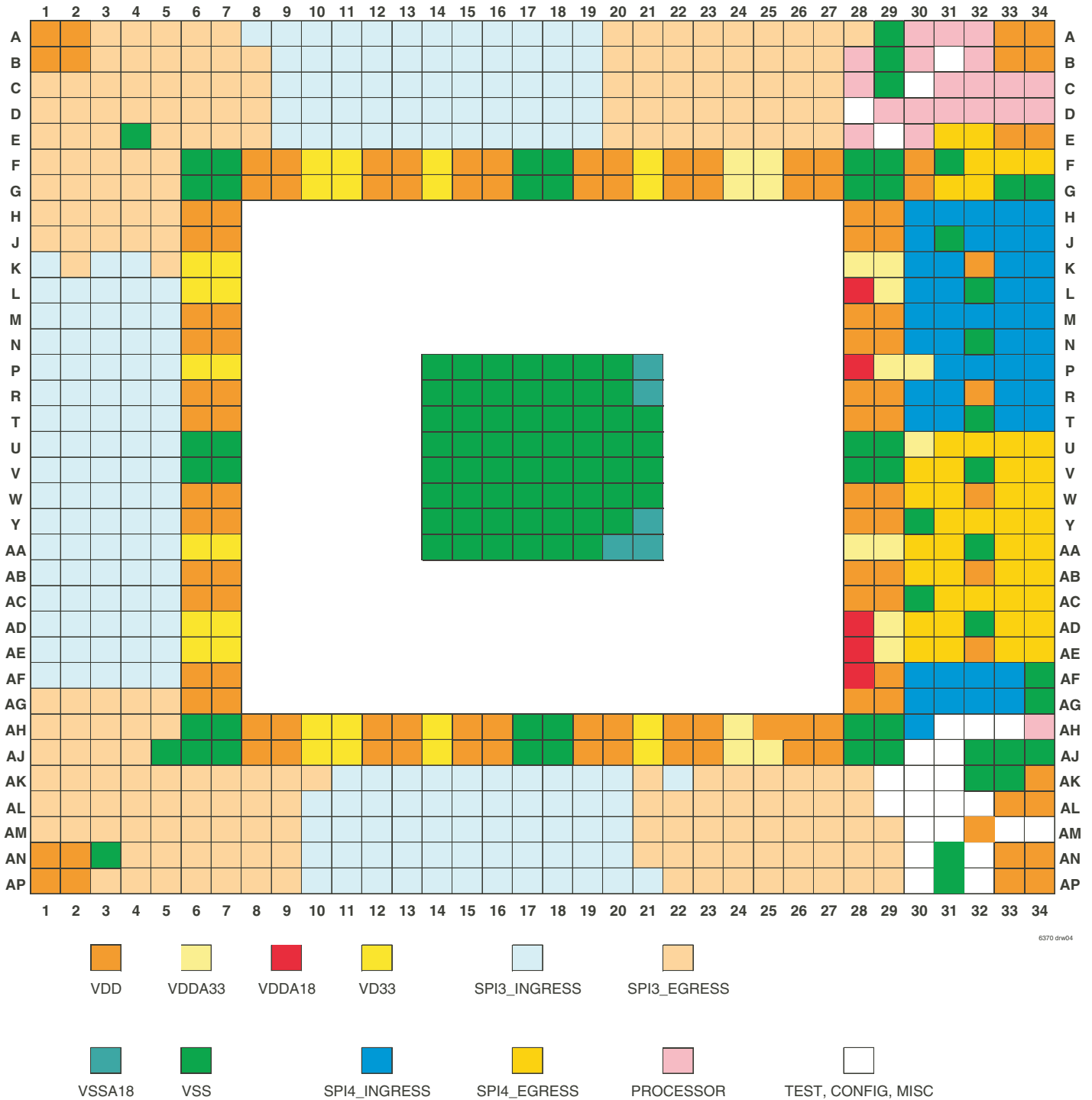
Figure 41. Microprocessor serial peripheral interface timing diagram

**TABLE 142 – MICROPROCESSOR SERIAL PERIPHERAL INTERFACE TIMING**

Symbol	Description	Min.	Max.	Unit
fOP	SCLK Frequency		2.0	MHz
tCSH	Minimum CSB High Time	100		ns
tCSS	CSB Setup Time	50		ns
tCSD	CSB Hold Time	100		ns
tCLD	SCLK Clock Disable Time	50		ns
tCLH	SCLK Clock High Time	205		ns
tCLL	SCLK Clock Low Time	205		ns
tDIS	SDI Data Setup Time	50		ns
tDIH	SDI Data Hold Time	150		ns
tPD	SDO Output Delay		150	ns
tDF	SDO Output Disable Time		50	ns

# 12. MECHANICAL CHARACTERISTICS

## 12.1 Device overview



PBGA (BH820-1, order code: BH)  
TOP VIEW

## 12.2 Pin name/ball location table

SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL
SPI3B_E_DAT[21]	A4	SPI3B_I_DAT[26]	N5	SPI3D_I_DAT[25]	AC5	SPI3D_E_DAT[20]	AM6
SPI3B_E_DAT[20]	E5	SPI3B_I_DAT[25]	N3	SPI3D_I_DAT[26]	AC4	SPI3D_E_DAT[21]	AL6
SPI3B_E_DAT[19]	B4	SPI3B_I_DAT[24]	N2	SPI3D_I_DAT[27]	AD1	SPI3D_E_DAT[22]	AP6
SPI3B_E_DAT[18]	D4	SPI3B_I_DAT[23]	N1	SPI3D_I_DAT[28]	AD2	SPI3D_E_DAT[23]	AN6
SPI3B_E_DAT[17]	A3	SPI3B_I_DAT[22]	R5	SPI3D_I_DAT[29]	AD5	SPI3D_E_DAT[24]	AL7
SPI3B_E_DAT[16]	C4	SPI3B_I_DAT[21]	P5	SPI3D_I_DAT[30]	AD4	SPI3D_E_DAT[25]	AK7
SPI3B_E_DAT[15]	B3	SPI3B_I_DAT[20]	P4	SPI3D_I_DAT[31]	AD3	SPI3D_E_DAT[26]	AN7
SPI3B_E_DAT[14]	C2	SPI3B_I_DAT[19]	P2	SPI3D_I_EOP	AE1	SPI3D_E_DAT[27]	AM7
SPI3B_E_DAT[13]	C1	SPI3B_I_DAT[18]	P1	SPI3D_I_MOD[0]	AE3	SPI3D_E_DAT[28]	AK8
SPI3B_E_DAT[12]	C3	SPI3B_I_DAT[17]	P3	SPI3D_I_MOD[1]	AE2	SPI3D_E_DAT[29]	AP7
SPI3B_E_DAT[11]	D3	SPI3B_I_DAT[16]	R4	SPI3D_I_PRTY	AE4	SPI3D_E_DAT[30]	AM8
SPI3B_E_DAT[10]	D1	SPI3B_I_DAT[15]	R3	SPI3D_I_SOP	AF3	SPI3D_E_DAT[31]	AL8
SPI3B_E_DAT[9]	F5	SPI3B_I_DAT[14]	T5	SPI3D_I_FCLK	AE5	SPI3D_E_PRTY	AK9
SPI3B_E_DAT[8]	D2	SPI3B_I_DAT[13]	R2	SPI3D_I_ERR	AF1	SPI3D_E_SX	AN8
SPI3B_E_DAT[7]	E3	SPI3B_I_DAT[12]	R1	SPI3D_I_RVAL	AF2	SPI3D_E_ENB	AP8
SPI3B_E_DAT[6]	E1	SPI3B_I_DAT[11]	T4	SPI3D_I_SX	AF5	SPI3D_E_EOP	AL9
SPI3B_E_DAT[5]	F4	SPI3B_I_DAT[10]	U5	SPI3D_I_ENB	AF4	SPI3D_E_MOD[0]	AK10
SPI3B_E_DAT[4]	E2	SPI3B_I_DAT[9]	T3	SPI3D_DTPA[0]	AG2	SPI3D_E_MOD[1]	AM9
SPI3B_E_DAT[3]	F3	SPI3B_I_DAT[8]	T2	SPI3D_DTPA[1]	AG1	SPI3D_E_ERR	AN9
SPI3B_E_DAT[2]	F2	SPI3B_I_DAT[7]	T1	SPI3D_DTPA[2]	AG4	SPI3D_E_SOP	AP9
SPI3B_E_DAT[1]	G4	SPI3B_I_DAT[6]	U3	SPI3D_DTPA[3]	AG3	SPI3C_I_DAT[0]	AL10
SPI3B_E_DAT[0]	F1	SPI3B_I_DAT[5]	U4	SPI3D_STPA	AH1	SPI3C_I_DAT[1]	AM10
SPI3B_E_FCLK	G5	SPI3B_I_DAT[4]	U2	SPI3D_TADR[0]	AH2	SPI3C_I_DAT[2]	AN10
SPI3B_PTPA	G3	SPI3B_I_DAT[3]	U1	SPI3D_TADR[1]	AH3	SPI3C_I_DAT[3]	AP10
SPI3B_TADR[7]	H5	SPI3B_I_DAT[2]	V5	SPI3D_TADR[2]	AG5	SPI3C_I_DAT[4]	AK12
SPI3B_TADR[6]	G2	SPI3B_I_DAT[1]	V1	SPI3D_TADR[3]	AH4	SPI3C_I_DAT[5]	AK11
SPI3B_TADR[5]	G1	SPI3B_I_DAT[0]	V2	SPI3D_TADR[4]	AJ1	SPI3C_I_DAT[6]	AL11
SPI3B_TADR[4]	H4	SPI3D_I_DAT[0]	V4	SPI3D_TADR[5]	AJ4	SPI3C_I_DAT[7]	AN11
SPI3B_TADR[3]	H2	SPI3D_I_DAT[1]	W3	SPI3D_TADR[6]	AJ2	SPI3C_I_DAT[8]	AP11
SPI3B_TADR[2]	H3	SPI3D_I_DAT[2]	W1	SPI3D_TADR[7]	AJ3	SPI3C_I_DAT[9]	AM11
SPI3B_TADR[1]	H1	SPI3D_I_DAT[3]	V3	SPI3D_PTPA	AK1	SPI3C_I_DAT[10]	AL12
SPI3B_TADR[0]	J5	SPI3D_I_DAT[4]	W2	SPI3D_E_FCLK	AK2	SPI3C_I_DAT[11]	AM12
SPI3B_STPA	J4	SPI3D_I_DAT[5]	W5	SPI3D_E_DAT[0]	AH5	SPI3C_I_DAT[12]	AK13
SPI3B_DTPA[3]	J3	SPI3D_I_DAT[6]	W4	SPI3D_E_DAT[1]	AK4	SPI3C_I_DAT[13]	AN12
SPI3B_DTPA[2]	J2	SPI3D_I_DAT[7]	Y1	SPI3D_E_DAT[2]	AL1	SPI3C_I_DAT[14]	AP12
SPI3B_DTPA[1]	J1	SPI3D_I_DAT[8]	Y2	SPI3D_E_DAT[3]	AL2	SPI3C_I_DAT[15]	AL13
SPI3B_DTPA[0]	K5	SPI3D_I_DAT[9]	Y5	SPI3D_E_DAT[4]	AK3	SPI3C_I_DAT[16]	AM13
SPI3B_I_ENB	K4	SPI3D_I_DAT[10]	Y3	SPI3D_E_DAT[5]	AM1	SPI3C_I_DAT[17]	AN13
SPI3B_I_SX	K3	SPI3D_I_DAT[11]	Y4	SPI3D_E_DAT[6]	AM2	SPI3C_I_DAT[18]	AP13
SPI3B_I_RVAL	K2	SPI3D_I_DAT[12]	AA1	SPI3D_E_DAT[7]	AM3	SPI3C_I_DAT[19]	AN14
SPI3B_I_ERR	K1	SPI3D_I_DAT[13]	AA5	SPI3D_E_DAT[8]	AL3	SPI3C_I_DAT[20]	AK14
SPI3B_I_FCLK	L5	SPI3D_I_DAT[14]	AA2	SPI3D_E_DAT[9]	AP3	SPI3C_I_DAT[21]	AL14
SPI3B_I_SOP	L4	SPI3D_I_DAT[15]	AA3	SPI3D_E_DAT[10]	AL4	SPI3C_I_DAT[22]	AM14
SPI3B_I_PRTY	L3	SPI3D_I_DAT[16]	AA4	SPI3D_E_DAT[11]	AM4	SPI3C_I_DAT[23]	AP14
SPI3B_I_MOD[1]	M5	SPI3D_I_DAT[17]	AB4	SPI3D_E_DAT[12]	AN4	SPI3C_I_DAT[24]	AM15
SPI3B_I_MOD[0]	L2	SPI3D_I_DAT[18]	AB1	SPI3D_E_DAT[13]	AP4	SPI3C_I_DAT[25]	AL15
SPI3B_I_EOP	L1	SPI3D_I_DAT[19]	AB2	SPI3D_E_DAT[14]	AK5	SPI3C_I_DAT[26]	AN15
SPI3B_I_DAT[31]	M4	SPI3D_I_DAT[20]	AC2	SPI3D_E_DAT[15]	AL5	SPI3C_I_DAT[27]	AP15
SPI3B_I_DAT[30]	M3	SPI3D_I_DAT[21]	AB5	SPI3D_E_DAT[16]	AN5	SPI3C_I_DAT[28]	AK15
SPI3B_I_DAT[29]	M2	SPI3D_I_DAT[22]	AB3	SPI3D_E_DAT[17]	AM5	SPI3C_I_DAT[29]	AL16
SPI3B_I_DAT[28]	M1	SPI3D_I_DAT[23]	AC1	SPI3D_E_DAT[18]	AK6	SPI3C_I_DAT[30]	AK16
SPI3B_I_DAT[27]	N4	SPI3D_I_DAT[24]	AC3	SPI3D_E_DAT[19]	AP5	SPI3C_I_DAT[31]	AM16

## 12.2. Pin name/ball location table (continued)

SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL
SPI3C_I_EOP	AK17	SPI3C_E_DAT[27]	AL26	SPI4_E_DAT_N[12]	AC31	SPI4_I_DAT_N[4]	K31
SPI3C_I_MOD[0]	AN16	SPI3C_E_DAT[28]	AM27	SPI4_E_DAT_P[11]	AC34	SPI4_I_DAT_P[3]	K30
SPI3C_I_MOD[1]	AP16	SPI3C_E_DAT[29]	AN27	SPI4_E_DAT_N[11]	AC33	SPI4_I_DAT_N[3]	J30
SPI3C_I_PRTY	AL17	SPI3C_E_DAT[30]	AK27	SPI4_E_DAT_P[10]	AB31	SPI4_I_DAT_P[2]	J32
SPI3C_I_SOP	AK18	SPI3C_E_DAT[31]	AL27	SPI4_E_DAT_N[10]	AB30	SPI4_I_DAT_N[2]	J33
SPI3C_I_FCLK	AM17	SPI3C_E_PRTY	AN28	SPI4_E_DAT_P[9]	AB34	SPI4_I_DAT_P[1]	J34
SPI3C_I_ERR	AN17	SPI3C_E_SX	AP28	SPI4_E_DAT_N[9]	AB33	SPI4_I_DAT_N[1]	H34
SPI3C_RVAL	AP17	SPI3C_E_ENB	AL28	SPI4_E_DAT_P[8]	AA31	SPI4_I_DAT_P[0]	H32
SPI3C_I_SX	AK20	SPI3C_E_EOP	AM28	SPI4_E_DAT_N[8]	AA30	SPI4_I_DAT_N[0]	H33
SPI3C_I_ENB	AL18	SPI3C_E_MOD[0]	AP29	SPI4_E_DAT_P[7]	AA34	SPI4_I_CTRL_P	H30
SPI3C_DTPA[0]	AM18	SPI3C_E_MOD[1]	AK28	SPI4_E_DAT_N[7]	AA33	SPI4_I_CTRL_N	H31
SPI3C_DTPA[1]	AP18	SPI3C_E_ERR	AM29	SPI4_E_DAT_P[6]	Y32	SPI4_E_SCLK_P	G31
SPI3C_DTPA[2]	AK19	SPI3C_E_SOP	AN29	SPI4_E_DAT_N[6]	Y31	SPI4_E_SCLK_N	G32
SPI3C_DTPA[3]	AP19	REF_CLK	AP30	SPI4_E_DAT_P[5]	Y34	SPI4_E_STAT_P[1]	F33
SPI3C_STPA	AN18	TCK	AL29	SPI4_E_DAT_N[5]	Y33	SPI4_E_STAT_N[1]	F34
SPI3C_TADR[0]	AN19	LVDS_STA	AN30	SPI4_E_DAT_P[4]	W31	SPI4_E_STAT_P[0]	F32
SPI3C_TADR[1]	AK22	TDI	AM30	SPI4_E_DAT_N[4]	W30	SPI4_E_STAT_N[0]	E32
SPI3C_TADR[2]	AL19	TDO	AP32	SPI4_E_DAT_P[3]	W34	SPI4_E_STAT_T[1]	L30
SPI3C_TADR[3]	AP20	TMS	AN32	SPI4_E_DAT_N[3]	W33	SPI4_E_STAT_T[0]	M30
SPI3C_TADR[4]	AM20	GPIO[4]	AM31	SPI4_E_DAT_P[2]	V31	SPI4_E_SCLK_T	E31
SPI3C_TADR[5]	AM19	GPIO[3]	AM33	SPI4_E_DAT_N[2]	V30	SPI_EN	E29
SPI3C_TADR[6]	AL20	GPIO[2]	AM34	SPI4_E_DAT_P[1]	V34	DBUS[7]	E30
SPI3C_TADR[7]	AN20	CLK_SEL[3]	AL31	SPI4_E_DAT_N[1]	V33	ADD[5]	D33
SPI3C_PTPA	AP21	GPIO[1]	AL30	SPI4_E_DAT_P[0]	U32	DBUS[6]	D34
SPI3C_E_FCLK	AK21	CLK_SEL[2]	AK29	SPI4_E_DAT_N[0]	U31	ADD[4]	D31
SPI3C_E_DAT[0]	AN21	GPIO[0]	AL32	SPI4_E_CTRL_P	U34	DBUS[5]	D32
SPI3C_E_DAT[1]	AM21	CLK_SEL[1]	AK30	SPI4_E_CTRL_N	U33	ADD[3]	C34
SPI3C_E_DAT[2]	AL21	OCLK[3]	AK31	SPI4_I_DCLK_P	T34	DBUS[4]	D30
SPI3C_E_DAT[3]	AP22	CLK_SEL[0]	AH31	SPI4_I_DCLK_N	T33	WRB	C33
SPI3C_E_DAT[4]	AN22	OCLK[2]	AJ30	SPI4_I_DAT_P[15]	T31	DBUS[3]	C32
SPI3C_E_DAT[5]	AM22	TIMEBASE	AH34	SPI4_I_DAT_N[15]	T30	CSB	B32
SPI3C_E_DAT[6]	AL22	OCLK[1]	AH32	SPI4_I_DAT_P[14]	R34	DBUS[2]	C31
SPI3C_E_DAT[7]	AK23	OCLK[0]	AH33	SPI4_I_DAT_N[14]	R33	DBUS[1]	A32
SPI3C_E_DAT[8]	AP23	SPI4_I_STAT_T[1]	AH30	SPI4_I_DAT_P[13]	R31	RESETB	C30
SPI3C_E_DAT[9]	AN23	SPI4_I_STAT_T[0]	AG32	SPI4_I_DAT_N[13]	R30	INTB	A31
SPI3C_E_DAT[10]	AL23	SPI4_I_SCLK_T	AG33	SPI4_I_DAT_P[12]	P34	DBUS[0]	A30
SPI3C_E_DAT[11]	AM23	BIAS	AJ31	SPI4_I_DAT_N[12]	P33	ADD[2]	B30
SPI3C_E_DAT[12]	AP24	SPI4_I_STAT_P[1]	AG31	SPI4_I_DAT_P[11]	P32	RDB	E28
SPI3C_E_DAT[13]	AM24	SPI4_I_STAT_P[0]	AF31	SPI4_I_DAT_N[11]	P31	ADD[1]	D29
SPI3C_E_DAT[14]	AN24	SPI4_I_STAT_N[1]	AG30	SPI4_I_DAT_P[10]	N34	TRSTB	D28
SPI3C_E_DAT[15]	AK24	SPI4_I_STAT_N[0]	AF30	SPI4_I_DAT_N[10]	N33	ADD[0]	B28
SPI3C_E_DAT[16]	AL24	SPI4_I_SCLK_P	AF33	SPI4_I_DAT_P[9]	N31	MPM	C28
SPI3C_E_DAT[17]	AP25	SPI4_I_SCLK_N	AF32	SPI4_I_DAT_N[9]	N30	SPI3A_E_SOP	A28
SPI3C_E_DAT[18]	AN25	SPI4_E_DCLK_P	AE34	SPI4_I_DAT_P[8]	M34	SPI3A_E_ERR	E27
SPI3C_E_DAT[19]	AK25	SPI4_E_DCLK_N	AE33	SPI4_I_DAT_N[8]	M33	SPI3A_E_MOD[1]	C27
SPI3C_E_DAT[20]	AL25	SPI4_E_DAT_P[15]	AE30	SPI4_I_DAT_P[7]	M32	SPI3A_E_MOD[0]	D27
SPI3C_E_DAT[21]	AM25	SPI4_E_DAT_N[15]	AE31	SPI4_I_DAT_N[7]	M31	SPI3A_E_EOP	B27
SPI3C_E_DAT[22]	AP26	SPI4_E_DAT_P[14]	AD31	SPI4_I_DAT_P[6]	L34	SPI3A_E_ENB	A27
SPI3C_E_DAT[23]	AN26	SPI4_E_DAT_N[14]	AD30	SPI4_I_DAT_N[6]	L33	SPI3A_E_SX	D26
SPI3C_E_DAT[24]	AP27	SPI4_E_DAT_P[13]	AD34	SPI4_I_DAT_P[5]	K33	SPI3A_E_PRTY	E26
SPI3C_E_DAT[25]	AM26	SPI4_E_DAT_N[13]	AD33	SPI4_I_DAT_N[5]	K34	SPI3A_E_DAT[31]	C26
SPI3C_E_DAT[26]	AK26	SPI4_E_DAT_P[12]	AC32	SPI4_I_DAT_P[4]	L31	SPI3A_E_DAT[30]	B26

## 12.2 Pin name/ball location table (continued)

SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL	SIGNAL PIN NAME	BALL
SPI3A_E_DAT[29]	A26	SPI3A_E_DAT[2]	C20	SPI3A_I_EOP	D15	SPI3A_I_DAT[5]	A9
SPI3A_E_DAT[28]	E25	SPI3A_E_DAT[1]	B20	SPI3A_I_DAT[31]	E15	SPI3A_I_DAT[4]	B9
SPI3A_E_DAT[27]	D25	SPI3A_E_DAT[0]	D20	SPI3A_I_DAT[30]	A14	SPI3A_I_DAT[3]	D9
SPI3A_E_DAT[26]	C25	SPI3A_E_FCLK	A20	SPI3A_I_DAT[29]	E14	SPI3A_I_DAT[2]	E9
SPI3A_E_DAT[25]	A25	SPI3A_PTPA	C19	SPI3A_I_DAT[28]	B14	SPI3A_I_DAT[1]	C9
SPI3A_E_DAT[24]	B25	SPI3A_TADR[7]	B19	SPI3A_I_DAT[27]	C14	SPI3A_I_DAT[0]	A8
SPI3A_E_DAT[23]	C24	SPI3A_TADR[6]	D19	SPI3A_I_DAT[26]	D14	SPI3B_E_SOP	C8
SPI3A_E_DAT[22]	D24	SPI3A_TADR[5]	A19	SPI3A_I_DAT[25]	E13	SPI3B_E_ERR	D8
SPI3A_E_DAT[21]	D23	SPI3A_TADR[4]	A18	SPI3A_I_DAT[24]	A13	SPI3B_E_MOD[1]	B8
SPI3A_E_DAT[20]	E23	SPI3A_TADR[3]	B18	SPI3A_I_DAT[23]	B13	SPI3B_E_MOD[0]	E8
SPI3A_E_DAT[19]	B24	SPI3A_TADR[2]	E19	SPI3A_I_DAT[22]	C13	SPI3B_E_EOP	B7
SPI3A_E_DAT[18]	A24	SPI3A_TADR[1]	C18	SPI3A_I_DAT[21]	E12	SPI3B_E_ENB	C7
SPI3A_E_DAT[17]	C23	SPI3A_TADR[0]	D18	SPI3A_I_DAT[20]	D13	SPI3B_E_SX	A7
SPI3A_E_DAT[16]	E24	SPI3A_STPA	A17	SPI3A_I_DAT[19]	A12	SPI3B_E_PRTY	A6
SPI3A_E_DAT[15]	B23	SPI3A_DTPA[3]	E18	SPI3A_I_DAT[18]	B12	SPI3B_E_DAT[31]	D7
SPI3A_E_DAT[14]	A23	SPI3A_DTPA[2]	B17	SPI3A_I_DAT[17]	E11	SPI3B_E_DAT[30]	B6
SPI3A_E_DAT[13]	D22	SPI3A_DTPA[1]	C17	SPI3A_I_DAT[16]	D12	SPI3B_E_DAT[29]	E7
SPI3A_E_DAT[12]	E22	SPI3A_DTPA[0]	D17	SPI3A_I_DAT[15]	C12	SPI3B_E_DAT[28]	E6
SPI3A_E_DAT[11]	C22	SPI3A_I_ENB	E17	SPI3A_I_DAT[14]	A11	SPI3B_E_DAT[27]	C6
SPI3A_E_DAT[10]	B22	SPI3A_I_SX	A16	SPI3A_I_DAT[13]	C11	SPI3B_E_DAT[26]	A5
SPI3A_E_DAT[9]	A22	SPI3A_RVAL	B16	SPI3A_I_DAT[12]	D11	SPI3B_E_DAT[25]	D6
SPI3A_E_DAT[8]	E21	SPI3A_I_ERR	C16	SPI3A_I_DAT[11]	B11	SPI3B_E_DAT[24]	B5
SPI3A_E_DAT[7]	D21	SPI3A_I_FCLK	D16	SPI3A_I_DAT[10]	A10	SPI3B_E_DAT[23]	D5
SPI3A_E_DAT[6]	C21	SPI3A_I_SOP	B15	SPI3A_I_DAT[9]	B10	SPI3B_E_DAT[22]	C5
SPI3A_E_DAT[5]	B21	SPI3A_I_PRTY	A15	SPI3A_I_DAT[8]	C10		
SPI3A_E_DAT[4]	E20	SPI3A_I_MOD[1]	C15	SPI3A_I_DAT[7]	E10		
SPI3A_E_DAT[3]	A21	SPI3A_I_MOD[0]	E16	SPI3A_I_DAT[6]	D10		

POWER PIN NAME	BALL(S)
VDDA18_CLKGEN	AF28
VSSA18_CLKGEN	AA20
VDDA18_ISTX	AD28
VSSA18_ISTX	Y21
VDDA18_EDTX	AE28
VSSA18_EDTX	AA21
VDDA18_IDRX	P28
VSSA18_IDRX	R21
VDDA18_ESRX	L28
VSSA18_ESRX	P21
VSS (GND)	A29, B29, C29, E4, F6, F7, F17, F18, F28, F29, F31, G6, G7, G17, G18, G28, G29, G33, G34, J31, L32, N32, P14 - P20, R14- R20, T14 - T21, T32, U6, U7, U14 - U21, U28, U29, V6, V7, V14 - V21, V28, V29, V32, W14 - W21, Y14 - Y20, Y30, AA14 - AA19, AA32, AC30, AD32, AF34, AG34, AH6, AH7, AH17, AH18, AH28, AH29, AJ5 - AJ7, AJ17, AJ18, AJ28, AJ29, AJ32 - AJ34, AK32, AK33, AN3, AP31, AN31, B31
VDD18 (1.8 VOLTS)	A1, A2, A33, A34, B1, B2, B33, B34, E33, E34, F8, F9, F12, F13, F15, F16, F19, F20, F22, F23, F26, F27, F30, G8, G9, G12, G13, G15, G16, G19, G20, G22, G23, G26, G27, G30, H6, H7, H28, H29, J6, J7, J28, J29, K32, M6, M7, M28, M29, N6, N7, N28, N29, R6, R7, R28, R29, R32, T6, T7, T28, T29, W6, W7, W28, W29, W32, Y6, Y7, Y28, Y29, AB6, AB7, AB28, AB29, AB32, AC6, AC7, AC28, AC29, AE32, AF6, AF7, AH25, AF29, AG6, AG7, AG28, AG29, AH8, AH9, AH12, AH13, AH15, AH16, AH19, AH20, AH22, AH23, AH26, AH27, AJ8, AJ9, AJ12, AJ13, AJ15, AJ16, AJ19, AJ20, AJ22, AJ23, AJ26, AJ27, AK34, AL33, AL34, AN1, AN2, AN33, AN34, AP1, AP2, AP33, AP34
VD33 (3.3 VOLTS)	F10, F11, F14, F21, G10, G11, G14, G21, K6, K7, L6, L7, P6, P7, AA6, AA7, AD6, AD7, AE6, AE7, AH10, AH11, AH14, AH21, AJ10, AJ11, AJ14, AJ21, AM32
VDDA33 (3.3 VOLTS)	F24, F25, G24, G25, K28, K29, L29, P29, P30, U30, AA28, AA29, AD29, AE29, AJ24, AH24, AJ25

### 12.3 Device package

The SPI Exchange IDT88P8344 device is packaged in a 35 mm by 35 mm 820-ball one millimeter ball pitch thermally-enhanced plastic ball grid array. All balls, whether used or unused, must be soldered to pads.

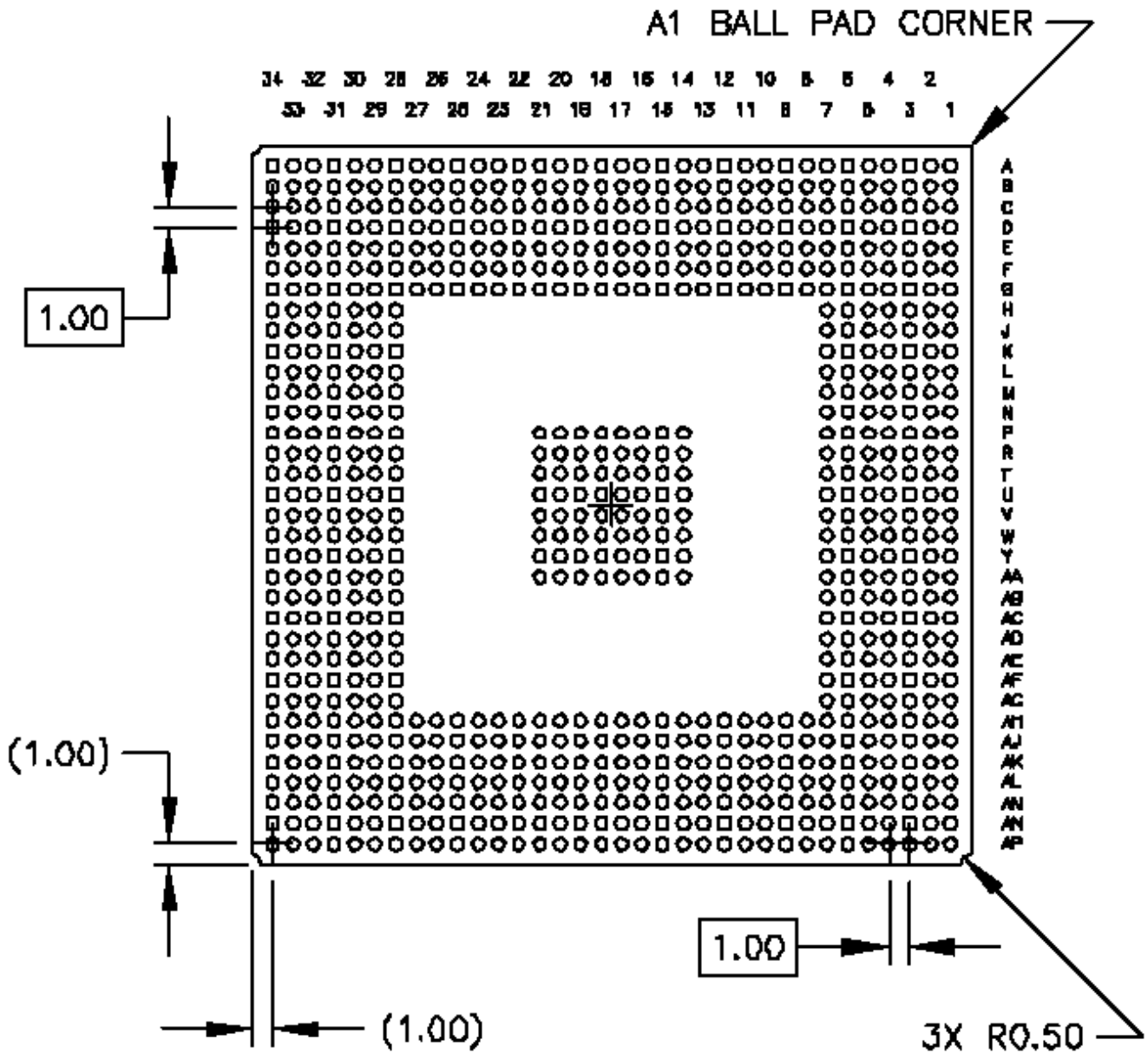
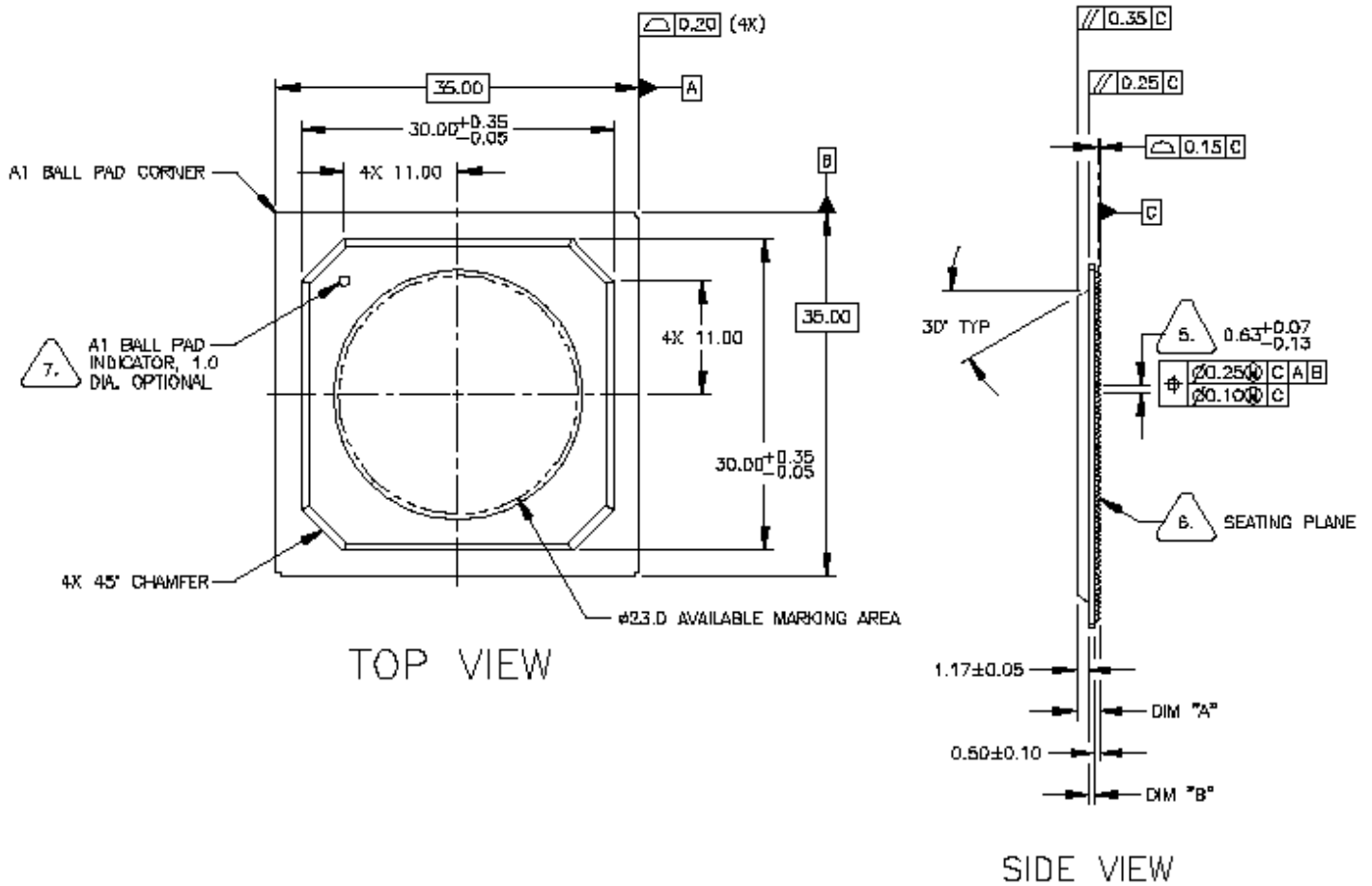


Figure 42. IDT88P8344 820PBGA package, bottom view



4	2.23±0.21	0.56±0.06	STANDARD
NO. LAYERS	DIM "A"	DIM "B"	NOTES
TEPBGA-2 THICKNESS SCHEDULE			

Figure 43. IDT88P8344 820PBGA package, top and side views

**13. GLOSSARY**

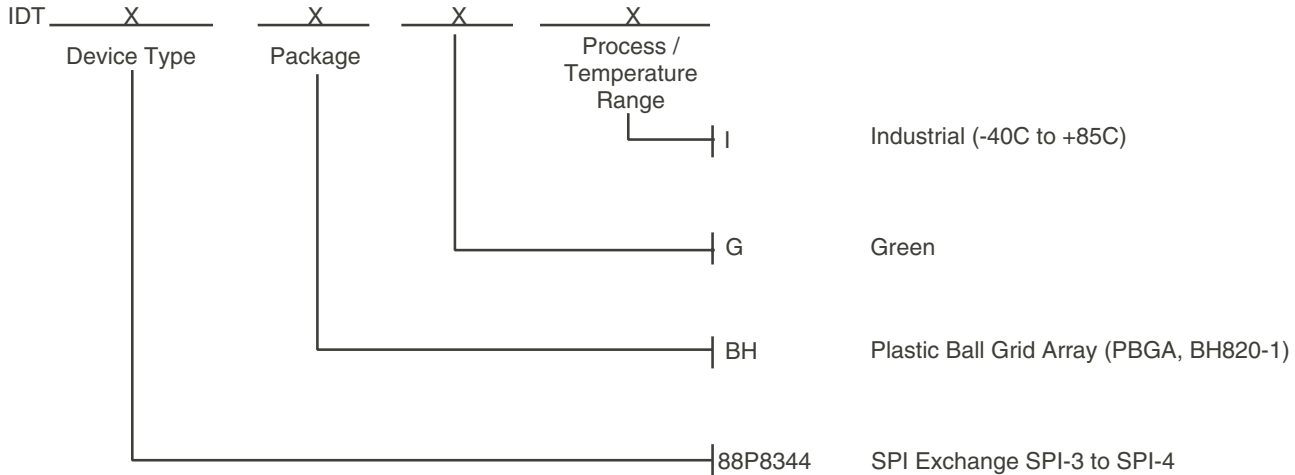
ACRONYM	MEANING
FIFO	First In First Out memory
LID	Logical Identifier See also Logical Port (LP) The entity associated with a flow of data between a SPI-3 LP to a SPI-4 LP, or a SPI-4 LP to a SPI-3 LP.
LP	Logical Port. See also Logical Identifier (LID) The entity associated with a SPI-3 or SPI-4 address.
OIF	Optical Internetworking Forum
SPI-3	System Packet Interface Level 3 This interface is defined by the OIF implementation agreement OIF-SPI3-01.0 - SPI-3 Packet Interface for Physical and Link Layers for OC-48 available at <a href="http://www.oiforum.com/public/impagreements.html">http://www.oiforum.com/public/impagreements.html</a>
SPI-4	System Packet Interface Level 4 phase 2 This interface is defined by the OIF implementation agreement OIF-SPI4-02.1 - System Packet Interface Level 4 (SPI-4) Phase 2: OC-192 System Interface for Physical and Link Layer Devices available at <a href="http://www.oiforum.com/public/impagreements.html">http://www.oiforum.com/public/impagreements.html</a>



**14. DATASHEET DOCUMENT REVISION HISTORY**

ISSUE	DATE	DESCRIPTION
0.7	05/21/04	<ul style="list-style-type: none"> <li>• General Release</li> </ul>
0.8	10/01/04	<ul style="list-style-type: none"> <li>• AG30 ball location changed to SPI4_I_STAT_N[1] and AF31 ball location changed to SPI4_I_STAT_P[0] on Pin name/ball location table (table 12.2) on page 88</li> </ul>
0.9	03/01/05	<ul style="list-style-type: none"> <li>• Updated Chip Configuration Sequence (p. 41)</li> <li>• Update Table 21: "BIT Order Within a 16-BIT Address Register" (p. 45)</li> <li>• Update Table 27: "Indirect Access Address Register" (p. 46)</li> <li>• Updated Direct Access Registers (p. 48-49)</li> <li>• Updated Common Module Indirect Registers (p. 66)</li> <li>• Updated Electrical and Thermal Specification (p. 77) (the section name changed from Electrical Characteristics to Electrical and Thermal Specification). Updated JTAG Instructions.</li> <li>• Added Document Revision History (p. 95)</li> </ul>
0.91	05/09/05	<ul style="list-style-type: none"> <li>• Added sections System Reset and Power on Sequence (p.41)</li> <li>• Updated PFR to PFP in Table 49: "Module A/B/C/D indirect register" (p.55)</li> <li>• Updated length for Reserved in Table 83: "SPI-4 ingress port descriptor" (p.65)</li> <li>• Added Green to Ordering information (p.96)</li> </ul>
0.92	08/05/05	<ul style="list-style-type: none"> <li>• Updated Table 128: "Absolute maximum ratings" (p.78)</li> </ul>
0.93	10/20/05	<ul style="list-style-type: none"> <li>• Updated Table 7: "Parallel microprocessor interface" (p.12)</li> <li>• Updated Table 131: "Thermal Characteristics" (p.79)</li> <li>• Updated Microprocessor parallel port section (p.84-87)</li> </ul>
0.94	01/09/06	<ul style="list-style-type: none"> <li>• Updated Figure 4: "PHY mode SPI-3 ingress interface" (p.14)</li> <li>• Deleted Table 13: "NR_LID Field Encoding". Updated SPI-4 egress queues, Normal operation section (p.26)</li> <li>• Updated Section 8.2.5 "SPI-4 status channel software" (p.43)</li> <li>• Updated Table 127: "Absolute maximum ratings" (p.78)</li> <li>• Updated Table 129: "Terminal Capacitance" (p.79)</li> </ul>
1.0	04/10/06	<ul style="list-style-type: none"> <li>• Initial Release of Final Datasheet with new section 8.2.7 "Software Eye-Opening Check on SPI-4" &amp; new Figure 33. "DDR interface and eye opening check through over sampling" (p.44-45)</li> <li>• Updated Clock generator (pg. 39)</li> <li>• Updated SPI-4 ingress watermark register (pg. 71)</li> <li>• Updated Clock generator control register (pg. 77)</li> <li>• Updated Table 130: Thermal Characteristics (pg. 80)</li> <li>• Updated Table 132: SPI-3 AC Input/Output timing specifications (pg. 82)</li> <li>• Updated Table 136: OCLK[3:0] outputs and MCLK internal clock (pg. 84)</li> </ul>

## 15. ORDERING INFORMATION



6370 drwlast

**NOTE:**

1. Green parts are available.



**CORPORATE HEADQUARTERS**  
6024 Silver Creek Valley Road  
San Jose, CA 95138

**for SALES:**  
800-345-7015 or 408-284-8200  
fax: 408-284-2775  
www.idt.com

**for Tech Support:**  
408-360-1716  
email: SPI@idt.com