

October 2009

## Features

### General

- Circuit Emulation Services over Packet (CESoP) transport for MPLS, IP and Ethernet networks
- On chip timing & synchronization recovery across a packet network
- On chip dual reference Stratum 4 DPLL (Stratum 3 Holdover accuracy)
- Grooming capability for Nx64 Kbps trunking
- Fully compatible with Zarlink's ZL50110, ZL50111, ZL50112 and ZL50114 CESoP processors

### Circuit Emulation Services

- Supports ITU-T recommendation Y.1413 and Y.1453
- Supports IETF RFC4553 and RFC5086
- Supports MEF8 and MFA 8.0.0
- Structured, synchronous CESoP with clock recovery
- Unstructured, asynchronous CESoP, with integral per stream clock recovery



### Ordering Information

ZL50115GAG	324 Ball PBGA	trays, bake & dry pack
ZL50116GAG	324 Ball PBGA	trays, bake & dry pack
ZL50117GAG	324 Ball PBGA	trays, bake & dry pack
ZL50118GAG	324 Ball PBGA	trays, bake & dry pack
ZL50119GAG	324 Ball PBGA	trays, bake & dry pack
ZL50120GAG	324 Ball PBGA	trays, bake & dry pack
ZL50115GAG2	324 Ball PBGA**	trays, bake & dry pack
ZL50116GAG2	324 Ball PBGA**	trays, bake & dry pack
ZL50117GAG2	324 Ball PBGA**	trays, bake & dry pack
ZL50118GAG2	324 Ball PBGA**	trays, bake & dry pack
ZL50119GAG2	324 Ball PBGA**	trays, bake & dry pack
ZL50120GAG2	324 Ball PBGA**	trays, bake & dry pack

\*\*Pb Free Tin/Silver/Copper

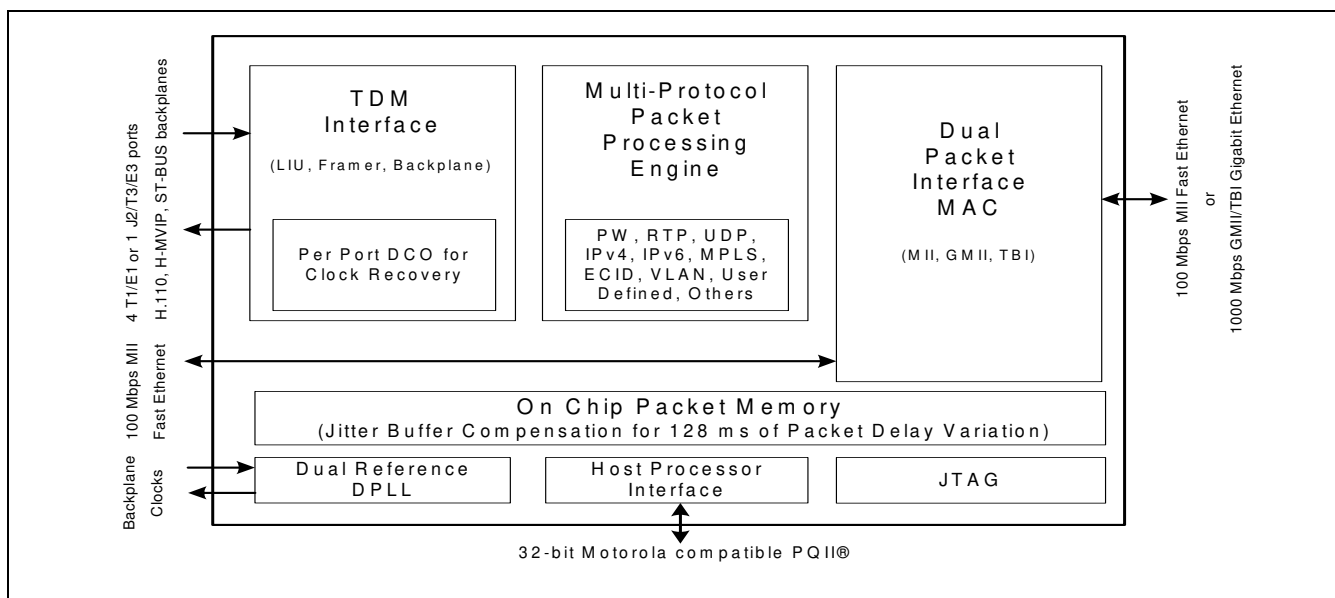
-40°C to +85°C

### Customer Side TDM Interfaces

- Up to 4 T1/E1, 1 J2 or 1 T3/E3 ports
- H.110, H-MVIP, ST-BUS backplane
- Up to 128 bi-directional 64 Kbps channels
- Direct connection to LIUs, framers, backplanes

### Customer Side Packet Interfaces

- 100 Mbps MII Fast Ethernet (ZL50118/19/20 only) (may also be used as a second provider side packet interface)



**Figure 1 - ZL50115/16/17/18/19/20 High Level Overview**

**Provider Side Packet Interfaces**

- 100 Mbps MII Fast Ethernet or 1000 Mbps GMII/TBI Gigabit Ethernet

**System Interfaces**

- Flexible 32 bit Motorola host interface
- On-chip packet memory with jitter buffer compensation for over 128 ms of packet delay variation

**Packet Processing Functions**

- Flexible, multi-protocol packet encapsulation including IPv4, IPv6, RTP, MPLS, L2TPv3, ITU-T Y.1413, IETF CESoPSN, IETF SAToP and user programmable
- Packet re-sequencing to allow lost packet detection and re-ordering
- Four classes of service with programmable priority mechanisms (WFQ and SP) using egress queues
- Programmable classification of incoming packets at layers 2 through 5
- Wire speed processing of all packets regardless of classification providing low latency
- Supports up to 128 separate CESoP connections across the Packet Switched Network

**Applications**

- Circuit Emulation Services over Packet Networks
  - Leased Line support over packet networks
  - TDM over Cable
  - TDM over WiFi (802.11x)
  - TDM over WiMAX (802.16)
  - Fibre To The Premises G/E-PON
  - Layer 2 VPN services
- Customer-premise and Provider Edge Routers and Switches
- Ethernet and IP based IADs

## Table of Contents

<b>1.0 Changes Summary</b> .....	<b>8</b>
<b>2.0 Device Line Up</b> .....	<b>9</b>
<b>2.0 Description</b> .....	<b>10</b>
<b>3.0 Physical Specification</b> .....	<b>11</b>
<b>4.0 External Interface Description</b> .....	<b>26</b>
4.1 TDM Interface .....	26
4.1.1 TDM stream connections .....	26
4.1.2 TDM Signals common to ZL50115/16/17/18/19/20 .....	28
4.2 PAC Interface .....	29
4.3 Packet Interfaces .....	30
4.4 CPU Interface .....	35
4.5 System Function Interface .....	38
4.6 Test Facilities .....	38
4.6.1 Administration, Control and Test Interface .....	38
4.6.2 JTAG Interface .....	39
4.7 Miscellaneous Inputs .....	39
4.8 Power and Ground Connections .....	40
4.9 Internal Connections .....	41
4.10 No Connections .....	41
4.11 Device ID .....	41
<b>5.0 Typical Applications</b> .....	<b>42</b>
5.1 Leased Line Provision .....	42
5.2 Remote Concentrator Unit .....	43
5.3 FTTP .....	44
5.4 Wireless - WiFi or WiMAX .....	45
5.5 Digital Loop Carrier .....	46
5.6 Integrated Access Device .....	47
<b>6.0 Functional Description</b> .....	<b>47</b>
6.1 Block Diagram .....	48
6.2 Data and Control Flows .....	48
6.3 SYSTEM_CLK Considerations .....	49
6.4 TDM Interface .....	49
6.4.1 TDM Interface Block .....	49
6.4.2 Structured TDM Port Data Formats .....	50
6.4.3 TDM Clock Structure .....	51
6.4.3.1 Synchronous TDM Clock Generation .....	51
6.4.3.2 Asynchronous TDM Clock Generation .....	51
6.5 Payload Assembly .....	51
6.5.1 Structured Payload Operation .....	52
6.5.1.1 Payload Order .....	53
6.5.2 Unstructured Payload Operation .....	53
6.6 Protocol Engine .....	54
6.7 Packet Transmission .....	54
6.8 Packet Reception .....	54
6.9 TDM Formatter .....	54
6.10 Ethernet Traffic Aggregation (ZL50118/19/20 only) .....	54
<b>7.0 Clock Recovery</b> .....	<b>55</b>
7.1 Differential Clock Recovery .....	55
7.2 Adaptive Clock Recovery .....	56
<b>8.0 System Features</b> .....	<b>56</b>
8.1 Latency .....	56

## Table of Contents

8.2 Loopback Modes . . . . .	57
8.3 Host Packet Generation . . . . .	57
8.4 Loss of Service (LOS) . . . . .	57
8.5 Power Up Sequence . . . . .	58
8.6 JTAG Interface and Board Level Test Features. . . . .	58
8.7 External Component Requirements . . . . .	58
8.8 Miscellaneous Features. . . . .	59
8.9 Test Modes Operation . . . . .	59
8.9.1 Overview . . . . .	59
8.9.1.1 System Normal Mode . . . . .	59
8.9.1.2 System Tri-State Mode . . . . .	59
8.9.2 Test Mode Control . . . . .	59
8.9.3 System Normal Mode. . . . .	59
8.9.4 System Tri-state Mode. . . . .	59
<b>9.0 DPLL Specification . . . . .</b>	<b>60</b>
9.1 Modes of Operation . . . . .	60
9.1.1 Locking Mode (normal operation) . . . . .	60
9.1.2 Holdover Mode. . . . .	61
9.1.3 Freerun Mode. . . . .	61
9.1.4 Powerdown Mode . . . . .	61
9.2 Reference Monitor Circuit . . . . .	61
9.3 Locking Mode Reference Switching . . . . .	62
9.4 Locking Range. . . . .	62
9.5 Locking Time . . . . .	62
9.6 Lock Status . . . . .	63
9.7 Jitter. . . . .	63
9.7.1 Acceptance of Input Wander . . . . .	63
9.7.2 Intrinsic Jitter . . . . .	63
9.7.3 Jitter Tolerance . . . . .	63
9.7.4 Jitter Transfer. . . . .	64
9.8 Maximum Time Interval Error (MTIE) . . . . .	64
<b>10.0 Memory Map and Register Definitions. . . . .</b>	<b>65</b>
<b>11.0 DC Characteristics. . . . .</b>	<b>65</b>
<b>12.0 AC Characteristics. . . . .</b>	<b>68</b>
12.1 TDM Interface Timing - ST-BUS . . . . .	68
12.1.1 ST-BUS Slave Clock Mode . . . . .	68
12.1.2 ST-BUS Master Clock Mode . . . . .	70
12.2 TDM Interface Timing - H.110 Mode . . . . .	71
12.3 TDM Interface Timing - H-MVIP . . . . .	72
12.4 TDM LIU Interface Timing . . . . .	74
12.5 PAC Interface Timing . . . . .	75
12.6 Packet Interface Timing. . . . .	75
12.6.1 MII Transmit Timing . . . . .	75
12.6.2 MII Receive Timing . . . . .	76
12.6.3 GMII Transmit Timing. . . . .	78
12.6.4 GMII Receive Timing . . . . .	79
12.6.5 TBI Interface Timing. . . . .	80
12.6.6 Management Interface Timing . . . . .	81
12.7 CPU Interface Timing . . . . .	82
12.8 System Function Port . . . . .	85
12.9 JTAG Interface Timing. . . . .	86
<b>13.0 Power Characteristics . . . . .</b>	<b>87</b>

---

## Table of Contents

<b>14.0 Design and Layout Guidelines</b> .....	<b>88</b>
14.1 High Speed Clock & Data Interfaces .....	88
14.1.1 GMAC Interface - Special Considerations During Layout .....	89
14.1.2 TDM Interface - Special Considerations During Layout .....	89
14.1.3 Summary .....	89
14.2 CPU TA Output .....	90
<b>15.0 Reference Documents</b> .....	<b>91</b>
15.1 External Standards/Specifications .....	91
15.2 Zarlink Standards .....	91
<b>16.0 Glossary</b> .....	<b>92</b>

## List of Figures

Figure 1 - ZL50115/16/17/18/19/20 High Level Overview . . . . .	1
Figure 2 - ZL50115 Package View and Ball Positions . . . . .	12
Figure 3 - ZL50116 Package View and Ball Positions . . . . .	13
Figure 4 - ZL50117 Package View and Ball Positions . . . . .	14
Figure 5 - ZL50118 Package View and Ball Positions . . . . .	15
Figure 6 - ZL50119 Package View and Ball Positions . . . . .	16
Figure 7 - ZL50120 Package View and Ball Positions . . . . .	17
Figure 8 - Leased Line Services Over a Circuit Emulation Link. . . . .	42
Figure 9 - Remote Concentrator Unit using CESoP. . . . .	43
Figure 10 - EPON using CESoP . . . . .	44
Figure 11 - Wi-Fi and WiMAX using CESoP . . . . .	45
Figure 12 - Digital Loop Carrier using CESoP . . . . .	46
Figure 13 - Integrated Access Device Using CESoP . . . . .	47
Figure 14 - ZL50115/16/17/18/19/20 Family Operation . . . . .	47
Figure 15 - ZL50115/16/17/18/19/20 Data and Control Flows . . . . .	48
Figure 16 - Synchronous TDM Clock Generation . . . . .	51
Figure 17 - ZL50115/16/17/18/19/20 Packet Format - Structured Mode . . . . .	52
Figure 18 - Channel Order for Packet Formation. . . . .	53
Figure 19 - ZL50115/16/17/18/19/20 Packet Format - Unstructured Mode . . . . .	53
Figure 20 - Differential Clock Recovery . . . . .	55
Figure 21 - Adaptive Clock Recovery . . . . .	56
Figure 22 - Powering Up the ZL5011x . . . . .	58
Figure 23 - Jitter Transfer Function . . . . .	64
Figure 24 - Jitter Transfer Function - Detail . . . . .	65
Figure 25 - TDM ST-BUS Slave Mode Timing at 8.192 Mbps . . . . .	69
Figure 26 - TDM ST-BUS Slave Mode Timing at 2.048 Mbps . . . . .	69
Figure 27 - TDM Bus Master Mode Timing at 8.192 Mbps . . . . .	70
Figure 28 - TDM Bus Master Mode Timing at 2.048 Mbps . . . . .	71
Figure 29 - H.110 Timing Diagram . . . . .	72
Figure 30 - TDM - H-MVIP Timing Diagram for 16 MHz Clock (8.192 Mbps) . . . . .	73
Figure 31 - TDM-LIU Structured Transmission/Reception . . . . .	74
Figure 32 - MII Transmit Timing Diagram. . . . .	76
Figure 33 - MII Receive Timing Diagram . . . . .	77
Figure 34 - GMII Transmit Timing Diagram . . . . .	78
Figure 35 - GMII Receive Timing Diagram. . . . .	79
Figure 36 - TBI Transmit Timing Diagram . . . . .	80
Figure 37 - TBI Receive Timing Diagram. . . . .	81
Figure 38 - Management Interface Timing for Ethernet Port - Read . . . . .	81
Figure 39 - Management Interface Timing for Ethernet Port - Write . . . . .	81
Figure 40 - CPU Read - MPC8260 . . . . .	83
Figure 41 - CPU Write - MPC8260. . . . .	83
Figure 42 - CPU DMA Read - MPC8260 . . . . .	84
Figure 43 - CPU DMA Write - MPC8260 . . . . .	84
Figure 44 - JTAG Signal Timing . . . . .	86
Figure 45 - JTAG Clock and Reset Timing. . . . .	87
Figure 46 - ZL50115/16/17/18/19/20 Power Consumption Plot . . . . .	87
Figure 47 - CPU_TA Board Circuit. . . . .	90

## List of Tables

Table 1 - Capacity of Devices in the ZL50115/16/17/18/19/20 Family . . . . .	9
Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment . . . . .	18
Table 3 - TDM Interface Stream Pin Definition . . . . .	26
Table 4 - TDM Interface Common Pin Definition . . . . .	28
Table 5 - PAC Interface Package Ball Definition . . . . .	29
Table 6 - Packet Interface Signal Mapping - MII to GMII/TBI . . . . .	30
Table 7 - MII Management Interface Package Ball Definition . . . . .	31
Table 8 - MII Port 0 Interface Package Ball Definition . . . . .	31
Table 9 - MII Port 1 Interface Package Ball Definition . . . . .	34
Table 10 - CPU Interface Package Ball Definition . . . . .	35
Table 11 - System Function Interface Package Ball Definition . . . . .	38
Table 12 - Administration/Control Interface Package Ball Definition . . . . .	38
Table 13 - JTAG Interface Package Ball Definition . . . . .	39
Table 14 - Miscellaneous Inputs Package Ball Definitions . . . . .	39
Table 15 - Power and Ground Package Ball Definition . . . . .	40
Table 16 - Internal Connections Package Ball Definitions . . . . .	41
Table 17 - Miscellaneous Inputs Package Ball Definitions . . . . .	41
Table 18 - Device ID Ball Definition . . . . .	41
Table 19 - Standard Device Flows . . . . .	48
Table 20 - TDM Services Offered by the ZL50115/16/17/18/19/20 Family . . . . .	49
Table 21 - Some of the TDM Port Formats Accepted by the ZL50115/16/17/18/19/20 Family . . . . .	50
Table 22 - DMA Maximum Bandwidths . . . . .	57
Table 23 - Test Mode Control . . . . .	59
Table 24 - DPLL Input Reference Frequencies . . . . .	60
Table 25 - TDM ST-BUS Master Timing Specification . . . . .	70
Table 26 - TDM H.110 Timing Specification . . . . .	71
Table 27 - TDM H-MVIP Timing Specification . . . . .	72
Table 28 - TDM - LIU Structured Transmission/Reception . . . . .	74
Table 29 - PAC Timing Specification . . . . .	75
Table 30 - MII Transmit Timing - 100 Mbps . . . . .	75
Table 31 - MII Receive Timing - 100 Mbps . . . . .	76
Table 32 - GMII Transmit Timing - 1000 Mbps . . . . .	78
Table 33 - GMII Receive Timing - 1000 Mbps . . . . .	79
Table 34 - TBI Timing - 1000 Mbps . . . . .	80
Table 35 - MAC Management Timing Specification . . . . .	81
Table 36 - CPU Timing Specification . . . . .	82
Table 37 - System Clock Timing . . . . .	85
Table 38 - JTAG Interface Timing . . . . .	86

## 1.0 Change Summary

The following table captures the changes from the March 2009 issue.

Page	Item	Change
1	MEF logo	Added MEF logo to show MEF 18 certification.

The following table captures the changes from the May 2008 issue.

Page	Item	Change
50	Section 6.3	Replaced ZLAN-143 with ZL5011x Design Manual section "3.6 System Clock Block".
52 & 55	Section 6.5 and Section 6.8	Replaced ZLAN-202 with ZL5011x Design Manual section "13.1 Understanding forceDelete".
60	Section 8.4	Replaced ZLAN-159 with ZL5011x Design Manual section "3.1.1 Connection to LIU.
81	Section 12.6.5	Replaced ZLAN-239 with ZL5011x Design Manual section "7.1.3.1 TBI Interface Timing"
92	Section 15.1	Removed reference to IETF PWE3 draft-ietf-l2tpext-l2tp-base-02"

The following table captures the changes from the February 2006 issue.

Page	Item	Change
1, 2 and 11	Standard	Updated IETF RFC number and standards in general
1, 51,62, 63 and 66	Stratum 3 DPLL	Updated the description for Stratum 3 DPLL
1,10, 11, 27, 28 and 49	STS-1 stream	Remove STS-1 stream
31	Section 4.3	Include more detailed description for the packet interface
49	Section 6	Add a note about jumbo packets
50	Section 6.4	Include a paragraph to clarify the support for structure and unstructure modes at the same time
53	Section 6.5	Include more detailed description for the Payload Assembly
55	Section 6.5.2	Add a note at the end of the section
56	Section 6.9	Include more detailed description for the TDM formatter
57	Section 7	Include more detailed description for Clock Recovery
57	Section 7.1	Include more detailed description for Differential Clock Recovery
58	Section 7.2	Updated the description of Adaptive Clock Recovery
60	Section 8.5	Update Power Up Sequence
80	Section 12.6.3	Updated values for $t_{DV}$ , $t_{EV}$ and $t_{ER}$ in Table 32



Page	Item	Change
82	Section 12.6.5	Updated TXD[9:0] output delay
83	Section 12.6.6	Updated Section 12.6.6 Management Interface Timing (M_MDIO hold time and Figure 39)
85	CPU_TS_ALE and CPU_TA	Added mode details in Figure 40 and Figure 41 Added the CPU_TA assertion time

The following table captures the changes from the July 2005 issue.

Page	Item	Change
39, 40	Section 4.5 and Section 4.6.2	Added external pull-up/pull-down resistor recommendations for SYSTEM_RST, SYSTEM_DEBUG, JTAG_TRST, JTAG_TCK.

The following table captures the changes from the April 2005 issue.

Page	Item	Change
50	Section 6.3	Added Section 6.3 SYSTEM_CLK Considerations.

The following table captures the changes from the January 2005 issue.

Page	Item	Change
		Clarified data sheet to indicate ZL5011x supports clock recovery in both synchronous and asynchronous modes of operation.
85	Figure 42	Inverted polarity of CPU_DREQ0 and CPU_DREQ1 to conform with default MPC8260. Polarity of CPU_DREQ and CPU_SDACK remains programmable through API.
85	Figure 43	Inverted polarity of CPU_DREQ0 and CPU_DREQ1 to conform with default MPC8260. Polarity of CPU_DREQ and CPU_SDACK remains programmable through API.

The following table captures the changes from the November 2004 issue.

Page	Item	Change
39	Section 4.6.1	Added 5 kohm pulldown recommendation to GPIO signals.

## 2.0 Device Line Up

There are six products within the ZL5011x family, with capacities as shown in Table 1.

Product Number	TDM Interface	Provider Side Packet Interface	Customer Side Packet Interface
ZL50115	1 T1 or 1 E1 stream or 1 MVIP/ST-BUS stream at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps (Maximum of 32 DS0 or Nx64 kbps channels)	100 Mbps MII or 1000 Mbps GMII/TBI	None
ZL50116	2 T1 or 2 E1 streams or 2 MVIP/ST-BUS streams at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps (Maximum of 64 DS0 or Nx64 kbps channels)	100 Mbps MII or 1000 Mbps GMII/TBI	None
ZL50117	4 T1 or 4 E1 streams or 1 J2, 1 T3 or 1 E3 or 4 MVIP/ST-BUS streams at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps	100 Mbps MII or 1000 Mbps GMII/TBI	None
ZL50118	1 T1 or 1 E1 stream or 1 MVIP/ST-BUS stream at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps (Maximum of 32 DS0 or Nx64 kbps channels)	100 Mbps MII or 1000 Mbps GMII/TBI	100 Mbps MII
ZL50119	2 T1 or 2 E1 streams or 2 MVIP/ST-BUS streams at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps (Maximum of 64 DS0 or Nx64 kbps channels)	100 Mbps MII or 1000 Mbps GMII/TBI	100 Mbps MII
ZL50120	4 T1 or 4 E1 streams or 1 J2, 1 T3 or 1 E3 or 4 MVIP/ST-BUS streams at 2.048 Mbps or 1 H.110/H-MVIP/ST-BUS streams at 8.192 Mbps	100 Mbps MII or 1000 Mbps GMII/TBI	100 Mbps MII

**Table 1 - Capacity of Devices in the ZL50115/16/17/18/19/20 Family**

## 2.0 Description

The ZL5011x family (ZL50115, ZL50116, ZL50117, ZL50118, ZL50119, ZL50120) of CESoP processors are highly functional TDM to Packet bridging devices. The ZL5011x provides both structured and unstructured circuit emulation services (CESoP) for T1 and E1 streams across a packet network based on MPLS, IP or Ethernet. The ZL50117/20 also supports unstructured J2, T3 and E3.

The circuit emulation features in the ZL5011x supports the ITU Recommendation Y.1413 and Y.1453, as well as the CESoP standards from the Metro Ethernet Forum (MEF) and the MPLS and Frame Relay Alliance. The ZL5011x also supports IETF RFC4553 and RFC5086.

The ZL50118/19/20 provides a customer side 100 Mbps MII port to aggregate data traffic with voice traffic to the provider side 1000 Mbps GMII/TBI port, thereby eliminating the need for an external Ethernet switch.

The ZL5011x incorporates a range of powerful clock recovery mechanisms for each TDM stream, allowing the frequency of the source clock to be faithfully generated at the destination, enabling greater system performance and quality. Timing is carried using RTP or similar protocols, and both adaptive and differential clock recovery schemes are included, allowing the customer to choose the correct scheme for the application. An externally supplied clock may also be used to drive the TDM interface of the ZL5011x.

The ZL5011x incur very low latency for the data flow, thereby increasing QoS when carrying voice services across the Packet Switched Network. Voice, when carried using CESoP, which typically has latencies of less than 10 ms, does not require expensive processing such as compression and echo cancellation.

The ZL5011x are cost effective devices aimed at the low density applications such as customer premise routers, IADs, ePON termination and Broadband DLCs. For network systems, the ZL5011x is fully compatible and interoperable with the ZL50110/11/12/14 family.

The ZL5011x is capable of assembling user-defined packets of TDM traffic from the TDM interface and transmitting them out the packet interfaces using a variety of protocols. The ZL5011x supports a range of different packet switched networks, including Ethernet VLANs, IP and MPLS. The devices also supports four different classes of service on packet egress, allowing priority treatment of TDM-based traffic. This can be used to help minimize latency variation in the TDM data.

The ZL5011x can support up to 4 protocol stacks at the same time, provided that each protocol stack can be uniquely identified by a mask & match approach.

Packets received from the packet interfaces are parsed to determine the egress destination, and are appropriately queued to the TDM interface, they can also be forwarded to the host interface, or back toward the packet interface. Packets queued to the TDM interface can be re-ordered based on sequence number, and lost packets filled in to maintain timing integrity.

The ZL5011x includes on-chip memory sufficient for all applications, thereby reducing system costs, board area, power, and design complexity.

A comprehensive evaluation system is available upon request from your local Zarlink representative or distributor. This system includes the CESoP processor, various TDM interfaces and a fully featured evaluation software GUI that will run on a Windows PC.

**3.0 Physical Specification**

The ZL5011x will be packaged in a PBGA device.

**Features:**

- Body Size: 23 mm x 23 mm (typ)
- Ball Count: 324
- Ball Pitch: 1.00 mm (typ)
- Ball Matrix: 22 x 22
- Ball Diameter: 0.60 mm (typ)
- Total Package Thickness: 2.03 mm (typ)

ZL50115 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22			
<b>A</b>	VDD_IO	NC	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID[4]	CPU_DATA[26]	CPU_DATA[24]	GND	CPU_DATA[23]	GND	CPU_DATA[19]	CPU_DATA[12]	CPU_DATA[9]	CPU_DATA[8]	CPU_DATA[7]	CPU_SDAC	VDD_IO			
<b>B</b>	NC	VDD_IO	GND	NC	NC	M0_CRS	M0_RXD[0]	M0_RBC1	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	NC	CPU_DATA[27]	CPU_DATA[22]	CPU_DATA[20]	CPU_DATA[13]	GND	VDD_IO	CPU_TA			
<b>C</b>	NC	GND	VDD_IO	NC	NC	NC	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA[31]	NC	CPU_DATA[29]	CPU_DATA[26]	VDD_IO	GND	CPU_DRE			
<b>D</b>	NC	NC	NC	VDD_IO	NC	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR	M0_RXD[2]	M0_REFCLK	M0_TXD[6]	M0_TXD[1]	VDD_COR	VDD_COR	VDD_IO	M0_ACTIVE_LED	VDD_COR	VDD_IO	CPU_DATA[25]	CPU_ADDR[23]	CPU_DATA[5]			
<b>E</b>	NC	M0_GIGABIT_LED	NC	NC																CPU_DATA[30]	CPU_DATA[21]	CPU_DATA[15]	CPU_DATA[14]		
<b>F</b>	NC	NC	DEVICE_ID[1]	VDD_COR																VDD_COR	CPU_DATA[18]	CPU_DATA[17]	CPU_DATA[16]		
<b>G</b>	M_MDIO	DEVICE_ID[0]	M0_LINKUP_LED	VDD_IO																VDD_IO	CPU_IREQ[11]	CPU_DATA[9]	CPU_DATA[8]		
<b>H</b>	M_MDC	GND	NC	VDD_COR																	CPU_DATA[10]	CPU_DATA[11]	CPU_DATA[9]	NC	
<b>J</b>	NC	NC	NC	VDD_COR				GND	GND	GND	GND	GND	GND	GND							VDD_COR	CPU_DATA[5]	CPU_DATA[3]	CPU_IREQ[5]	
<b>K</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND							GND	CPU_DATA[2]	NC	CPU_DRE[0]	
<b>L</b>	GND	AUX_CLKO	AUX_CLKI	VDD_COR				GND	GND	GND	GND	GND	GND	GND								CPU_CLK	GND	CPU_SDAC	C_VDD_IO
<b>M</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND							GND	CPU_TS_ALE	CPU_WE	CPU_OE	
<b>N</b>	NC	NC	NC	VDD_COR				GND	GND	GND	GND	GND	GND	GND							VDD_IO	CPU_ADDR[22]	CPU_CS	CPU_ADDR[19]	
<b>P</b>	NC	GND	VDD_IO	VDD_COR				GND	GND	GND	GND	GND	GND	GND							VDD_COR	CPU_ADDR[17]	CPU_ADDR[18]	CPU_ADDR[21]	
<b>R</b>	NC	NC	TDM_CLKI[0]	NC																	GND	CPU_ADDR[11]	CPU_ADDR[13]	CPU_ADDR[20]	
<b>T</b>	NC	NC	TDM_FRM_REF	VDD_IO																	VDD_IO	VDD_IO	CPU_ADDR[14]	CPU_ADDR[16]	
<b>U</b>	TDM_STI[0]	VDD_IO	GND	TDM_CLKI[5]																	VDD_COR	JTAG_TMS	CPU_ADDR[15]	CPU_ADDR[12]	
<b>V</b>	TDM_STO[0]	TDM_CLKO[0]	TDM_CLKO_REF	TDM_CLKI[6]																	DEVICE_ID[3]	JTAG_TCK	CPU_ADDR[10]	CPU_ADDR[9]	
<b>W</b>	IC	TDM_CLKI_REF	TDM_FRM_O_REF	VDD_IO	VDD_IO	VDD_COR	VDD_IO	VDD_IO	VDD_COR	PLL_SEC	IC_GND	GND	SYSTEM_CLK	SYSTEM_CVDD_COR	GPIOC[9]	VDD_IO	GPIOC[15]	DEVICE_ID[2]	VDD_IO	JTAG_TDO	CPU_ADDR[4]	CPU_ADDR[8]	CPU_ADDR[8]		
<b>Y</b>	IC	GND	VDD_IO	IC	IC	VDD_COR	IC	IC	PLL_PRI	IC	C_GND	IC	GND	GND	GPIOC[8]	GPIOC[14]	TEST_MODE[1]	JTAG_TRST	C_GND	VDD_IO	GND	CPU_ADDR[7]	CPU_ADDR[7]		
<b>AA</b>	IC	VDD_IO	GND	VDD_IO	VDD_IO	IC	GND	A1VDD_PL1	IC	IC	SYSTEM_DEBUG	SYSTEM_RST	GPIOC[1]	GPIOC[2]	GPIOC[7]	GPIOC[12]	TEST_MODE[0]	JTAG_TDI	C_GND	GND	VDD_IO	CPU_ADDR[6]	CPU_ADDR[6]		
<b>AB</b>	VDD_IO	IC	IC	IC	GND	IC	IC	IC	GPIOC[0]	GPIOC[3]	GPIOC[4]	GPIOC[5]	GPIOC[6]	GPIOC[10]	GPIOC[11]	GPIOC[13]	TEST_MODE[2]	C_GND	CPU_ADDR[2]	CPU_ADDR[3]	CPU_ADDR[5]	VDD_IO	VDD_IO		

Figure 2 - ZL50115 Package View and Ball Positions

ZL50116 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
<b>A</b>	VDD_IO	NC	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID	CPU_DATA	CPU_DATA	GND	CPU_DATA	GND	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_SDAC	VDD_IO
<b>B</b>	NC	VDD_IO	GND	NC	NC	M0_CRS	M0_RXD[0]	M0_RBC0	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	NC	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	GND	VDD_IO	CPU_TA	
<b>C</b>	NC	GND	VDD_IO	NC	NC	NC	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA	NC	CPU_DATA	CPU_DATA	VDD_IO	GND	CPU_DRE	
<b>D</b>	NC	NC	NC	VDD_IO	NC	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR	M0_RXD[2]	M0_REFCLK	M0_TXD[6]	M0_TXD[1]	VDD_COR	VDD_COR	VDD_IO	M0_ACTIVE_LED	VDD_COR	VDD_IO	CPU_DATA	CPU_ADDR	CPU_DATA	
<b>E</b>	NC	M0_GIGABIT_LED	NC	NC															CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	
<b>F</b>	NC	NC	DEVICE_ID	VDD_COR														VDD_COR	CPU_DATA	CPU_DATA	CPU_DATA		
<b>G</b>	M_MDIO	DEVICE_ID	M0_LINKUP_LED	VDD_IO														VDD_IO	CPU_IREQ	CPU_DATA	CPU_DATA		
<b>H</b>	M_MDC	GND	NC	VDD_COR															CPU_DATA	CPU_DATA	CPU_DATA	NC	
<b>J</b>	NC	NC	NC	VDD_COR				GND	GND	GND	GND	GND	GND	GND				VDD_COR	CPU_DATA	CPU_DATA	CPU_IREQ		
<b>K</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND				GND	CPU_DATA	NC	CPU_DRE		
<b>L</b>	GND	AUX_CLKO	AUX_CLKI	VDD_COR				GND	GND	GND	GND	GND	GND	GND					CPU_CLK	GND	CPU_SDAC	C_VDD_IO	
<b>M</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND					GND	CPU_TS_A	CPU_WE	CPU_OE	
<b>N</b>	NC	NC	NC	VDD_COR				GND	GND	GND	GND	GND	GND	GND					VDD_IO	CPU_ADD	CPU_CS	CPU_ADDR	
<b>P</b>	NC	GND	VDD_IO	VDD_COR				GND	GND	GND	GND	GND	GND	GND					VDD_COR	CPU_ADD	CPU_ADDR	CPU_ADDR	
<b>R</b>	NC	TDM_STI	TDM_CLKI	TDM_STOI															GND	CPU_ADD	CPU_ADDR	CPU_ADDR	
<b>T</b>	TDM_CLKI	TDM_CLKO	TDM_FRM	VDD_IO															VDD_IO	VDD_IO	CPU_ADDR	CPU_ADDR	
<b>U</b>	TDM_STI	VDD_IO	GND	TDM_CLKI															VDD_COR	JTAG_TMS	CPU_ADDR	CPU_ADDR	
<b>V</b>	TDM_STOI	TDM_CLKO	TDM_CLKO	TDM_CLKI															DEVICE_ID	JTAG_TCK	CPU_ADDR	CPU_ADDR	
<b>W</b>	C	TDM_CLKI	TDM_FRM	VDD_IO	VDD_IO	VDD_COR	VDD_IO	VDD_COR	PLL_SEC	C_GND	GND	SYSTEM_CLK	VDD_COR	GPIOC[9]	VDD_IO	GPIOC[15]	DEVICE_ID	VDD_IO	JTAG_TDO	CPU_ADDR	CPU_ADDR		
<b>Y</b>	C	GND	VDD_IO	C	C	VDD_COR	C	C	PLL_PRI	C	C_GND	C	GND	GND	GPIOC[8]	GPIOC[14]	TEST_MOD	JTAG_TRS	C_GND	VDD_IO	GND	CPU_ADDR	
<b>AA</b>	C	VDD_IO	GND	VDD_IO	VDD_IO	C	GND	A1VDD_PL	C	C	SYSTEM_DEBUG	SYSTEM_RST	GPIOC[1]	GPIOC[2]	GPIOC[7]	GPIOC[12]	TEST_MOD	JTAG_TDI	C_GND	GND	VDD_IO	CPU_ADDR	
<b>AB</b>	VDD_IO	C	C	C	GND	C	C	C	GPIOC[0]	GPIOC[3]	GPIOC[4]	GPIOC[5]	GPIOC[6]	GPIOC[10]	GPIOC[11]	GPIOC[13]	TEST_MOD	C_GND	CPU_ADD	CPU_ADD	CPU_ADDR	VDD_IO	

Figure 3 - ZL50116 Package View and Ball Positions

ZL50117 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
<b>A</b>	VDD_IO	NC	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID	CPU_DATA	CPU_DATA	GND	CPU_DATA	GND	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	CPU_SDAC	VDD_IO
<b>B</b>	NC	VDD_IO	GND	NC	NC	M0_CRS	M0_RXD[0]	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	NC	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	GND	VDD_IO	CPU_TA		
<b>C</b>	NC	GND	VDD_IO	NC	NC	NC	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA	NC	CPU_DATA	CPU_DATA	VDD_IO	GND	CPU_DRE	
<b>D</b>	NC	NC	NC	VDD_IO	NC	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR	M0_RXD[2]	M0_REFCLK	M0_TXD[6]	M0_TXD[1]	VDD_COR	VDD_COR	VDD_IO	M0_ACTIVE_LED	VDD_COR	VDD_IO	CPU_DATA	CPU_ADDR	CPU_DATA	CPU_DATA
<b>E</b>	NC	M0_GIGABIT_LED	NC	NC															CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	
<b>F</b>	NC	NC	DEVICE_ID	VDD_COR														VDD_COR	CPU_DATA	CPU_DATA	CPU_DATA	CPU_DATA	
<b>G</b>	M_MDIO	DEVICE_ID	M0_LINKUP_LED	VDD_IO														VDD_IO	CPU_IREQ	CPU_DATA	CPU_DATA	CPU_DATA	
<b>H</b>	M_MDC	GND	NC	VDD_COR															CPU_DATA	CPU_DATA	CPU_DATA	NC	
<b>J</b>	NC	NC	NC	VDD_COR				GND	GND	GND	GND	GND	GND	GND				VDD_COR	CPU_DATA	CPU_DATA	CPU_DATA	CPU_IREQ	
<b>K</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND				GND	CPU_DATA	NC	CPU_DRE		
<b>L</b>	GND	AUX_CLKO	AUX_CLKI	VDD_COR				GND	GND	GND	GND	GND	GND	GND					CPU_CLK	GND	CPU_SDAC	C_VDD_IO	
<b>M</b>	TDM_CLKI	TDM_STO	TDM_STI	VDD_IO				GND	GND	GND	GND	GND	GND	GND					GND	CPU_TS_A	CPU_WE	CPU_OE	
<b>N</b>	TDM_STO	TDM_CLKO	TDM_STI	VDD_COR				GND	GND	GND	GND	GND	GND	GND				VDD_IO	CPU_ADD	CPU_CS	CPU_ADDR	CPU_ADDR	
<b>P</b>	TDM_CLKI	GND	VDD_IO	VDD_COR				GND	GND	GND	GND	GND	GND	GND				VDD_COR	CPU_ADD	CPU_ADDR	CPU_ADDR	CPU_ADDR	
<b>R</b>	TDM_CLKO	TDM_STI	TDM_CLKI	TDM_STO														GND	CPU_ADD	CPU_ADDR	CPU_ADDR	CPU_ADDR	
<b>T</b>	TDM_CLKI	TDM_CLKO	TDM_FRM	VDD_IO														VDD_IO	VDD_IO	CPU_ADDR	CPU_ADDR	CPU_ADDR	
<b>U</b>	TDM_STI	VDD_IO	GND	TDM_CLKI														VDD_COR	JTAG_TMS	CPU_ADDR	CPU_ADDR	CPU_ADDR	
<b>V</b>	TDM_STO	TDM_CLKO	TDM_CLKO	TDM_CLKI															DEVICE_ID	JTAG_TCK	CPU_ADDR	CPU_ADDR	CPU_ADDR
<b>W</b>	C	TDM_CLKI	TDM_FRM	VDD_IO	VDD_IO	VDD_COR	VDD_IO	VDD_IO	VDD_COR	PLL_SEC	C_GND	GND	SYSTEM_CLK	VDD_COR	GPIO[9]	VDD_IO	GPIO[15]	DEVICE_ID	VDD_IO	JTAG_TDO	CPU_ADDR	CPU_ADDR	CPU_ADDR
<b>Y</b>	C	GND	VDD_IO	C	C	VDD_COR	C	C	PLL_PRI	C	C_GND	C	GND	GND	GPIO[8]	GPIO[14]	TEST_MOD	JTAG_TRS	C_GND	VDD_IO	GND	CPU_ADDR	CPU_ADDR
<b>AA</b>	C	VDD_IO	GND	VDD_IO	VDD_IO	C	GND	A1VDD_PL	C	C	SYSTEM_DEBUG	SYSTEM_RST	GPIO[1]	GPIO[2]	GPIO[7]	GPIO[12]	TEST_MOD	JTAG_TDI	C_GND	GND	VDD_IO	CPU_ADDR	CPU_ADDR
<b>AB</b>	VDD_IO	C	C	C	GND	C	C	C	GPIO[0]	GPIO[3]	GPIO[4]	GPIO[5]	GPIO[6]	GPIO[10]	GPIO[11]	GPIO[13]	TEST_MOD	C_GND	CPU_ADD	CPU_ADD	CPU_ADD	VDD_IO	

Figure 4 - ZL50117 Package View and Ball Positions

ZL50118 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		
<b>A</b>	VDD_IO	M1_TXEN	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID[4]	CPU_DATA[28]	CPU_DATA[24]	GND	CPU_DATA[23]	GND	CPU_DATA[19]	CPU_DATA[12]	CPU_DATA[9]	CPU_DATA[8]	CPU_DATA[7]	CPU_SDAC[K1]	VDD_IO		
<b>B</b>	M1_TXD[2]	VDD_IO	GND	M1_TXD[0]	M1_TXD[1]	M0_CRS	M0_RXD[0]	M0_RBC0	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	M1_ACTIVATE_LED	CPU_DATA[27]	CPU_DATA[22]	CPU_DATA[20]	CPU_DATA[13]	GND	VDD_IO	CPU_TA		
<b>C</b>	M1_TXD[3]	GND	VDD_IO	M1_RXCLK	M1_COL	M1_TXER	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA[31]	M1_LINKUP_LED	CPU_DATA[29]	CPU_DATA[26]	VDD_IO	GND	CPU_DREQ1		
<b>D</b>	M1_RXD[1]	M1_RXD[0]	M1_RXD[2]	VDD_IO	M1_RXDV	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR[E]	M0_RXD[2]	M0_REFCLK	M0_TXD[8]	M0_TXD[1]	VDD_COR[E]	VDD_COR[E]	VDD_IO	M0_ACTIVATE_LED	VDD_COR[E]	VDD_IO	CPU_DATA[25]	CPU_ADDR[23]	CPU_DATA[6]		
<b>E</b>	M1_RXD[3]	M0_GIGABIT_LED	M1_TXCLK	M1_RXER																CPU_DATA[30]	CPU_DATA[21]	CPU_DATA[15]	CPU_DATA[14]	
<b>F</b>	NC	M1_CRS	DEVICE_ID[1]	VDD_COR[E]																VDD_COR[E]	CPU_DATA[18]	CPU_DATA[17]	CPU_DATA[16]	
<b>G</b>	M_MDIO	DEVICE_ID[0]	M0_LINKUP_LED	VDD_IO																VDD_IO	CPU_IREQ1	CPU_DATA[11]	CPU_DATA[0]	
<b>H</b>	M_MDC	GND	NC	VDD_COR[E]																	CPU_DATA[10]	CPU_DATA[1]	CPU_DATA[4]	IC
<b>J</b>	NC	NC	NC	VDD_COR[E]					GND	GND	GND	GND	GND	GND							VDD_COR[E]	CPU_DATA[5]	CPU_DATA[3]	CPU_IREQ0
<b>K</b>	NC	NC	NC	VDD_IO					GND	GND	GND	GND	GND	GND							GND	CPU_DATA[2]	IC	CPU_DREQ0
<b>L</b>	GND	AUX_CLK0	AUX_CLK1	VDD_COR[E]					GND	GND	GND	GND	GND	GND							CPU_CLK	GND	CPU_SDACK2	IC_VDD_IO
<b>M</b>	NC	NC	NC	VDD_IO					GND	GND	GND	GND	GND	GND							GND	CPU_TS_ALE	CPU_WE	CPU_OE
<b>N</b>	NC	NC	NC	VDD_COR[E]					GND	GND	GND	GND	GND	GND							VDD_IO	CPU_ADDR[22]	CPU_CS	CPU_ADDR[19]
<b>P</b>	NC	GND	VDD_IO	VDD_COR[E]					GND	GND	GND	GND	GND	GND							VDD_COR[E]	CPU_ADDR[17]	CPU_ADDR[18]	CPU_ADDR[21]
<b>R</b>	NC	NC	TDM_CLK[0]	NC																	GND	CPU_ADDR[11]	CPU_ADDR[13]	CPU_ADDR[20]
<b>T</b>	NC	NC	TDM_FRM_REF	VDD_IO																	VDD_IO	VDD_IO	CPU_ADDR[14]	CPU_ADDR[16]
<b>U</b>	TDM_STI[0]	VDD_IO	GND	TDM_CLK[S]																	VDD_COR[E]	JTAG_TMS	CPU_ADDR[15]	CPU_ADDR[12]
<b>V</b>	TDM_STI[0]	TDM_CLK[0]	TDM_CLKO_REF	TDM_CLK[P]																	DEVICE_ID[3]	JTAG_TCK	CPU_ADDR[10]	CPU_ADDR[9]
<b>W</b>	IC	TDM_CLKI_REF	TDM_FRMO_REF	VDD_IO	VDD_IO	VDD_COR[E]	VDD_IO	VDD_IO	VDD_COR[E]	PLL_SEC	IC_GND	GND	SYSTEM_CLK	VDD_COR[E]	GPIO[9]	VDD_IO	GPIO[15]	DEVICE_ID[2]	VDD_IO	JTAG_TDO	CPU_ADDR[4]	CPU_ADDR[8]	CPU_ADDR[8]	
<b>Y</b>	IC	GND	VDD_IO	IC	IC	VDD_COR[E]	IC	IC	PLL_PRI	IC	IC_GND	IC	GND	GND	GPIO[8]	GPIO[14]	TEST_MODE[1]	JTAG_TRST	IC_GND	VDD_IO	GND	CPU_ADDR[7]	CPU_ADDR[7]	
<b>AA</b>	IC	VDD_IO	GND	VDD_IO	VDD_IO	IC	GND	A1VDD_PL1	IC	IC	SYSTEM_DEBUG	SYSTEM_RST	GPIO[1]	GPIO[2]	GPIO[7]	GPIO[12]	TEST_MODE[0]	JTAG_TDI	IC_GND	GND	VDD_IO	CPU_ADDR[6]	CPU_ADDR[6]	
<b>AB</b>	VDD_IO	IC	IC	IC	GND	IC	IC	IC	GPIO[0]	GPIO[3]	GPIO[4]	GPIO[5]	GPIO[6]	GPIO[10]	GPIO[11]	GPIO[13]	TEST_MODE[2]	IC_GND	CPU_ADDR[2]	CPU_ADDR[3]	CPU_ADDR[5]	VDD_IO	VDD_IO	

Figure 5 - ZL50118 Package View and Ball Positions



ZL50119 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22			
<b>A</b>	VDD_IO	M1_TXEN	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID[4]	CPU_DATA[28]	CPU_DATA[24]	GND	CPU_DATA[23]	GND	CPU_DATA[19]	CPU_DATA[12]	CPU_DATA[9]	CPU_DATA[8]	CPU_DATA[7]	CPU_SDAC[K1]	VDD_IO			
<b>B</b>	M1_TXD[2]	VDD_IO	GND	M1_TXD[0]	M1_TXD[1]	M0_CRS	M0_RXD[0]	M0_RBC0	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	M1_ACTIVE_LED	CPU_DATA[27]	CPU_DATA[22]	CPU_DATA[20]	CPU_DATA[13]	GND	VDD_IO	CPU_TA			
<b>C</b>	M1_TXD[3]	GND	VDD_IO	M1_RXCLK	M1_COL	M1_TXER	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA[31]	M1_LINKUP_LED	CPU_DATA[29]	CPU_DATA[26]	VDD_IO	GND	CPU_DREQ1			
<b>D</b>	M1_RXD[1]	M1_RXD[0]	M1_RXD[2]	VDD_IO	M1_RXDV	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR_E	M0_RXD[2]	M0_REFCLK	M0_TXD[8]	M0_TXD[11]	VDD_COR_E	VDD_COR_E	VDD_IO	M0_ACTIVE_LED	VDD_COR_E	VDD_IO	CPU_DATA[25]	CPU_ADDR[23]	CPU_DATA[6]			
<b>E</b>	M1_RXD[3]	M0_GIGABIT_LED	M1_TXCLK	M1_RXER																CPU_DATA[30]	CPU_DATA[21]	CPU_DATA[15]	CPU_DATA[14]		
<b>F</b>	NC	M1_CRS	DEVICE_ID[1]	VDD_COR_E																VDD_COR_E	CPU_DATA[18]	CPU_DATA[17]	CPU_DATA[16]		
<b>G</b>	M_MDIO	DEVICE_ID[0]	M0_LINKUP_LED	VDD_IO																VDD_IO	CPU_IREQ[1]	CPU_DATA[11]	CPU_DATA[0]		
<b>H</b>	M_MDC	GND	NC	VDD_COR_E																	CPU_DATA[10]	CPU_DATA[11]	CPU_DATA[4]	IC	
<b>J</b>	NC	NC	NC	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND	GND						VDD_COR_E	CPU_DATA[5]	CPU_DATA[3]	CPU_IREQ[0]	
<b>K</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND	GND						GND	CPU_DATA[2]	IC	CPU_DREQ[0]	
<b>L</b>	GND	AUX_CLK0	AUX_CLK1	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND	GND						CPU_CLK	GND	CPU_SDAC[R2]	IC_VDD_IO	
<b>M</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND	GND						GND	CPU_TS_ALE	CPU_WE	CPU_OE	
<b>N</b>	NC	NC	NC	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND	GND						VDD_IO	CPU_ADDR[22]	CPU_CS	CPU_ADDR[19]	
<b>P</b>	NC	GND	VDD_IO	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND	GND						VDD_COR_E	CPU_ADDR[17]	CPU_ADDR[18]	CPU_ADDR[21]	
<b>R</b>	NC	TDM_STI[1]	TDM_CLKI[0]	TDM_STOI[1]																	GND	CPU_ADDR[11]	CPU_ADDR[13]	CPU_ADDR[20]	
<b>T</b>	TDM_CLKI[1]	TDM_CLKI[1]	TDM_FRM1_REF	VDD_IO																	VDD_IO	VDD_IO	CPU_ADDR[14]	CPU_ADDR[16]	
<b>U</b>	TDM_STI[0]	VDD_IO	GND	TDM_CLKIS																	VDD_COR_E	JTAG_TMS	CPU_ADDR[15]	CPU_ADDR[12]	
<b>V</b>	TDM_STOI[0]	TDM_CLKI[0]	TDM_CLKO_REF	TDM_CLKIP																		DEVICE_ID[3]	JTAG_TCK	CPU_ADDR[10]	CPU_ADDR[9]
<b>W</b>	IC	TDM_CLKI_REF	TDM_FRMO_REF	VDD_IO	VDD_IO	VDD_COR_E	VDD_IO	VDD_IO	VDD_COR_E	PLL_SEC	IC_GND	GND	SYSTEM_CLK	VDD_COR_E	GPIO[9]	VDD_IO	GPIO[15]	DEVICE_ID[2]	VDD_IO	JTAG_TDO	CPU_ADDR[4]	CPU_ADDR[8]	CPU_ADDR[8]		
<b>Y</b>	IC	GND	VDD_IO	IC	IC	VDD_COR_E	IC	IC	PLL_PRI	IC	IC_GND	IC	GND	GND	GPIO[8]	GPIO[14]	TEST_MODE[1]	JTAG_TRST	IC_GND	VDD_IO	GND	CPU_ADDR[7]			
<b>AA</b>	IC	VDD_IO	GND	VDD_IO	VDD_IO	IC	GND	A1VDD_PL1	IC	IC	SYSTEM_DEBUG	SYSTEM_RST	GPIO[1]	GPIO[2]	GPIO[7]	GPIO[12]	TEST_MODE[0]	JTAG_TDI	IC_GND	GND	VDD_IO	CPU_ADDR[6]	CPU_ADDR[6]		
<b>AB</b>	VDD_IO	IC	IC	IC	GND	IC	IC	IC	GPIO[0]	GPIO[3]	GPIO[4]	GPIO[5]	GPIO[6]	GPIO[10]	GPIO[11]	GPIO[13]	TEST_MODE[2]	IC_GND	CPU_ADDR[2]	CPU_ADDR[3]	CPU_ADDR[5]	VDD_IO			

Figure 6 - ZL50119 Package View and Ball Positions

ZL50120 Package view from TOP side. Note that ball A1 is non-chamfered corner.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
<b>A</b>	VDD_IO	M1_TXEN	M0_TXCLK	M0_RXD[7]	M0_RXD[6]	M0_RXD[4]	M0_COL	M0_GTX_CLK	M0_TXEN	DEVICE_ID[4]	CPU_DATA[28]	CPU_DATA[24]	GND	CPU_DATA[23]	GND	CPU_DATA[19]	CPU_DATA[12]	CPU_DATA[9]	CPU_DATA[8]	CPU_DATA[7]	CPU_SDAC[K1]	VDD_IO	
<b>B</b>	M1_TXD[2]	VDD_IO	GND	M1_TXD[0]	M1_TXD[1]	M0_CRS	M0_RXD[0]	M0_RBC0	M0_RBC0	M0_TXER	GND	M0_TXD[5]	M0_TXD[3]	M0_TXD[2]	M1_ACTIVATE_LED	CPU_DATA[27]	CPU_DATA[22]	CPU_DATA[20]	CPU_DATA[13]	GND	VDD_IO	CPU_TA	
<b>C</b>	M1_TXD[3]	GND	VDD_IO	M1_RXCLK	M1_COL	M1_TXER	M0_RXDV	M0_RXD[3]	M0_RXD[1]	M0_RXCLK	M0_TXD[7]	M0_TXD[4]	M0_TXD[0]	VDD_IO	VDD_IO	CPU_DATA[31]	M1_LINKUP_LED	CPU_DATA[29]	CPU_DATA[26]	VDD_IO	GND	CPU_DREQ1	
<b>D</b>	M1_RXD[1]	M1_RXD[0]	M1_RXD[2]	VDD_IO	M1_RXDV	M0_RXER	VDD_IO	M0_RXD[5]	VDD_COR_E	M0_RXD[2]	M0_REFCLK	M0_TXD[8]	M0_TXD[11]	VDD_COR_E	VDD_COR_E	VDD_IO	M0_ACTIVATE_LED	VDD_COR_E	VDD_IO	CPU_DATA[25]	CPU_ADDR[23]	CPU_DATA[6]	
<b>E</b>	M1_RXD[3]	M0_GIGABIT_LED	M1_TXCLK	M1_RXER																CPU_DATA[30]	CPU_DATA[21]	CPU_DATA[15]	CPU_DATA[14]
<b>F</b>	NC	M1_CRS	DEVICE_ID[1]	VDD_COR_E															VDD_COR_E	CPU_DATA[18]	CPU_DATA[17]	CPU_DATA[16]	
<b>G</b>	M_MDIO	DEVICE_ID[0]	M0_LINKUP_LED	VDD_IO															VDD_IO	CPU_IREQ[1]	CPU_DATA[11]	CPU_DATA[0]	
<b>H</b>	M_MDC	GND	NC	VDD_COR_E																CPU_DATA[10]	CPU_DATA[11]	CPU_DATA[4]	IC
<b>J</b>	NC	NC	NC	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND					VDD_COR_E	CPU_DATA[5]	CPU_DATA[3]	CPU_IREQ[0]	
<b>K</b>	NC	NC	NC	VDD_IO				GND	GND	GND	GND	GND	GND	GND					GND	CPU_DATA[2]	IC	CPU_DREQ0	
<b>L</b>	GND	AUX_CLK0	AUX_CLK1	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND						CPU_CLK	GND	CPU_SDAC[R2]	IC_VDD_IO
<b>M</b>	TDM_CLK[3]	TDM_STO[3]	TDM_STI[3]	VDD_IO				GND	GND	GND	GND	GND	GND	GND						GND	CPU_TS_ALE	CPU_WE	CPU_OE
<b>N</b>	TDM_STO[2]	TDM_CLK[3]	TDM_STI[2]	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND					VDD_IO	CPU_ADDR[22]	CPU_CS	CPU_ADDR[19]	
<b>P</b>	TDM_CLK[2]	GND	VDD_IO	VDD_COR_E				GND	GND	GND	GND	GND	GND	GND					VDD_COR_E	CPU_ADDR[17]	CPU_ADDR[18]	CPU_ADDR[21]	
<b>R</b>	TDM_CLK[2]	TDM_STI[1]	TDM_CLK[0]	TDM_STO[1]															GND	CPU_ADDR[11]	CPU_ADDR[13]	CPU_ADDR[20]	
<b>T</b>	TDM_CLK[1]	TDM_CLK[1]	TDM_FRM_REF	VDD_IO															VDD_IO	VDD_IO	CPU_ADDR[14]	CPU_ADDR[16]	
<b>U</b>	TDM_STI[0]	VDD_IO	GND	TDM_CLK[3]															VDD_COR_E	JTAG_TMS	CPU_ADDR[15]	CPU_ADDR[12]	
<b>V</b>	TDM_STO[0]	TDM_CLK[0]	TDM_FRM_REF	TDM_CLK[0]																DEVICE_ID[3]	JTAG_TCK	CPU_ADDR[10]	CPU_ADDR[9]
<b>W</b>	IC	TDM_CLK_REF	TDM_FRM_O_REF	VDD_IO	VDD_IO	VDD_COR_E	VDD_IO	VDD_IO	VDD_COR_E	PLL_SEC	IC_GND	GND	SYSTEM_CLK	VDD_COR_E	GPIO[9]	VDD_IO	GPIO[15]	DEVICE_ID[2]	VDD_IO	JTAG_TDO	CPU_ADDR[4]	CPU_ADDR[8]	
<b>Y</b>	IC	GND	VDD_IO	IC	IC	VDD_COR_E	IC	IC	PLL_PRI	IC	IC_GND	IC	GND	GND	GPIO[8]	GPIO[14]	TEST_MODE[1]	JTAG_TRST	IC_GND	VDD_IO	GND	CPU_ADDR[7]	
<b>AA</b>	IC	VDD_IO	GND	VDD_IO	VDD_IO	IC	GND	A1VDD_PL1	IC	IC	SYSTEM_DEBUG	SYSTEM_RST	GPIO[1]	GPIO[2]	GPIO[7]	GPIO[12]	TEST_MODE[0]	JTAG_TDI	IC_GND	GND	VDD_IO	CPU_ADDR[6]	
<b>AB</b>	VDD_IO	IC	IC	IC	GND	IC	IC	IC	GPIO[0]	GPIO[3]	GPIO[4]	GPIO[5]	GPIO[6]	GPIO[10]	GPIO[11]	GPIO[13]	TEST_MODE[2]	IC_GND	CPU_ADDR[2]	CPU_ADDR[3]	CPU_ADDR[5]	VDD_IO	

Figure 7 - ZL50120 Package View and Ball Positions

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
A1	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
A10	DEVICE_ID[4]	DEVICE_ID[4]	DEVICE_ID[4]	DEVICE_ID[4]	DEVICE_ID[4]	DEVICE_ID[4]	All
A11	CPU_DATA[28]	CPU_DATA[28]	CPU_DATA[28]	CPU_DATA[28]	CPU_DATA[28]	CPU_DATA[28]	All
A12	CPU_DATA[24]	CPU_DATA[24]	CPU_DATA[24]	CPU_DATA[24]	CPU_DATA[24]	CPU_DATA[24]	All
A13	GND	GND	GND	GND	GND	GND	All
A14	CPU_DATA[23]	CPU_DATA[23]	CPU_DATA[23]	CPU_DATA[23]	CPU_DATA[23]	CPU_DATA[23]	All
A15	GND	GND	GND	GND	GND	GND	All
A16	CPU_DATA[19]	CPU_DATA[19]	CPU_DATA[19]	CPU_DATA[19]	CPU_DATA[19]	CPU_DATA[19]	All
A17	CPU_DATA[12]	CPU_DATA[12]	CPU_DATA[12]	CPU_DATA[12]	CPU_DATA[12]	CPU_DATA[12]	All
A18	CPU_DATA[9]	CPU_DATA[9]	CPU_DATA[9]	CPU_DATA[9]	CPU_DATA[9]	CPU_DATA[9]	All
A19	CPU_DATA[8]	CPU_DATA[8]	CPU_DATA[8]	CPU_DATA[8]	CPU_DATA[8]	CPU_DATA[8]	All
A20	CPU_DATA[7]	CPU_DATA[7]	CPU_DATA[7]	CPU_DATA[7]	CPU_DATA[7]	CPU_DATA[7]	All
A21	CPU_SDACK1	CPU_SDACK1	CPU_SDACK1	CPU_SDACK1	CPU_SDACK1	CPU_SDACK1	All
A22	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
A2	NC	NC	NC	M1_TXEN	M1_TXEN	M1_TXEN	ZL50118/19/20
A3	M0_TXCLK	M0_TXCLK	M0_TXCLK	M0_TXCLK	M0_TXCLK	M0_TXCLK	All
A4	M0_RXD[7]	M0_RXD[7]	M0_RXD[7]	M0_RXD[7]	M0_RXD[7]	M0_RXD[7]	All
A5	M0_RXD[6]	M0_RXD[6]	M0_RXD[6]	M0_RXD[6]	M0_RXD[6]	M0_RXD[6]	All
A6	M0_RXD[4]	M0_RXD[4]	M0_RXD[4]	M0_RXD[4]	M0_RXD[4]	M0_RXD[4]	All
A7	M0_COL	M0_COL	M0_COL	M0_COL	M0_COL	M0_COL	All
A8	M0_GTX_CLK	M0_GTX_CLK	M0_GTX_CLK	M0_GTX_CLK	M0_GTX_CLK	M0_GTX_CLK	All
A9	M0_TXEN	M0_TXEN	M0_TXEN	M0_TXEN	M0_TXEN	M0_TXEN	All
B1	NC	NC	NC	M1_TXD[2]	M1_TXD[2]	M1_TXD[2]	ZL50118/19/20
B10	M0_TXER	M0_TXER	M0_TXER	M0_TXER	M0_TXER	M0_TXER	All
B11	GND	GND	GND	GND	GND	GND	All
B12	M0_TXD[5]	M0_TXD[5]	M0_TXD[5]	M0_TXD[5]	M0_TXD[5]	M0_TXD[5]	All
B13	M0_TXD[3]	M0_TXD[3]	M0_TXD[3]	M0_TXD[3]	M0_TXD[3]	M0_TXD[3]	All
B14	M0_TXD[2]	M0_TXD[2]	M0_TXD[2]	M0_TXD[2]	M0_TXD[2]	M0_TXD[2]	All
B15	NC	NC	NC	M1_ACTIVE_LED	M1_ACTIVE_LED	M1_ACTIVE_LED	ZL50118/19/20
B16	CPU_DATA[27]	CPU_DATA[27]	CPU_DATA[27]	CPU_DATA[27]	CPU_DATA[27]	CPU_DATA[27]	All
B17	CPU_DATA[22]	CPU_DATA[22]	CPU_DATA[22]	CPU_DATA[22]	CPU_DATA[22]	CPU_DATA[22]	All
B18	CPU_DATA[20]	CPU_DATA[20]	CPU_DATA[20]	CPU_DATA[20]	CPU_DATA[20]	CPU_DATA[20]	All
B19	CPU_DATA[13]	CPU_DATA[13]	CPU_DATA[13]	CPU_DATA[13]	CPU_DATA[13]	CPU_DATA[13]	All
B20	GND	GND	GND	GND	GND	GND	All
B21	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
B22	CPU_TA	CPU_TA	CPU_TA	CPU_TA	CPU_TA	CPU_TA	All
B2	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
B3	GND	GND	GND	GND	GND	GND	All
B4	NC	NC	NC	M1_TXD[0]	M1_TXD[0]	M1_TXD[0]	ZL50118/19/20
B5	NC	NC	NC	M1_TXD[1]	M1_TXD[1]	M1_TXD[1]	ZL50118/19/20
B6	M0_CRS	M0_CRS	M0_CRS	M0_CRS	M0_CRS	M0_CRS	All
B7	M0_RXD[0]	M0_RXD[0]	M0_RXD[0]	M0_RXD[0]	M0_RXD[0]	M0_RXD[0]	All
B8	M0_RBC1	M0_RBC1	M0_RBC1	M0_RBC1	M0_RBC1	M0_RBC1	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
B9	M0_RBC0	M0_RBC0	M0_RBC0	M0_RBC0	M0_RBC0	M0_RBC0	All
C1	NC	NC	NC	M1_TXD[3]	M1_TXD[3]	M1_TXD[3]	ZL50118/19/20
C10	M0_RXCLK	M0_RXCLK	M0_RXCLK	M0_RXCLK	M0_RXCLK	M0_RXCLK	All
C11	M0_TXD[7]	M0_TXD[7]	M0_TXD[7]	M0_TXD[7]	M0_TXD[7]	M0_TXD[7]	All
C12	M0_TXD[4]	M0_TXD[4]	M0_TXD[4]	M0_TXD[4]	M0_TXD[4]	M0_TXD[4]	All
C13	M0_TXD[0]	M0_TXD[0]	M0_TXD[0]	M0_TXD[0]	M0_TXD[0]	M0_TXD[0]	All
C14	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
C15	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
C16	CPU_DATA[31]	CPU_DATA[31]	CPU_DATA[31]	CPU_DATA[31]	CPU_DATA[31]	CPU_DATA[31]	All
C17	NC	NC	NC	M1_LINKUP_LED	M1_LINKUP_LED	M1_LINKUP_LED	ZL50118/19/20
C18	CPU_DATA[29]	CPU_DATA[29]	CPU_DATA[29]	CPU_DATA[29]	CPU_DATA[29]	CPU_DATA[29]	All
C19	CPU_DATA[26]	CPU_DATA[26]	CPU_DATA[26]	CPU_DATA[26]	CPU_DATA[26]	CPU_DATA[26]	All
C20	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
C21	GND	GND	GND	GND	GND	GND	All
C22	CPU_DREQ1	CPU_DREQ1	CPU_DREQ1	CPU_DREQ1	CPU_DREQ1	CPU_DREQ1	All
C2	GND	GND	GND	GND	GND	GND	All
C3	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
C4	NC	NC	NC	M1_RXCLK	M1_RXCLK	M1_RXCLK	ZL50118/19/20
C5	NC	NC	NC	M1_COL	M1_COL	M1_COL	ZL50118/19/20
C6	NC	NC	NC	M1_TXER	M1_TXER	M1_TXER	ZL50118/19/20
C7	M0_RXDV	M0_RXDV	M0_RXDV	M0_RXDV	M0_RXDV	M0_RXDV	All
C8	M0_RXD[3]	M0_RXD[3]	M0_RXD[3]	M0_RXD[3]	M0_RXD[3]	M0_RXD[3]	All
C9	M0_RXD[1]	M0_RXD[1]	M0_RXD[1]	M0_RXD[1]	M0_RXD[1]	M0_RXD[1]	All
D1	NC	NC	NC	M1_RXD[1]	M1_RXD[1]	M1_RXD[1]	ZL50118/19/20
D10	M0_RXD[2]	M0_RXD[2]	M0_RXD[2]	M0_RXD[2]	M0_RXD[2]	M0_RXD[2]	All
D11	M0_REFCLK	M0_REFCLK	M0_REFCLK	M0_REFCLK	M0_REFCLK	M0_REFCLK	All
D12	M0_TXD[6]	M0_TXD[6]	M0_TXD[6]	M0_TXD[6]	M0_TXD[6]	M0_TXD[6]	All
D13	M0_TXD[1]	M0_TXD[1]	M0_TXD[1]	M0_TXD[1]	M0_TXD[1]	M0_TXD[1]	All
D14	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
D15	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
D16	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
D17	M0_ACTIVE_LED	M0_ACTIVE_LED	M0_ACTIVE_LED	M0_ACTIVE_LED	M0_ACTIVE_LED	M0_ACTIVE_LED	All
D18	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
D19	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
D20	CPU_DATA[25]	CPU_DATA[25]	CPU_DATA[25]	CPU_DATA[25]	CPU_DATA[25]	CPU_DATA[25]	All
D21	CPU_ADDR[23]	CPU_ADDR[23]	CPU_ADDR[23]	CPU_ADDR[23]	CPU_ADDR[23]	CPU_ADDR[23]	All
D22	CPU_DATA[6]	CPU_DATA[6]	CPU_DATA[6]	CPU_DATA[6]	CPU_DATA[6]	CPU_DATA[6]	All
D2	NC	NC	NC	M1_RXD[0]	M1_RXD[0]	M1_RXD[0]	ZL50118/19/20
D3	NC	NC	NC	M1_RXD[2]	M1_RXD[2]	M1_RXD[2]	ZL50118/19/20
D4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
D5	NC	NC	NC	M1_RXDV	M1_RXDV	M1_RXDV	ZL50118/19/20
D6	M0_RXER	M0_RXER	M0_RXER	M0_RXER	M0_RXER	M0_RXER	All
D7	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
D8	M0_RXD[5]	M0_RXD[5]	M0_RXD[5]	M0_RXD[5]	M0_RXD[5]	M0_RXD[5]	All
D9	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
E1	NC	NC	NC	M1_RXD[3]	M1_RXD[3]	M1_RXD[3]	ZL50118/19/20
E19	CPU_DATA[30]	CPU_DATA[30]	CPU_DATA[30]	CPU_DATA[30]	CPU_DATA[30]	CPU_DATA[30]	All
E20	CPU_DATA[21]	CPU_DATA[21]	CPU_DATA[21]	CPU_DATA[21]	CPU_DATA[21]	CPU_DATA[21]	All
E21	CPU_DATA[15]	CPU_DATA[15]	CPU_DATA[15]	CPU_DATA[15]	CPU_DATA[15]	CPU_DATA[15]	All
E22	CPU_DATA[14]	CPU_DATA[14]	CPU_DATA[14]	CPU_DATA[14]	CPU_DATA[14]	CPU_DATA[14]	All
E2	M0_GIGABIT_LED	M0_GIGABIT_LED	M0_GIGABIT_LED	M0_GIGABIT_LED	M0_GIGABIT_LED	M0_GIGABIT_LED	All
E3	NC	NC	NC	M1_TXCLK	M1_TXCLK	M1_TXCLK	ZL50118/19/20
E4	NC	NC	NC	M1_RXER	M1_RXER	M1_RXER	ZL50118/19/20
F1	NC	NC	NC	NC	NC	NC	All
F19	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
F20	CPU_DATA[18]	CPU_DATA[18]	CPU_DATA[18]	CPU_DATA[18]	CPU_DATA[18]	CPU_DATA[18]	All
F21	CPU_DATA[17]	CPU_DATA[17]	CPU_DATA[17]	CPU_DATA[17]	CPU_DATA[17]	CPU_DATA[17]	All
F22	CPU_DATA[16]	CPU_DATA[16]	CPU_DATA[16]	CPU_DATA[16]	CPU_DATA[16]	CPU_DATA[16]	All
F2	NC	NC	NC	M1_CRS	M1_CRS	M1_CRS	ZL50118/19/20
F3	DEVICE_ID[1]	DEVICE_ID[1]	DEVICE_ID[1]	DEVICE_ID[1]	DEVICE_ID[1]	DEVICE_ID[1]	All
F4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
G1	M_MDIO	M_MDIO	M_MDIO	M_MDIO	M_MDIO	M_MDIO	All
G19	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
G20	CPU_IREQ1	CPU_IREQ1	CPU_IREQ1	CPU_IREQ1	CPU_IREQ1	CPU_IREQ1	All
G21	CPU_DATA[11]	CPU_DATA[11]	CPU_DATA[11]	CPU_DATA[11]	CPU_DATA[11]	CPU_DATA[11]	All
G22	CPU_DATA[0]	CPU_DATA[0]	CPU_DATA[0]	CPU_DATA[0]	CPU_DATA[0]	CPU_DATA[0]	All
G2	DEVICE_ID[0]	DEVICE_ID[0]	DEVICE_ID[0]	DEVICE_ID[0]	DEVICE_ID[0]	DEVICE_ID[0]	All
G3	M0_LINKUP_LED	M0_LINKUP_LED	M0_LINKUP_LED	M0_LINKUP_LED	M0_LINKUP_LED	M0_LINKUP_LED	All
G4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
H1	M_MDC	M_MDC	M_MDC	M_MDC	M_MDC	M_MDC	All
H19	CPU_DATA[10]	CPU_DATA[10]	CPU_DATA[10]	CPU_DATA[10]	CPU_DATA[10]	CPU_DATA[10]	All
H20	CPU_DATA[1]	CPU_DATA[1]	CPU_DATA[1]	CPU_DATA[1]	CPU_DATA[1]	CPU_DATA[1]	All
H21	CPU_DATA[4]	CPU_DATA[4]	CPU_DATA[4]	CPU_DATA[4]	CPU_DATA[4]	CPU_DATA[4]	All
H22	IC	IC	IC	IC	IC	IC	All
H2	GND	GND	GND	GND	GND	GND	All
H3	NC	NC	NC	NC	NC	NC	All
H4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
J1	NC	NC	NC	NC	NC	NC	All
J10	GND	GND	GND	GND	GND	GND	All
J11	GND	GND	GND	GND	GND	GND	All
J12	GND	GND	GND	GND	GND	GND	All
J13	GND	GND	GND	GND	GND	GND	All
J14	GND	GND	GND	GND	GND	GND	All
J19	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
J20	CPU_DATA[5]	CPU_DATA[5]	CPU_DATA[5]	CPU_DATA[5]	CPU_DATA[5]	CPU_DATA[5]	All
J21	CPU_DATA[3]	CPU_DATA[3]	CPU_DATA[3]	CPU_DATA[3]	CPU_DATA[3]	CPU_DATA[3]	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
J22	CPU_IREQ0	CPU_IREQ0	CPU_IREQ0	CPU_IREQ0	CPU_IREQ0	CPU_IREQ0	All
J2	NC	NC	NC	NC	NC	NC	All
J3	NC	NC	NC	NC	NC	NC	All
J4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
J9	GND	GND	GND	GND	GND	GND	All
K1	NC	NC	NC	NC	NC	NC	All
K10	GND	GND	GND	GND	GND	GND	All
K11	GND	GND	GND	GND	GND	GND	All
K12	GND	GND	GND	GND	GND	GND	All
K13	GND	GND	GND	GND	GND	GND	All
K14	GND	GND	GND	GND	GND	GND	All
K19	GND	GND	GND	GND	GND	GND	All
K20	CPU_DATA[2]	CPU_DATA[2]	CPU_DATA[2]	CPU_DATA[2]	CPU_DATA[2]	CPU_DATA[2]	All
K21	IC	IC	IC	IC	IC	IC	All
K22	CPU_DREQ0	CPU_DREQ0	CPU_DREQ0	CPU_DREQ0	CPU_DREQ0	CPU_DREQ0	All
K2	NC	NC	NC	NC	NC	NC	All
K3	NC	NC	NC	NC	NC	NC	All
K4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
K9	GND	GND	GND	GND	GND	GND	All
L1	GND	GND	GND	GND	GND	GND	All
L10	GND	GND	GND	GND	GND	GND	All
L11	GND	GND	GND	GND	GND	GND	All
L12	GND	GND	GND	GND	GND	GND	All
L13	GND	GND	GND	GND	GND	GND	All
L14	GND	GND	GND	GND	GND	GND	All
L19	CPU_CLK	CPU_CLK	CPU_CLK	CPU_CLK	CPU_CLK	CPU_CLK	All
L20	GND	GND	GND	GND	GND	GND	All
L21	CPU_SDACK2	CPU_SDACK2	CPU_SDACK2	CPU_SDACK2	CPU_SDACK2	CPU_SDACK2	All
L22	IC_VDD_IO	IC_VDD_IO	IC_VDD_IO	IC_VDD_IO	IC_VDD_IO	IC_VDD_IO	All
L2	AUX_CLKO	AUX_CLKO	AUX_CLKO	AUX_CLKO	AUX_CLKO	AUX_CLKO	All
L3	AUX_CLKI	AUX_CLKI	AUX_CLKI	AUX_CLKI	AUX_CLKI	AUX_CLKI	All
L4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
L9	GND	GND	GND	GND	GND	GND	All
M1	NC	NC	TDM_CLKI[3]	NC	NC	TDM_CLKI[3]	ZL50117/20
M10	GND	GND	GND	GND	GND	GND	All
M11	GND	GND	GND	GND	GND	GND	All
M12	GND	GND	GND	GND	GND	GND	All
M13	GND	GND	GND	GND	GND	GND	All
M14	GND	GND	GND	GND	GND	GND	All
M19	GND	GND	GND	GND	GND	GND	All
M20	CPU_TS_ALE	CPU_TS_ALE	CPU_TS_ALE	CPU_TS_ALE	CPU_TS_ALE	CPU_TS_ALE	All
M21	CPU_WE	CPU_WE	CPU_WE	CPU_WE	CPU_WE	CPU_WE	All
M22	CPU_OE	CPU_OE	CPU_OE	CPU_OE	CPU_OE	CPU_OE	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
M2	NC	NC	TDM_STO[3]	NC	NC	TDM_STO[3]	ZL50117/20
M3	NC	NC	TDM_STI[3]	NC	NC	TDM_STI[3]	ZL50117/20
M4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
M9	GND	GND	GND	GND	GND	GND	All
N1	NC	NC	TDM_STO[2]	NC	NC	TDM_STO[2]	ZL50117/20
N10	GND	GND	GND	GND	GND	GND	All
N11	GND	GND	GND	GND	GND	GND	All
N12	GND	GND	GND	GND	GND	GND	All
N13	GND	GND	GND	GND	GND	GND	All
N14	GND	GND	GND	GND	GND	GND	All
N19	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
N20	CPU_ADDR[22]	CPU_ADDR[22]	CPU_ADDR[22]	CPU_ADDR[22]	CPU_ADDR[22]	CPU_ADDR[22]	All
N21	CPU_CS	CPU_CS	CPU_CS	CPU_CS	CPU_CS	CPU_CS	All
N22	CPU_ADDR[19]	CPU_ADDR[19]	CPU_ADDR[19]	CPU_ADDR[19]	CPU_ADDR[19]	CPU_ADDR[19]	All
N2	NC	NC	TDM_CLKO[3]	NC	NC	TDM_CLKO[3]	ZL50117/20
N3	NC	NC	TDM_STI[2]	NC	NC	TDM_STI[2]	ZL50117/20
N4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
N9	GND	GND	GND	GND	GND	GND	All
P1	NC	NC	TDM_CLKI[2]	NC	NC	TDM_CLKI[2]	ZL50117/20
P10	GND	GND	GND	GND	GND	GND	All
P11	GND	GND	GND	GND	GND	GND	All
P12	GND	GND	GND	GND	GND	GND	All
P13	GND	GND	GND	GND	GND	GND	All
P14	GND	GND	GND	GND	GND	GND	All
P19	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
P20	CPU_ADDR[17]	CPU_ADDR[17]	CPU_ADDR[17]	CPU_ADDR[17]	CPU_ADDR[17]	CPU_ADDR[17]	All
P21	CPU_ADDR[18]	CPU_ADDR[18]	CPU_ADDR[18]	CPU_ADDR[18]	CPU_ADDR[18]	CPU_ADDR[18]	All
P22	CPU_ADDR[21]	CPU_ADDR[21]	CPU_ADDR[21]	CPU_ADDR[21]	CPU_ADDR[21]	CPU_ADDR[21]	All
P2	GND	GND	GND	GND	GND	GND	All
P3	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
P4	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
P9	GND	GND	GND	GND	GND	GND	All
R1	NC	NC	TDM_CLKO[2]	NC	NC	TDM_CLKO[2]	ZL50117/20
R19	GND	GND	GND	GND	GND	GND	All
R20	CPU_ADDR[11]	CPU_ADDR[11]	CPU_ADDR[11]	CPU_ADDR[11]	CPU_ADDR[11]	CPU_ADDR[11]	All
R21	CPU_ADDR[13]	CPU_ADDR[13]	CPU_ADDR[13]	CPU_ADDR[13]	CPU_ADDR[13]	CPU_ADDR[13]	All
R22	CPU_ADDR[20]	CPU_ADDR[20]	CPU_ADDR[20]	CPU_ADDR[20]	CPU_ADDR[20]	CPU_ADDR[20]	All
R2	NC	TDM_STI[1]	TDM_STI[1]	NC	TDM_STI[1]	TDM_STI[1]	ZL50116/17/19/20
R3	TDM_CLKI[0]	TDM_CLKI[0]	TDM_CLKI[0]	TDM_CLKI[0]	TDM_CLKI[0]	TDM_CLKI[0]	All
R4	NC	TDM_STO[1]	TDM_STO[1]	NC	TDM_STO[1]	TDM_STO[1]	ZL50116/17/19/20
T1	NC	TDM_CLKI[1]	TDM_CLKI[1]	NC	TDM_CLKI[1]	TDM_CLKI[1]	ZL50116/17/19/20
T19	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
T20	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
T21	CPU_ADDR[14]	CPU_ADDR[14]	CPU_ADDR[14]	CPU_ADDR[14]	CPU_ADDR[14]	CPU_ADDR[14]	All
T22	CPU_ADDR[16]	CPU_ADDR[16]	CPU_ADDR[16]	CPU_ADDR[16]	CPU_ADDR[16]	CPU_ADDR[16]	All
T2	NC	TDM_CLKO[1]	TDM_CLKO[1]	NC	TDM_CLKO[1]	TDM_CLKO[1]	ZL50116/17/19/20
T3	TDM_FRMI_REF	TDM_FRMI_REF	TDM_FRMI_REF	TDM_FRMI_REF	TDM_FRMI_REF	TDM_FRMI_REF	All
T4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
U1	TDM_STI[0]	TDM_STI[0]	TDM_STI[0]	TDM_STI[0]	TDM_STI[0]	TDM_STI[0]	All
U19	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
U20	JTAG_TMS	JTAG_TMS	JTAG_TMS	JTAG_TMS	JTAG_TMS	JTAG_TMS	All
U21	CPU_ADDR[15]	CPU_ADDR[15]	CPU_ADDR[15]	CPU_ADDR[15]	CPU_ADDR[15]	CPU_ADDR[15]	All
U22	CPU_ADDR[12]	CPU_ADDR[12]	CPU_ADDR[12]	CPU_ADDR[12]	CPU_ADDR[12]	CPU_ADDR[12]	All
U2	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
U3	GND	GND	GND	GND	GND	GND	All
U4	TDM_CLKiS	TDM_CLKiS	TDM_CLKiS	TDM_CLKiS	TDM_CLKiS	TDM_CLKiS	All
V1	TDM_STO[0]	TDM_STO[0]	TDM_STO[0]	TDM_STO[0]	TDM_STO[0]	TDM_STO[0]	All
V19	DEVICE_ID[3]	DEVICE_ID[3]	DEVICE_ID[3]	DEVICE_ID[3]	DEVICE_ID[3]	DEVICE_ID[3]	All
V20	JTAG_TCK	JTAG_TCK	JTAG_TCK	JTAG_TCK	JTAG_TCK	JTAG_TCK	All
V21	CPU_ADDR[10]	CPU_ADDR[10]	CPU_ADDR[10]	CPU_ADDR[10]	CPU_ADDR[10]	CPU_ADDR[10]	All
V22	CPU_ADDR[9]	CPU_ADDR[9]	CPU_ADDR[9]	CPU_ADDR[9]	CPU_ADDR[9]	CPU_ADDR[9]	All
V2	TDM_CLKO[0]	TDM_CLKO[0]	TDM_CLKO[0]	TDM_CLKO[0]	TDM_CLKO[0]	TDM_CLKO[0]	All
V3	TDM_CLKO_REF	TDM_CLKO_REF	TDM_CLKO_REF	TDM_CLKO_REF	TDM_CLKO_REF	TDM_CLKO_REF	All
V4	TDM_CLKiP	TDM_CLKiP	TDM_CLKiP	TDM_CLKiP	TDM_CLKiP	TDM_CLKiP	All
W1	IC	IC	IC	IC	IC	IC	All
W10	PLL_SEC	PLL_SEC	PLL_SEC	PLL_SEC	PLL_SEC	PLL_SEC	All
W11	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	All
W12	GND	GND	GND	GND	GND	GND	All
W13	SYSTEM_CLK	SYSTEM_CLK	SYSTEM_CLK	SYSTEM_CLK	SYSTEM_CLK	SYSTEM_CLK	All
W14	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
W15	GPIO[9]	GPIO[9]	GPIO[9]	GPIO[9]	GPIO[9]	GPIO[9]	All
W16	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W17	GPIO[15]	GPIO[15]	GPIO[15]	GPIO[15]	GPIO[15]	GPIO[15]	All
W18	DEVICE_ID[2]	DEVICE_ID[2]	DEVICE_ID[2]	DEVICE_ID[2]	DEVICE_ID[2]	DEVICE_ID[2]	All
W19	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W20	JTAG_TDO	JTAG_TDO	JTAG_TDO	JTAG_TDO	JTAG_TDO	JTAG_TDO	All
W21	CPU_ADDR[4]	CPU_ADDR[4]	CPU_ADDR[4]	CPU_ADDR[4]	CPU_ADDR[4]	CPU_ADDR[4]	All
W22	CPU_ADDR[8]	CPU_ADDR[8]	CPU_ADDR[8]	CPU_ADDR[8]	CPU_ADDR[8]	CPU_ADDR[8]	All
W2	TDM_CLKI_REF	TDM_CLKI_REF	TDM_CLKI_REF	TDM_CLKI_REF	TDM_CLKI_REF	TDM_CLKI_REF	All
W3	TDM_FRMO_REF	TDM_FRMO_REF	TDM_FRMO_REF	TDM_FRMO_REF	TDM_FRMO_REF	TDM_FRMO_REF	All
W4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W5	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W6	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
W7	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W8	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
W9	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)



Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
Y1	IC	IC	IC	IC	IC	IC	All
Y10	IC	IC	IC	IC	IC	IC	All
Y11	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	All
Y12	IC	IC	IC	IC	IC	IC	All
Y13	GND	GND	GND	GND	GND	GND	All
Y14	GND	GND	GND	GND	GND	GND	All
Y15	GPIO[8]	GPIO[8]	GPIO[8]	GPIO[8]	GPIO[8]	GPIO[8]	All
Y16	GPIO[14]	GPIO[14]	GPIO[14]	GPIO[14]	GPIO[14]	GPIO[14]	All
Y17	TEST_MODE[1]	TEST_MODE[1]	TEST_MODE[1]	TEST_MODE[1]	TEST_MODE[1]	TEST_MODE[1]	All
Y18	JTAG_TRST	JTAG_TRST	JTAG_TRST	JTAG_TRST	JTAG_TRST	JTAG_TRST	All
Y19	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	All
Y20	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
Y21	GND	GND	GND	GND	GND	GND	All
Y22	CPU_ADDR[7]	CPU_ADDR[7]	CPU_ADDR[7]	CPU_ADDR[7]	CPU_ADDR[7]	CPU_ADDR[7]	All
Y2	GND	GND	GND	GND	GND	GND	All
Y3	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
Y4	IC	IC	IC	IC	IC	IC	All
Y5	IC	IC	IC	IC	IC	IC	All
Y6	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	VDD_CORE	All
Y7	IC	IC	IC	IC	IC	IC	All
Y8	IC	IC	IC	IC	IC	IC	All
Y9	PLL_PRI	PLL_PRI	PLL_PRI	PLL_PRI	PLL_PRI	PLL_PRI	All
AA1	IC	IC	IC	IC	IC	IC	All
AA10	IC	IC	IC	IC	IC	IC	All
AA11	SYSTEM_DEBUG	SYSTEM_DEBUG	SYSTEM_DEBUG	SYSTEM_DEBUG	SYSTEM_DEBUG	SYSTEM_DEBUG	All
AA12	SYSTEM_RST	SYSTEM_RST	SYSTEM_RST	SYSTEM_RST	SYSTEM_RST	SYSTEM_RST	All
AA13	GPIO[1]	GPIO[1]	GPIO[1]	GPIO[1]	GPIO[1]	GPIO[1]	All
AA14	GPIO[2]	GPIO[2]	GPIO[2]	GPIO[2]	GPIO[2]	GPIO[2]	All
AA15	GPIO[7]	GPIO[7]	GPIO[7]	GPIO[7]	GPIO[7]	GPIO[7]	All
AA16	GPIO[12]	GPIO[12]	GPIO[12]	GPIO[12]	GPIO[12]	GPIO[12]	All
AA17	TEST_MODE[0]	TEST_MODE[0]	TEST_MODE[0]	TEST_MODE[0]	TEST_MODE[0]	TEST_MODE[0]	All
AA18	JTAG_TDI	JTAG_TDI	JTAG_TDI	JTAG_TDI	JTAG_TDI	JTAG_TDI	All
AA19	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	All
AA20	GND	GND	GND	GND	GND	GND	All
AA21	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AA22	CPU_ADDR[6]	CPU_ADDR[6]	CPU_ADDR[6]	CPU_ADDR[6]	CPU_ADDR[6]	CPU_ADDR[6]	All
AA2	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AA3	GND	GND	GND	GND	GND	GND	All
AA4	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AA5	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AA6	IC	IC	IC	IC	IC	IC	All
AA7	GND	GND	GND	GND	GND	GND	All
AA8	A1VDD_PLL1	A1VDD_PLL1	A1VDD_PLL1	A1VDD_PLL1	A1VDD_PLL1	A1VDD_PLL1	All

Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)

Ball #	ZL50115 Signal Name	ZL50116 Signal Name	ZL50117 Signal Name	ZL50118 Signal Name	ZL50119 Signal Name	ZL50120 Signal Name	Variant
AA9	IC	IC	IC	IC	IC	IC	All
AB1	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AB10	GPIO[3]	GPIO[3]	GPIO[3]	GPIO[3]	GPIO[3]	GPIO[3]	All
AB11	GPIO[4]	GPIO[4]	GPIO[4]	GPIO[4]	GPIO[4]	GPIO[4]	All
AB12	GPIO[5]	GPIO[5]	GPIO[5]	GPIO[5]	GPIO[5]	GPIO[5]	All
AB13	GPIO[6]	GPIO[6]	GPIO[6]	GPIO[6]	GPIO[6]	GPIO[6]	All
AB14	GPIO[10]	GPIO[10]	GPIO[10]	GPIO[10]	GPIO[10]	GPIO[10]	All
AB15	GPIO[11]	GPIO[11]	GPIO[11]	GPIO[11]	GPIO[11]	GPIO[11]	All
AB16	GPIO[13]	GPIO[13]	GPIO[13]	GPIO[13]	GPIO[13]	GPIO[13]	All
AB17	TEST_MODE[2]	TEST_MODE[2]	TEST_MODE[2]	TEST_MODE[2]	TEST_MODE[2]	TEST_MODE[2]	All
AB18	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	IC_GND	All
AB19	CPU_ADDR[2]	CPU_ADDR[2]	CPU_ADDR[2]	CPU_ADDR[2]	CPU_ADDR[2]	CPU_ADDR[2]	All
AB20	CPU_ADDR[3]	CPU_ADDR[3]	CPU_ADDR[3]	CPU_ADDR[3]	CPU_ADDR[3]	CPU_ADDR[3]	All
AB21	CPU_ADDR[5]	CPU_ADDR[5]	CPU_ADDR[5]	CPU_ADDR[5]	CPU_ADDR[5]	CPU_ADDR[5]	All
AB22	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	VDD_IO	All
AB2	IC	IC	IC	IC	IC	IC	All
AB3	IC	IC	IC	IC	IC	IC	All
AB4	IC	IC	IC	IC	IC	IC	All
AB5	GND	GND	GND	GND	GND	GND	All
AB6	IC	IC	IC	IC	IC	IC	All
AB7	IC	IC	IC	IC	IC	IC	All
AB8	IC	IC	IC	IC	IC	IC	All
AB9	GPIO[0]	GPIO[0]	GPIO[0]	GPIO[0]	GPIO[0]	GPIO[0]	All

**Table 2 - ZL50115/16/17/18/19/20 Ball Signal Assignment (continued)**

NC - Not Connected - leave open circuit.

IC - Internally Connected - leave open circuit.

IC\_GND - Internally Connected - tie to ground

IC\_VDD\_IO - Internally Connected - tie to VDD\_IO

## 4.0 External Interface Description

The following key applies to all tables:

I	Input
O	Output
D	Internal 100 k $\Omega$ pull-down resistor present
U	Internal 100 k $\Omega$ pull-up resistor present
T	Tri-state Output

### 4.1 TDM Interface

All TDM Interface signals are 5 V tolerant.

All TDM Interface inputs (including data, clock and frame pulse) have internal pull-down resistors so they can be safely left unconnected if not used.

#### 4.1.1 TDM Stream Connections

There are three interfaces possible among the ZL5011x.

The ZL50117/20 supports four TDM ports [3:0] at 2 Mbps, or one TDM port [0] at 8 Mbps or one unstructured TDM port [0] for J2/E3/T3.

The ZL50116/19 supports two TDM ports [1:0] at 2 Mbps, or one TDM port [0] at 8 Mbps (up to 64 DS0).

The ZL50115/18 supports one TDM port [0] at 2 Mbps, or one TDM port [0] at 8 Mbps (up to 32 DS0)

Signal	I/O	Package Balls	Description
TDM_STi[3:0]	I D	[3] M3 [2] N3 [1] R2 [0] U1	TDM port serial data input streams. For different standards these pins are given different identities: ST-BUS: TDM_STi[3:0] H.110: TDM_D[3:0] H-MVIP: TDM_HDS[3:0] Triggered on rising edge or falling edge depending on standard. At 8.192 Mbps only stream [0] is used. Stream [0] is used for unstructured J2 or T3/E3 on the ZL50117/20.

**Table 3 - TDM Interface Stream Pin Definition**

Signal	I/O	Package Balls	Description
TDM_STo[3:0]	OT	[3] M2 [2] N1 [1] R4 [0] V1	TDM port serial data output streams. For different standards these pins are given different identities: ST-BUS: TDM_STo[3:0] H.110: TDM_D[3:0] H-MVIP: TDM_HDS[3:0] Triggered on rising edge or falling edge depending on standard. At 8.192 Mbps only stream [0] is used. Stream [0] is used for unstructured J2 or T3/E3 on the ZL50117/20.
TDM_CLKi[3:0]	ID	[3] M1 [2] P1 [1] T1 [0] R3	TDM port clock inputs programmable as active high or low. Can accept frequencies of 1.544 MHz, 2.048 MHz, 4.096 MHz, 8.192 MHz, 6.312 MHz or 16.384 MHz depending on standard used. At 8.192 Mbps only stream [0] is used. Stream [0] is used for unstructured J2 or T3/E3 on the ZL50117/20.
TDM_CLKo[3:0]	O	[3] N2 [2] R1 [1] T2 [0] V2	TDM port clock outputs. Will generate 1.544 MHz, 2.048 MHz, 4.096 MHz, 6.312 MHz, 8.192 MHz or 16.384 MHz depending on standard used. At 8.192 Mbps only stream [0] is used. Stream [0] is used for unstructured J2 or T3/E3 on the ZL50117/20.

**Table 3 - TDM Interface Stream Pin Definition**

Note: Speed modes:  
 2.048 Mbps - 32 channels per stream.  
 8.192 Mbps - 128 channels per stream.  
 J2 - 98 channels per stream  
 E3 - 537 channels per stream  
 T3 - 699 channels per stream

Note: All TDM Interface inputs (including data, clock and frame pulse) have internal pull-down resistors so they can be safely left unconnected if not used.

#### 4.1.2 TDM Signals common to ZL50115/16/17/18/19/20

Signal	I/O	Package Balls	Description
TDM_CLKi_REF	ID	W2	TDM port reference clock input for backplane operation
TDM_CLKo_REF	O	V3	TDM port reference clock output for backplane operation

**Table 4 - TDM Interface Common Pin Definition**

Signal	I/O	Package Balls	Description
TDM_FRMi_REF	I D	T3	TDM port reference frame input. For different standards this pin is given a different identity: ST-BUS: TDM_F0i H.110: TDM_FRAME H-MVIP: TDM_F0 Signal is normally active low, but can be active high depending on standard. Indicates the start of a TDM frame by pulsing every 125 $\mu$ s. Normally will straddle rising edge or falling edge of clock pulse, depending on standard and clock frequency.
TDM_FRMo_REF	O	W3	TDM port reference frame output. For different standards this pin is given a different identity: ST-BUS: TDM_F0o H.110: TDM_FRAME H-MVIP: TDM_F0 Signal is normally active low, but can be active high depending on standard. Indicates the start of a TDM frame by pulsing every 125 $\mu$ s. Normally will straddle rising edge or falling edge of clock pulse, depending on standard and clock frequency.
AUX_CLKI	I D	L3	Auxiliary clock input. Typically connected to AUX_CLKO.
AUX_CLKO	OT	L2	Auxiliary clock output. Typically connected to AUX_CLKI.

Table 4 - TDM Interface Common Pin Definition

## 4.2 PAC Interface

All PAC Interface signals are 5 V tolerant.

All PAC Interface outputs are high impedance while System Reset is LOW.

Signal	I/O	Package Balls	Description
TDM_CLKiP	I D	V4	Primary reference clock input. Should be driven by external clock source to provide locking reference to internal / optional external DPLL in TDM master mode. Also provides PRS clock for RTP timestamps in synchronous modes. Acceptable frequency range: 8 kHz - 34.368 MHz (generally should be between 10 MHz and 25 MHz as per ITU-T Y.1413).
TDM_CLKiS	I D	U4	Secondary reference clock input. Backup external reference for automatic switch-over in case of failure of TDM_CLKiP source.
PLL_PRI	OT	Y9	Primary reference output to optional external DPLL. Multiplexed & frequency divided reference output for support of optional external DPLL. Expected frequency range: 8 kHz - 16.384 MHz.
PLL_SEC	OT	W10	Secondary reference output to optional external DPLL Multiplexed & frequency divided reference output for support of optional external DPLL. Expected frequency range: 8 kHz - 16.384 MHz.

**Table 5 - PAC Interface Package Ball Definition**

### 4.3 Packet Interfaces

For the ZL50118/19/20 variants the packet interface is capable of either 2 MII interfaces, or 1 MII and 1 GMII interfaces, or 1 MII and 1 TBI (1000 Mbps) interfaces. The TBI interface is a PCS interface supported by an integrated 1000BASE-X PCS module. When the packet interface is programmed for PCS/TBI mode, by default the hardware will not enable auto-negotiation. The TBI auto-negotiation must be done by application software. The ZL50118/19/20 supports Port 0 and Port 1.

For the ZL50115/16/17 variants the packet interface is capable of 1 MII or 1 GMII or 1 TBI (1000 Mbps) interface. The TBI interface is a PCS interface supported by an integrated 1000BASE-X PCS module. The ZL50115/16/17 supports Port 0.

Data for all three types of packet switching is based on Specification IEEE Std. 802.3 - 2000. Only Port 0 has the 1000 Mbps capability necessary for the GMII/TBI interface.

The ZL5011x will not take action when receiving a PAUSE frame. It will not pause the transmission of traffic. It is normally not required to stop CESoP traffic because it is generally constant bit rate and time sensitive. If necessary, the limiting of egress non-CESoP traffic may be done external to the ZL5011x (e.g., in an Ethernet switch).

Table 6 maps the signal pins used in the MII interface to those used in the GMII and TBI interface. Table 7 shows MII Management Interface Package Ball Definition. Table 8 and Table 9 show respectively the MII Port 0 and Port 1 Interface Package Ball Definition.

All Packet Interface signals are 5 V tolerant, and all outputs are high impedance while System Reset is LOW.

MII	GMII	TBI (PCS)
Mn_LINKUP_LED	Mn_LINKUP_LED	Mn_LINKUP_LED
Mn_ACTIVE_LED	Mn_ACTIVE_LED	Mn_ACTIVE_LED
-	Mn_GIGABIT_LED	Mn_GIGABIT_LED
-	Mn_REFCLK	Mn_REFCLK
Mn_RXCLK	Mn_RXCLK	Mn_RBC0
Mn_COL	Mn_COL	Mn_RBC1
Mn_RXD[3:0]	Mn_RXD[7:0]	Mn_RXD[7:0]
Mn_RXDV	Mn_RXDV	Mn_RXD[8]
Mn_RXER	Mn_RXER	Mn_RXD[9]
Mn_CRS	Mn_CRS	Mn_Signal_Detect
Mn_TXCLK	-	-
Mn_TXD[3:0]	Mn_TXD[7:0]	Mn_TXD[7:0]
Mn_TXEN	Mn_TXEN	Mn_TXD[8]
Mn_TXER	Mn_TXER	Mn_TXD[9]
-	Mn_GTX_CLK	Mn_GTX_CLK

**Table 6 - Packet Interface Signal Mapping - MII to GMII/TBI**

Note: Mn can be either M0 or M1 for ZL5011x variants.

Signal	I/O	Package Balls	Description
M_MDC	O	H1	MII management data clock. Common for all four MII ports. It has a minimum period of 400 ns (maximum freq. 2.5 MHz), and is independent of the TXCLK and RXCLK.
M_MDIO	ID/ OT	G1	MII management data I/O. Common for all four MII ports at up to 2.5 MHz. It is bi-directional between the ZL5011x and the Ethernet station management entity. Data is passed synchronously with respect to M_MDC.

Table 7 - MII Management Interface Package Ball Definition

MII Port 0			
Signal	I/O	Package Balls	Description
M0_LINKUP_LED	O	G3	LED drive for MAC 0 to indicate port is linked up. Logic 0 output = LED on Logic 1 output = LED off
M0_ACTIVE_LED	O	D17	LED drive for MAC 0 to indicate port is transmitting or receiving packet data. Logic 0 output = LED on Logic 1 output = LED off
M0_GIGABIT_LED	O	E2	LED drive for MAC 0 to indicate operation at Gbps. Logic 0 output = LED on Logic 1 output = LED off
M0_REFCLK	I D	D11	<b>GMII/TBI</b> - Reference Clock input at 125 MHz. Can be used to lock receive circuitry (RX) to M0_GTXCLK rather than recovering the RXCLK (or RBC0 and RBC1). Useful, for example, in the absence of valid serial data. <b>NOTE:</b> In MII mode this pin must be driven with the same clock as M0_RXCLK.
M0_RXCLK	I U	C10	GMII/MII - M0_RXCLK. Accepts the following frequencies: 25.0 MHz    MII    100 Mbps 125.0 MHz   GMII 1 Gbps

Table 8 - MII Port 0 Interface Package Ball Definition



MII Port 0			
Signal	I/O	Package Balls	Description
M0_RBC0	I U	B9	TBI - M0_RBC0. Used as a clock when in TBI mode. Accepts 62.5 MHz and is 180° out of phase with M0_RBC1. Receive data is clocked at each rising edge of M0_RBC1 and M0_RBC0, resulting in 125 MHz sample rate.
M0_RBC1	I U	B8	TBI - M0_RBC1 Used as a clock when in TBI mode. Accepts 62.5 MHz, and is 180° out of phase with M0_RBC0. Receive data is clocked at each rising edge of M0_RBC1 and M0_RBC0, resulting in 125 MHz sample rate.
M0_COL	I D	A7	GMII/MII - M0_COL. Collision Detection. This signal is independent of M0_TXCLK and M0_RXCLK, and is asserted when a collision is detected on an attempted transmission. It is active high, and only specified for half-duplex operation.
M0_RXD[7:0]	I U	[7] A4 [3] C8 [6] A5 [2] D10 [5] D8 [1] C9 [4] A6 [0] B7	Receive Data. Only half the bus (bits [3:0]) are used in MII mode. Clocked on rising edge of M0_RXCLK (GMII/MII) or the rising edges of M0_RBC0 and M0_RBC1 (TBI).
M0_RXDV / M0_RXD[8]	I D	C7	GMII/MII - M0_RXDV Receive Data Valid. Active high. This signal is clocked on the rising edge of M0_RXCLK. It is asserted when valid data is on the M0_RXD bus. TBI - M0_RXD[8] Receive Data. Clocked on the rising edges of M0_RBC0 and M0_RBC1.
M0_RXER / M0_RXD[9]	I D	D6	GMII/MII - M0_RXER Receive Error. Active high signal indicating an error has been detected. Normally valid when M0_RXDV is asserted. Can be used in conjunction with M0_RXD when M0_RXDV signal is de-asserted to indicate a False Carrier. TBI - M0_RXD[9] Receive Data. Clocked on the rising edges of M0_RBC0 and M0_RBC1.

Table 8 - MII Port 0 Interface Package Ball Definition (continued)

MII Port 0			
Signal	I/O	Package Balls	Description
M0_CRCS / M0_Signal_Detect	I D	B6	GMII/MII - M0_CRCS Carrier Sense. This asynchronous signal is asserted when either the transmission or reception device is non-idle. It is active high. TBI - M0_Signal Detect Similar function to M0_CRCS.
M0_TXCLK	I U	A3	<b>MII only</b> - Transmit Clock Accepts the following frequencies: 25.0 MHz    MII    100 Mbps
M0_TXD[7:0]	O	[7] C11    [3] B13 [6] D12    [2] B14 [5] B12    [1] D13 [4] C12    [0] C13	Transmit Data. Only half the bus (bits [3:0]) are used in MII mode. Clocked on rising edge of M0_TXCLK (MII) or the rising edge of M0_GTXCLK (GMII/TBI).
M0_TXEN / M0_TXD[8]	O	A9	GMII/MII - M0_TXEN Transmit Enable. Asserted when the MAC has data to transmit, synchronously to M0_TXCLK with the first pre-amble of the packet to be sent. Remains asserted until the end of the packet transmission. Active high. TBI - M0_TXD[8] Transmit Data. Clocked on rising edge of M0_GTXCLK.
M0_TXER / M0_TXD[9]	O	B10	GMII/MII - M0_TXER Transmit Error. Transmitted synchronously with respect to M0_TXCLK, and active high. When asserted (with M0_TXEN also asserted) the ZL5011x will transmit a non-valid symbol, somewhere in the transmitted frame. TBI - M0_TXD[9] Transmit Data. Clocked on rising edge of M0_GTXCLK.
M0_GTX_CLK	O	A8	<b>GMII/TBI only</b> - Gigabit Transmit Clock Output of a clock for Gigabit operation at 125 MHz.

Table 8 - MII Port 0 Interface Package Ball Definition (continued)

MII Port 1 (ZL50118/19/20 only)			
Signal	I/O	Package Balls	Description
M1_LINKUP_LED	O	C17	LED drive for MAC 1 to indicate port is linked up. Logic 0 output = LED on Logic 1 output = LED off
M1_ACTIVE_LED	O	B15	LED drive for MAC 1 to indicate port is transmitting or receiving packet data. Logic 0 output = LED on Logic 1 output = LED off
M1_RXCLK	I U	C4	MII only - Receive Clock. Accepts the following frequencies: 25.0 MHz    MII    100 Mbps
M1_COL	I D	C5	Collision Detection. This signal is independent of M1_TXCLK and M1_RXCLK, and is asserted when a collision is detected on an attempted transmission. It is active high, and only specified for half-duplex operation.
M1_RXD[3:0]	I U	[3] E1            [1] D1 [2] D3            [0] D2	Receive Data. Clocked on rising edge of M1_RXCLK.
M1_RXDV	I D	D5	Receive Data Valid. Active high. This signal is clocked on the rising edge of M1_RXCLK. It is asserted when valid data is on the M1_RXD bus.
M1_RXER	I D	E4	Receive Error. Active high signal indicating an error has been detected. Normally valid when M1_RXDV is asserted. Can be used in conjunction with M1_RXD when M1_RXDV signal is de-asserted to indicate a False Carrier.
M1_CRS	I D	F2	Carrier Sense. This asynchronous signal is asserted when either the transmission or reception device is non-idle. It is active high.
M1_TXCLK	I U	E3	MII only - Transmit Clock Accepts the following frequencies: 25.0 MHz    MII    100 Mbps
M1_TXD[3:0]	O	[3] C1            [1] B5 [2] B1            [0] B4	Transmit Data. Clocked on rising edge of M1_TXCLK.
M1_TXEN	O	A2	Transmit Enable. Asserted when the MAC has data to transmit, synchronously to M1_TXCLK with the first pre-amble of the packet to be sent. Remains asserted until the end of the packet transmission. Active high.

Table 9 - MII Port 1 Interface Package Ball Definition

MII Port 1 (ZL50118/19/20 only)			
Signal	I/O	Package Balls	Description
M1_TXER	O	C6	Transmit Error. Transmitted synchronously with respect to M1_TXCLK, and active high. When asserted (with M1_TXEN also asserted) the ZL5011x will transmit a non-valid symbol, somewhere in the transmitted frame.

Table 9 - MII Port 1 Interface Package Ball Definition (continued)

#### 4.4 CPU Interface

All CPU Interface signals are 5 V tolerant.

All CPU Interface outputs are high impedance while System Reset is LOW.

Signal	I/O	Package Balls	Description
CPU_DATA[31:0]	I/OT	[31] C16      [15] E21 [30] E19      [14] E22 [29] C18      [13] B19 [28] A11      [12] A17 [27] B16      [11] G21 [26] C19      [10] H19 [25] D20      [9] A18 [24] A12      [8] A19 [23] A14      [7] A20 [22] B17      [6] D22 [21] E20      [5] J20 [20] B18      [4] H21 [19] A16      [3] J21 [18] F20      [2] K20 [17] F21      [1] H20 [16] F22      [0] G22	CPU Data Bus. Bi-directional data bus, synchronously transmitted with CPU_CLK rising edge.  NOTE: as with all ports in the ZL5011x device, CPU_DATA[0] is the least significant bit (lsb).
CPU_ADDR[23:2]	I	[23] D21      [11] R20 [22] N20      [10] V21 [21] P22      [9] V22 [20] R22      [8] W22 [19] N22      [7] Y22 [18] P21      [6] AA22 [17] P20      [5] AB21 [16] T22      [4] W21 [15] U21      [3] AB20 [14] T21      [2] AB19 [13] R21 [12] U22	CPU Address Bus. Address input from processor to ZL5011x, synchronously transmitted with CPU_CLK rising edge.  NOTE: as with all ports in the ZL5011x device, CPU_ADDR[2] is the least significant bit (lsb).

Table 10 - CPU Interface Package Ball Definition

Signal	I/O	Package Balls	Description
$\overline{\text{CPU\_CS}}$	I U	N21	CPU Chip Select. Synchronous to rising edge of CPU_CLK and active low. Is asserted with $\overline{\text{CPU\_TS\_ALE}}$ . Must be asserted with CPU_OE to asynchronously enable the CPU_DATA output during a read, including DMA read.
$\overline{\text{CPU\_WE}}$	I	M21	CPU Write Enable. Synchronously asserted with respect to CPU_CLK rising edge, and active low. Used for CPU writes from the processor to registers within the ZL5011x. Asserted one clock cycle after $\overline{\text{CPU\_TS\_ALE}}$ .
CPU_OE	I	M22	CPU Output Enable. Synchronously asserted with respect to CPU_CLK rising edge, and active low. Used for CPU reads from the processor to registers within the ZL5011x. Asserted one clock cycle after $\overline{\text{CPU\_TS\_ALE}}$ . Must be asserted with $\overline{\text{CPU\_CS}}$ to asynchronously enable the CPU_DATA output during a read, including DMA read.
$\overline{\text{CPU\_TS\_ALE}}$	I	M20	Synchronous input with rising edge of CPU_CLK. Latch Enable (ALE), active high signal. Asserted with $\overline{\text{CPU\_CS}}$ , for a single clock cycle.
$\overline{\text{CPU\_SDACK1}}$	I	A21	CPU/DMA 1 Acknowledge Input. Active low synchronous to CPU_CLK rising edge. Used to acknowledge request from ZL5011x for a DMA write transaction. Only used for DMA transfers, not for normal register access.
$\overline{\text{CPU\_SDACK2}}$	I	L21	CPU/DMA 2 Acknowledge Input Active low synchronous to CPU_CLK rising edge. Used to acknowledge request from ZL5011x for a DMA read transaction. Only used for DMA transfers, not for normal register access.
CPU_CLK	I	L19	CPU PowerQUICC™ II Bus Interface clock input. 66 MHz clock, with minimum of 6 ns high/low time. Used to time all host interface signals into and out of ZL5011x device.

Table 10 - CPU Interface Package Ball Definition (continued)

Signal	I/O	Package Balls	Description
$\overline{\text{CPU\_TA}}$	OT	B22	CPU Transfer Acknowledge. Driven from tri-state condition on the negative clock edge of CPU_CLK following the assertion of CPU_CS. Active low, asserted from the rising edge of CPU_CLK. For a read, asserted when valid data is available at CPU_DATA. The data is then read by the host on the following rising edge of CPU_CLK. For a write, is asserted when the ZL5011x is ready to accept data from the host. The data is written on the rising edge of CPU_CLK following the assertion. Returns to tri-state from the negative clock edge of CPU_CLK following the de-assertion of CPU_CS.
$\overline{\text{CPU\_DREQ0}}$	OT	K22	CPU DMA 0 Request Output Active low synchronous to CPU_CLK rising edge. Asserted by ZL5011x to request the host initiates a DMA write. Only used for DMA transfers, not for normal register access.
$\overline{\text{CPU\_DREQ1}}$	OT	C22	CPU DMA 1 Request Active low synchronous to CPU_CLK rising edge. Asserted by ZL5011x to indicate packet data is ready for transmission to the CPU, and request the host initiates a DMA read. Only used for DMA transfers, not for normal register access.
$\overline{\text{CPU\_IREQ0}}$	O	J22	CPU Interrupt 0 Request (Active Low)
$\overline{\text{CPU\_IREQ1}}$	O	G20	CPU Interrupt 1 Request (Active Low)

Table 10 - CPU Interface Package Ball Definition (continued)

## 4.5 System Function Interface

All System Function Interface signals are 5 V tolerant.

The core of the chip will be held in reset for 16383 SYSTEM\_CLK cycles after SYSTEM\_RST has gone HIGH to allow the PLL's to lock. No chip access should occur at this time.

Signal	I/O	Package Balls	Description
SYSTEM_CLK	I	W13	System Clock Input. The system clock frequency is 100 MHz. The quality of SYSTEM_CLK, or the oscillator that drives SYSTEM_CLK directly impacts the adaptive clock recovery performance. See Section 6.3.
SYSTEM_RST	I	AA12	System Reset Input. Active low. The system reset is asynchronous, and causes all registers within the ZL5011x to be reset to their default state. Recommend external pull-up.
SYSTEM_DEBUG	I	AA11	System Debug Enable. This is an asynchronous signal that, when de-asserted, prevents the software assertion of the debug-freeze command, regardless of the internal state of registers, or any error conditions. Active high. Recommend external pull-down.

**Table 11 - System Function Interface Package Ball Definition**

## 4.6 Test Facilities

### 4.6.1 Administration, Control and Test Interface

All Administration, Control and Test Interface signals are 5 V tolerant.

Signal	I/O	Package Balls	Description
GPIO[15:0]	ID/ OT	[15] W17 [7] AA15	General Purpose I/O pins. Connected to an internal register, so customer can set user-defined parameters. Bits [4:0] reserved at start-up or reset for memory TDL setup. See the ZL50115/16/17/18/19/20 Programmers Model for more details. Recommend 5 kohm pulldown on these signals.
		[14] Y16 [6] AB13	
		[13] AB16 [5] AB12	
		[12] AA16 [4] AB11	
		[11] AB15 [3] AB10	
		[10] AB14 [2] AA14	
		[9] W15 [1] AA13	
		[8] Y15 [0] AB9	

**Table 12 - Administration/Control Interface Package Ball Definition**

Signal	I/O	Package Balls	Description
TEST_MODE[2:0]	I D	[2] AB17 [1] Y17 [0] AA17	Test Mode input - ensure these pins are tied to ground for normal operation. 000 SYS_NORMAL_MODE 001-010 RESERVED 011 SYS_TRISTATE_MODE 100-111 RESERVED

Table 12 - Administration/Control Interface Package Ball Definition

#### 4.6.2 JTAG Interface

All JTAG Interface signals are 5 V tolerant, and conform to the requirements of IEEE1149.1 (2001).

Signal	I/O	Package Balls	Description
JTAG_TRST	I U	Y18	JTAG Reset. Asynchronous reset. In normal operation this pin should be pulled low. Recommend external pull-down.
JTAG_TCK	I	V20	JTAG Clock - maximum frequency is 25 MHz, typically run at 10 MHz. In normal operation this pin should be pulled either high or low. Recommend external pull-down.
JTAG_TMS	I U	U20	JTAG test mode select. Synchronous to JTAG_TCK rising edge. Used by the Test Access Port controller to set certain test modes.
JTAG_TDI	I U	AA18	JTAG test data input. Synchronous to JTAG_TCK.
JTAG_TDO	O	W20	JTAG test data output. Synchronous to JTAG_TCK.

Table 13 - JTAG Interface Package Ball Definition

#### 4.7 Miscellaneous Inputs

The following unused inputs must be tied low or high as appropriate.

Signal	Package Balls	Description
IC_GND	W11, Y11, Y19, AA19, AB18	Internally connected. Tie to GND.
IC_VDD_IO	L22	Internally connected. Tie to VDD_IO.

Table 14 - Miscellaneous Inputs Package Ball Definitions



#### 4.8 Power and Ground Connections

Signal	Package Balls				Description
VDD_IO	A1 AA4 B2 C20 D4 K4 T19 W16 W7	A22 AA5 B21 C3 D7 M4 T20 W19 W8	AA2 AB1 C14 D16 G19 N19 T4 W4 Y20	AA21 AB22 C15 D19 G4 P3 U2 W5 Y3	3.3 V VDD Power Supply for IO Ring
GND	A13 AA7 B3 J10 J14 K12 K9 L12 L9 M13 N10 N14 P12 P9 Y13	A15 AB5 C2 J11 J9 K13 L1 L13 M10 M14 N11 N9 P13 R19 Y14	AA20 B11 C21 J12 K10 K14 L10 L14 M11 M19 N12 P10 P14 U3 Y2	AA3 B20 H2 J13 K11 K19 L11 L20 M12 M9 N13 P11 P2 W12 Y21	0 V Ground Supply
VDD_CORE	D14 F19 J4 P4 W9	D15 F4 L4 U19 Y6	D18 H4 N4 W14	D9 J19 P19 W6	1.8 V VDD Power Supply for Core Region
A1VDD	AA8				1.8 V PLL Power Supply

**Table 15 - Power and Ground Package Ball Definition**

#### 4.9 Internal Connections

The following pins are connected internally, and must be left open circuit.

Signal	Package Balls				Description
IC	AA1 AB2 AB7 W1 Y4	AA10 AB3 AB8 Y1 Y5	AA6 AB4 H22 Y10 Y7	AA9 AB6 K21 Y12 Y8	Internally connected. Leave open circuit

**Table 16 - Internal Connections Package Ball Definitions**

**4.10 No Connections**

The following pins are not connected internally, and should be left open circuit.

Signal	Package Balls				Description
NC	F1 J3	H3 K1	J1 K2	J2 K3	No connection. Leave open circuit.

**Table 17 - Miscellaneous Inputs Package Ball Definitions**

**4.11 Device ID**

Signal	I/O	Package Balls		Description
DEVICE_ID[4:0]	O	[4] [3] [2] [1] [0]	A10 V19 W18 F3 G2	Device ID. ZL50115 = 00000 ZL50116 = 00001 ZL50117 = 00010 ZL50118 = 00011 ZL50119 = 00100 ZL50120 = 00101

**Table 18 - Device ID Ball Definition**

## 5.0 Typical Applications

### 5.1 Leased Line Provision

Circuit emulation is typically used to support the provision of leased line services to customers using legacy TDM equipment. For example, Figure 8 shows a leased line TDM service being carried across a packet network. The advantages are that a carrier can upgrade to a packet switched network, whilst still maintaining their existing TDM business.

The ZL5011x is capable of handling circuit emulation of both structured T1, E1, and J2 links (e.g., for support of fractional circuits) and unstructured (or clear channel) T1, E1, J2, T3 and E3 links. The device handles the data-plane requirements of the provider edge inter-working function (with the exception of the physical interfaces and line interface units). Control plane functions are forwarded to the host processor controlling the ZL5011x device.

The ZL5011x provides a per-stream clock recovery function, in unstructured mode, to reproduce the TDM service frequency at the egress of the packet network. This is required otherwise the queue at the egress of the packet network will either fill up or empty, depending on whether the regenerated clock is slower or faster than the original.

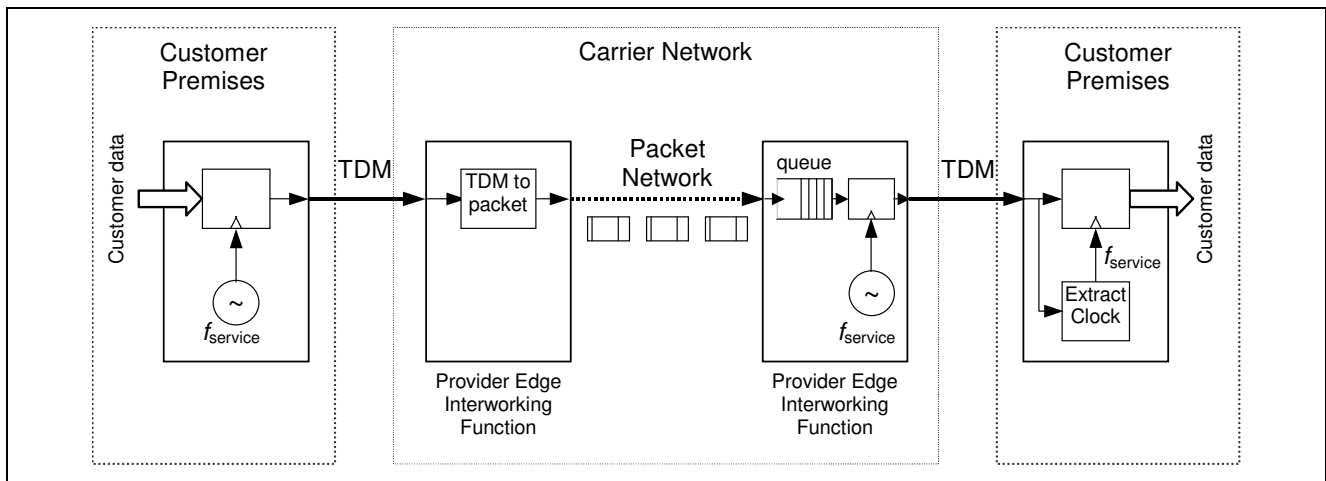


Figure 8 - Leased Line Services Over a Circuit Emulation Link

### 5.2 Remote Concentrator Unit

The remote concentrator application, shown in Figure 9, consists of a remote concentrators connected to the Central Office (CO) by a dedicated fiber link running Gigabit Ethernet (GE) or Ethernet over SONET (EoS) rather than by NxT1/E1 or DS3/E3. The remote concentrators provide both TDM service and native Ethernet service to the Multi-Tenant Unit or Multi-Dwelling Unit (MTU/MDU).

The ZL5011x is used to emulate TDM circuits over Ethernet by establishing CESoP connections between the remote concentrator and the CO. The native IP or Ethernet traffic is multiplexed with the CESoP traffic inside the remote concentrator and sent across the same GE connection to the CO. At the CO the native IP or Ethernet traffic is split from the CESoP connections at sent towards the packet network. Multiple T1/E1 CESoP connections from several remote concentrators are aggregated in the CO using a larger ZL5011x variant, converted back to TDM circuits, and connected to the PSTN through a higher bandwidth TDM circuit such as OC-3 or STM-1.

The use of CESoP here allows the convergence of voice and data on a single access network based on Ethernet. This convergence on Ethernet, a packet technology, rather than SONET/SDH, a switched circuit technology, provides cost and operational savings.

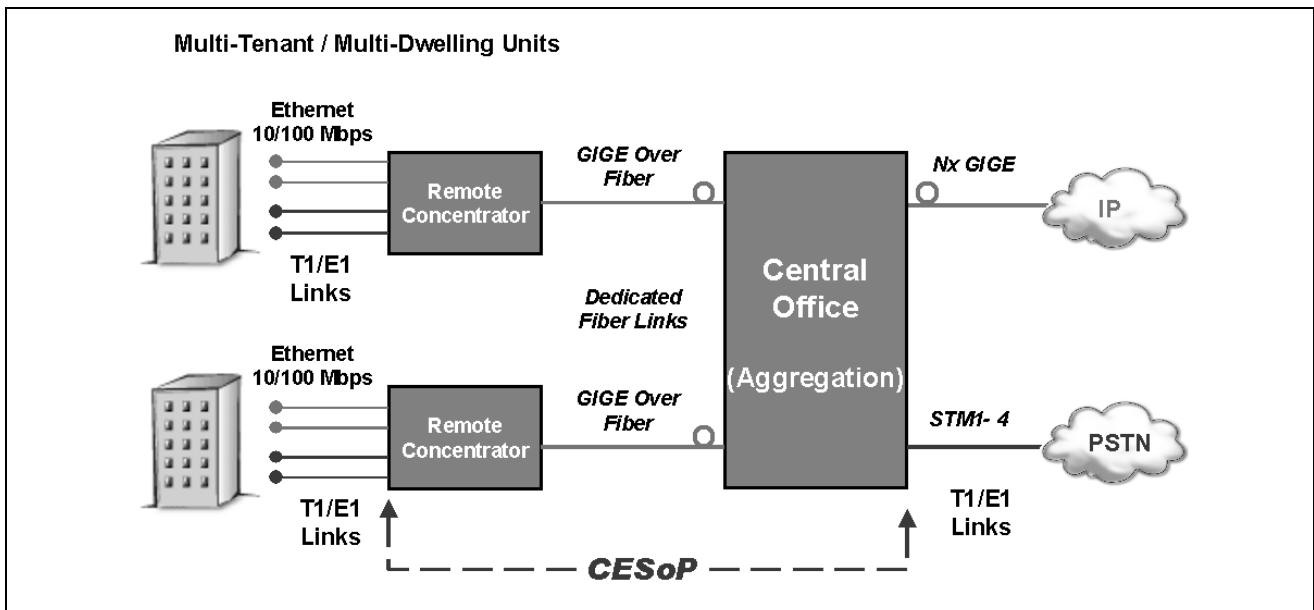


Figure 9 - Remote Concentrator Unit using CESoP

**5.3 FTTP**

The Fiber to the Premise (FTTP) application, shown in Figure 10, consists of an Ethernet Passive Optical Network (EPON) deployed in the Wide Area Network (WAN). The Optical Network Units (ONU) sit at the curb while the Optical Line Terminals (OLT) are located at the Central Office (CO). The ONUs are traditionally equipped with Ethernet interfaces to provide video and data service to the customer premise.

The ONU includes a ZL5011x which enables the box to provide T1/E1 service to the customer. The ZL5011x is used to establish CESoP connections between the ONU and the OLT to transparently carry TDM circuits across the EPON. The ONU would use a smaller variant of the ZL5011x and the OLT would use a larger variant to aggregate CESoP traffic from many ONUs and connect them at the CO to the PSTN. The native IP or Ethernet traffic from the ONU would be split off at the OLT and connected to the packet network.

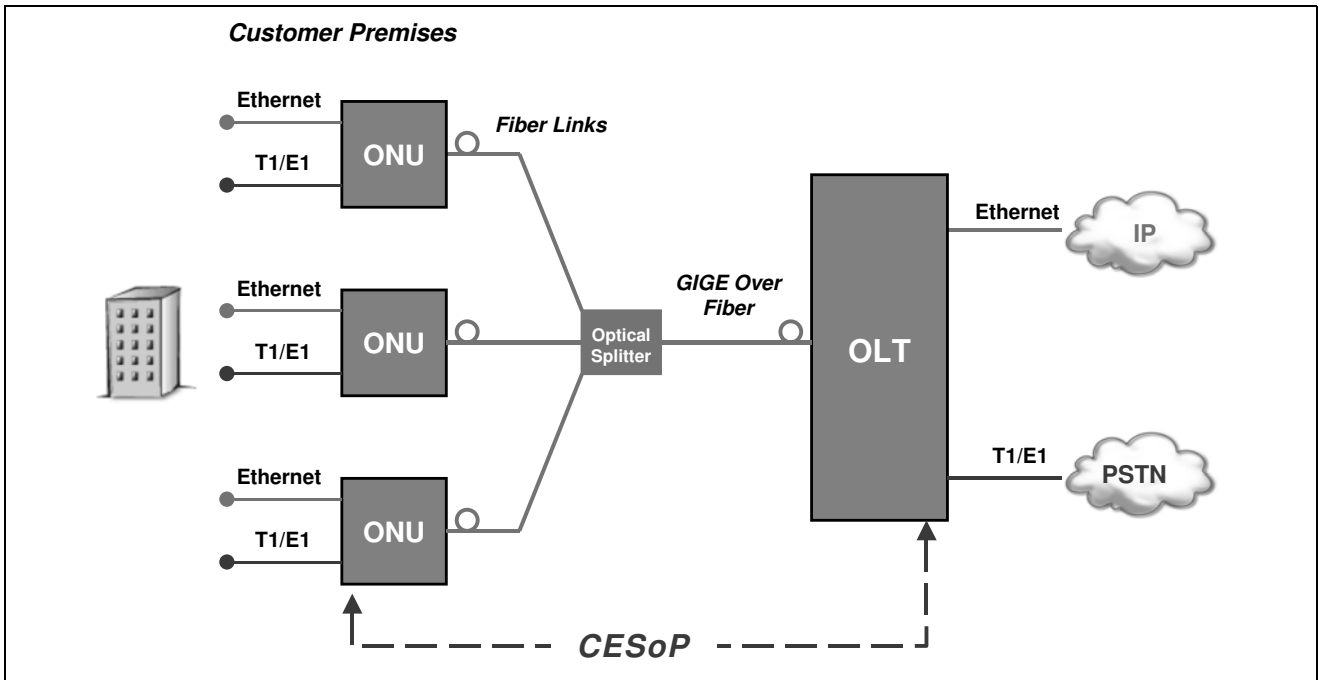


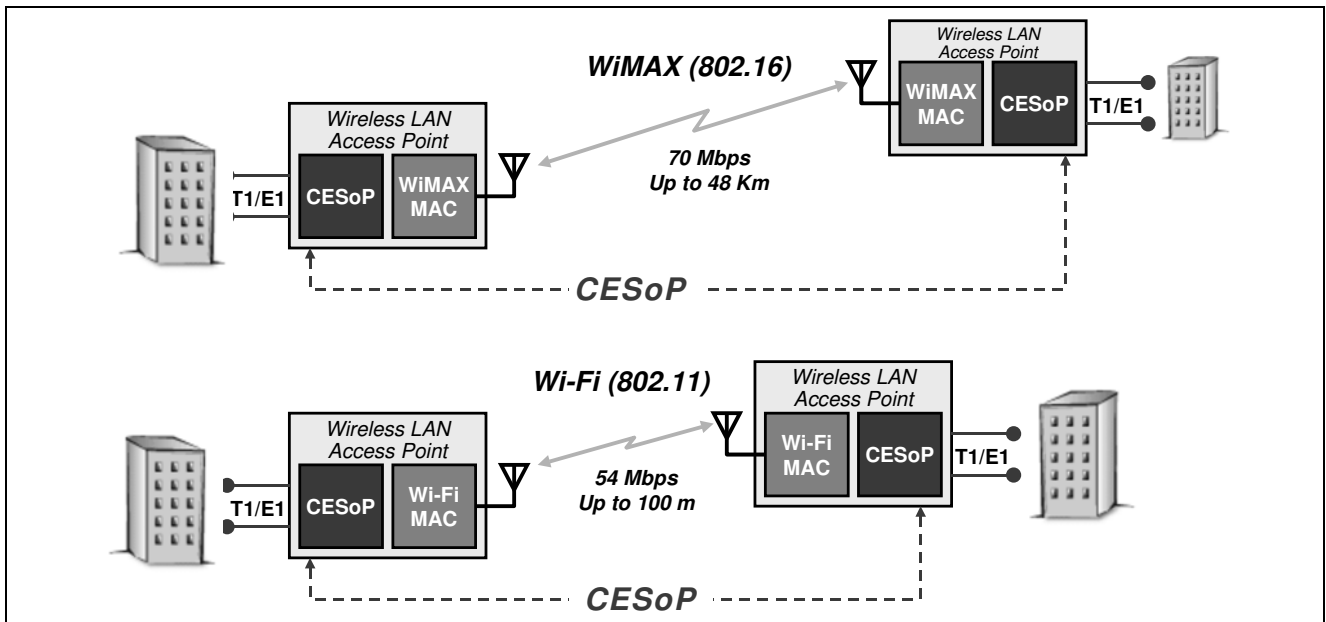
Figure 10 - EPON using CESoP

**5.4 Wireless - WiFi or WiMAX**

The wireless application, shown in Figure 11, may either be in the form of WiMAX for broadband access or Wi-Fi for smaller-scale Loans. Both technologies carry Ethernet over radio links between sites or pieces of equipment.

An application for CESoP technology over a WiMAX network is to enable the service provider to sell T1/E1 service in addition to video and data services that are natively carried across the WiMAX connection. A ZL5011x is used at the customer premise to packetize the T1/E1, fractional T1/E1 or TDM circuit into Ethernet packets, which are transported back to the Central Office (CO). At the CO the TDM circuit is re-assembled from the Ethernet packets and send to the PSTN. The CESoP traffic is converged onto the same WiMAX connection as the native Ethernet traffic for video and data.

An application for CESoP technology over a Wi-Fi network is to enable a distributed PBX system in either a single building or between buildings in a campus environment. In this application the T1/E1 connection from a PBX is connected using a CESoP to another PBX. A wireless site-to-site CESoP connection between buildings in a campus would allow for deployment savings against having to run dedicated copper cables between buildings.



**Figure 11 - Wi-Fi and WiMAX using CESoP**

### 5.5 Digital Loop Carrier

The Broadband Digital Loop Carrier (BBDLC) application, shown in Figure 12, consists of a BBDLC connected to the Central Office (CO) by a dedicated fiber link running Gigabit Ethernet (GE) rather than by NxT1/E1 or DS3/E3.

The ZL5011x is used to emulate TDM circuits over Ethernet by establishing CESoP connections between the BBDLC and the CO. At the CO the native IP or Ethernet traffic is split from the CESoP connections at sent towards the packet network. Multiple T1/E1 CESoP connections from several BBDLC are aggregated in the CO using a larger ZL5011x variant, converted back to TDM circuits, and connected to a class 5 switch destined towards the PSTN.

In this configuration T3/E3 services can also be provided. Using CESoP allows voice and data traffic to be converged onto a single link.

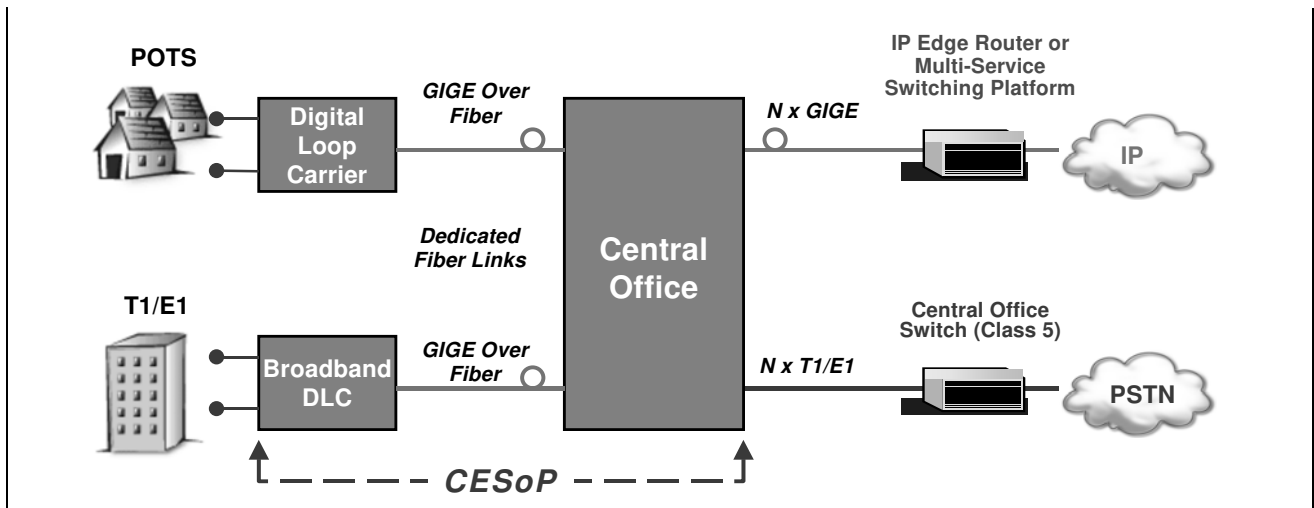


Figure 12 - Digital Loop Carrier using CESoP

### 5.6 Integrated Access Device

The Integrated Access Device (IAD) application consists of an IAD located at the curb or customer premise with an Ethernet connection to an TDM aggregation box sitting in the access area of the network.

The ZL5011x in the IAD modem packetizes the T1/E1 or fractional T1/E1 TDM circuit into Ethernet CESoP packets. The CESoP traffic is multiplexed with the native Ethernet data traffic from the IAD's Ethernet ports onto the Ethernet link to the aggregation equipment. The aggregator will split off the native Ethernet traffic from multiple IADs and send the traffic on to packet network. The aggregator will contain a larger ZL5011x that will terminate multiple CESoP connections from multiple IADs and send the TDM circuits to the PSTN, perhaps over a higher bandwidth TDM pipe such as DS3.

The use of CESoP in this application allows the IAD to support both native Ethernet service as well as T1/E1 service in the same box, while converging both types of traffic onto a single Ethernet connection back towards the provider.

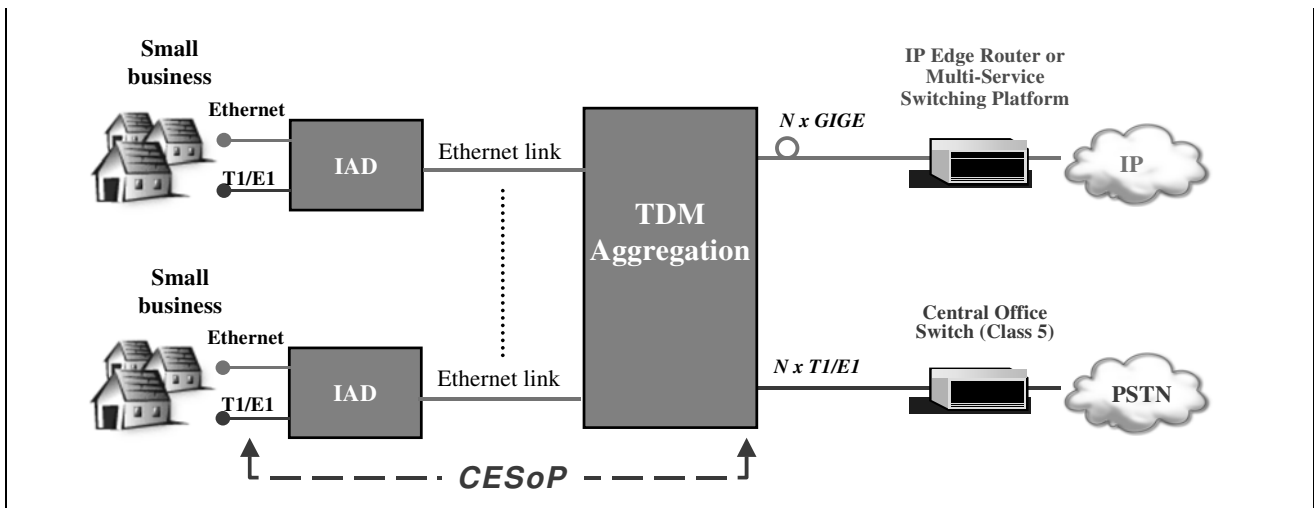


Figure 13 - Integrated Access Device Using CESoP

### 6.0 Functional Description

The ZL5011x family provides the data-plane processing to enable constant bit rate TDM services to be carried over a packet switched network, such as an Ethernet, IP or MPLS network. The device segments the TDM data into user-defined packets, and passes it transparently over the packet network to be reconstructed at the far end. This has a number of applications, including emulation of TDM circuits and packet backplanes for TDM-based equipment.

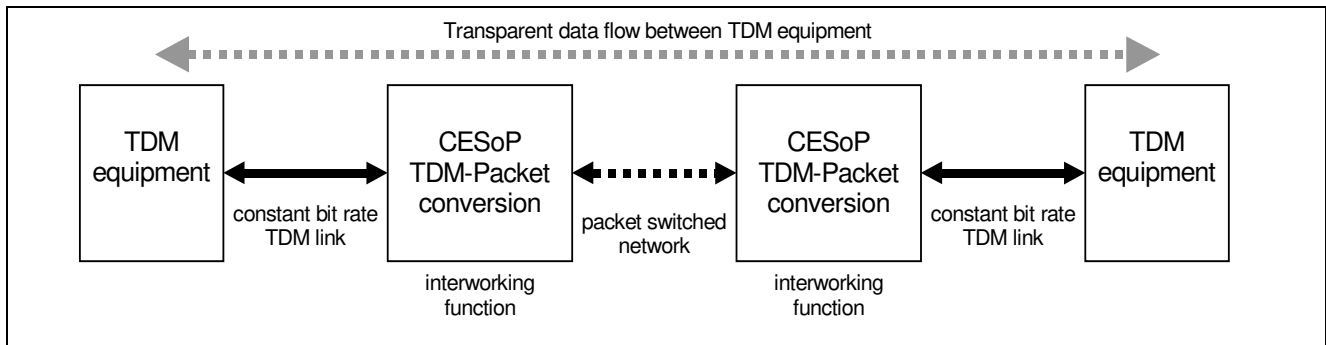


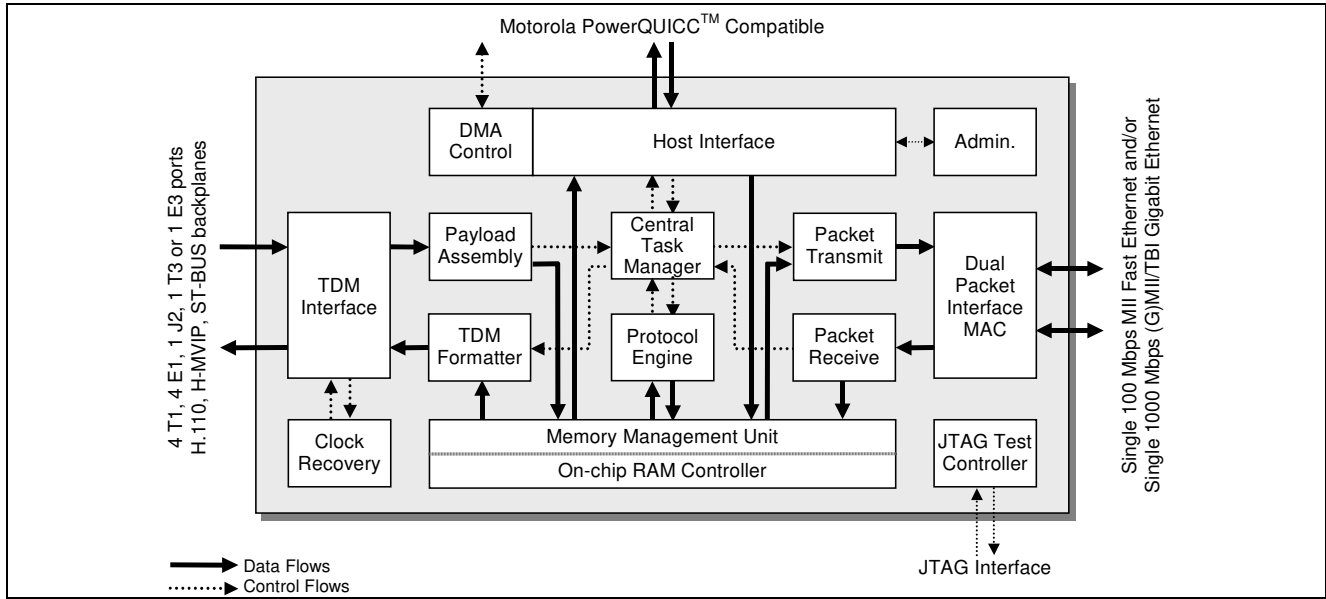
Figure 14 - ZL50115/16/17/18/19/20 Family Operation



Note: The ZL5011x does not support the transmission or reception of jumbo packets, or packet sizes larger than 1522 bytes.

**6.1 Block Diagram**

A diagram of the ZL5011x device is given in Figure 15, which shows the major data flows between functional components.



**Figure 15 - ZL50115/16/17/18/19/20 Data and Control Flows**

**6.2 Data and Control Flows**

There are numerous combinations that can be implemented to pass data through the ZL5011x device depending on the application requirements. The Task Manager can be considered the central pivot, through which all flows must operate. The Task Manager acts as a “router” in the centre of the chip, directing packets to the appropriate blocks for further processing. The task message contains a pointer to the relevant data, instructions as to what to do with the data, and ancillary information about the packet. Effectively this means the flow of data through the device can be programmed, by setting the task message contents appropriately.

Flow Number	Flow Through Device
1	TDM to (TM) to PE to (TM) to PKT
2	PKT to (TM) to PE to (TM) to TDM
3	TDM to (TM) to PKT
4	PKT to (TM) to TDM
5	TDM to (TM) to CPU
6	TDM to (TM) to PE to (TM) to CPU
7	CPU to (TM) to TDM
8	PKT to (TM) to CPU
9	CPU to (TM) to PKT

**Table 19 - Standard Device Flows**

Flow Number	Flow Through Device
10 <sup>1</sup>	TDM to (TM) to TDM
11 <sup>1</sup>	PKT to (TM) to PKT

**Table 19 - Standard Device Flows (continued)**

1. This flow is for loopback and may be helpful for test purposes

Each of the 11 data flows uses the Task Manager to route packet information to the next block or interface for onward transmission. The flow is determined by the Type field in the Task Message (see ZL50115/16/17/18/19/20 Programmers Model).

### 6.3 SYSTEM\_CLK Considerations

The quality of the 100 MHz SYSTEM\_CLK or the oscillator that drives SYSTEM\_CLK directly impacts the adaptive clock recovery performance. Zarlink has a recommended oscillator and guidelines for the selection of an oscillator. Please refer to ZL5011x Design Manual section “3.6 System Clock Block” before choosing an oscillator.

### 6.4 TDM Interface

The ZL5011x family offers the following types of TDM service across the packet network:

Service type	TDM interface	Interface type	Interfaces to
Unstructured asynchronous	T1, E1, J2, E3 and T3	Bit clock in and out Data in and out	Line interface unit
Structured synchronous (N x 64 Kbps)	T1, E1 and J2 Framed TDM data streams at 2.048 and 8.192 Mbps	Bit clock out Frame pulse out Data in and out	Framers TDM backplane (master)
		Bit clock in Frame in Data in and out	Framers TDM backplane (slave)

**Table 20 - TDM Services Offered by the ZL50115/16/17/18/19/20 Family**

Unstructured services are fully asynchronous, and include full support for clock recovery on a per stream basis. Both adaptive and differential clock recovery mechanisms can be used.

Structured services are synchronous, with all streams driven by a common clock and frame reference. These services can be offered in two ways:

- **Synchronous master mode** - the ZL5011x provides a common clock and frame pulse to all streams, which may be locked to an incoming clock or frame reference
- **Synchronous slave mode** - the ZL5011x accepts a common external clock and frame pulse to be used by all streams

In either structured mode, N x 64 Kbps trunking is supported as detailed in “Payload Order” on page 54.

The ZL5011x supports structured mode or unstructured mode, however it does not support structured mode and unstructured mode at the same time, all ports are either structured or unstructured. In structured mode, all TDM inputs must be synchronous.

### 6.4.1 TDM Interface Block

The TDM Interface contains two basic types of interface: unstructured clock and data, for interfacing directly to a line interface unit; or structured, framed data, for interfacing to a framer or TDM backplane.

Unstructured data is treated asynchronously, with every stream using its own clock. Clock recovery is provided on each output stream, to reproduce the TDM service frequency at the egress of the packet network. Structured data is treated synchronously, i.e., all data streams are timed by the same clock and frame references. These can either be supplied from an external source (slave mode) or generated internally using the on-chip stratum 4/4E PLL (master mode).

### 6.4.2 Structured TDM Port Data Formats

The ZL5011x is programmable such that the frame/clock polarity and clock alignment can be set to any desired combination. Table 21 shows a brief summary of four different TDM formats; ST-BUS, H.110, H-MVIP, and Generic (synchronous mode only), for more information see the relevant specifications shown. There are many additional formats for TDM transmission not depicted in Table 21, but the flexibility of the port will cover almost any scenario. The overall data format is set for the entire TDM Interface device, rather than on a per stream basis. It is possible to control the polarity of the master clock and frame pulse outputs, independent of the chosen data format (used when operating in synchronous master mode).

Data Format	Data Rate (Mbps)	Number of channels per frame	Clock Freq. (MHz)	Nominal Frame Pulse Width (ns)	Frame Pulse Polarity	Frame Boundary Alignment		Standard
						clock	frame pulse	
ST-bus	2.048	32	2.048	244	Negative	Rising Edge	Straddles boundary	MSAN-126 Rev B (Issue 4) Zarlink
	2.048	32	4.096	244	Negative	Falling Edge	Straddles boundary	
	8.192	128	16.384	61	Negative	Falling Edge	Straddles boundary	
H.110	8.192	128	8.192	122	Negative	Rising edge	Straddles boundary	ECTF H.110
H-MVIP	2.048	32	2.048	244	Negative	Rising Edge	Straddles boundary	H-MVIP Release 1.1a
	2.048	32	4.096	244	Negative	Falling Edge	Straddles boundary	
	8.192	128	16.384	244	Negative	Falling Edge	Straddles boundary	
Generic	2.048	32	2.048	488	Positive	Rising Edge	Rising edge of clock	
	8.192	128	8.192	122	Positive	Rising Edge	Rising edge of clock	

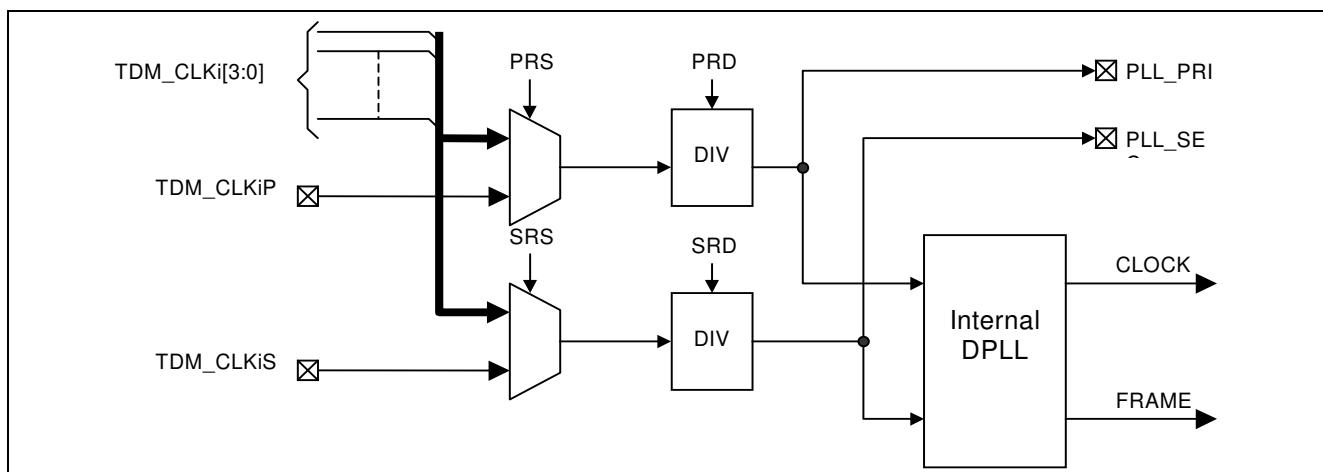
**Table 21 - Some of the TDM Port Formats Accepted by the ZL50115/16/17/18/19/20 Family**

### 6.4.3 TDM Clock Structure

The TDM interface can operate in two modes, synchronous for structured TDM data, and asynchronous for unstructured TDM data. The ZL5011x is capable of providing the TDM clock for either of the modes. The ZL5011x supports clock recovery in both synchronous and asynchronous modes of operation. In asynchronous operation each stream may have independent clock recovery.

#### 6.4.3.1 Synchronous TDM Clock Generation

In synchronous mode all 4 streams will be driven by a common clock source. When the ZL5011x is acting as a master device, the source can either be the internal DPLL or an external PLL. In both cases, the primary and secondary reference clocks are taken from either two TDM input clocks, or two external clock sources driven to the chip. The input clocks are then divided down where necessary and sent either to the internal DPLL or to the output pins for connection to an external DPLL. The DPLL then provides the common clock and frame pulse required to drive the TDM streams. See “DPLL Specification” on page 62 for further details.



**Figure 16 - Synchronous TDM Clock Generation**

When the ZL5011x is acting as a slave device, the common clock and frame pulse signals are taken from an external device providing the TDM master function.

#### 6.4.3.2 Asynchronous TDM Clock Generation

Each stream uses a separate internal DCO to provide an asynchronous TDM clock output. The DCO can be controlled to recover the clock from the original TDM source depending on the timing algorithm used.

## 6.5 Payload Assembly

Data traffic received on the TDM Interface is sampled in the TDM Interface block, and synchronized to the internal clock. It is then forwarded to the payload assembly process. The ZL5011x Payload Assembler can handle up to 128 active packet streams or “contexts” simultaneously. Each context generates a single stream of packets identified by a label in the packet header known as the “context ID”. Packet payloads are assembled in the format shown in Figure 17 - on page 53 in structured operation. This meets the requirements of the IETF CESoPSN standard (RFC 5086). Alternatively, packet payloads are assembled in the format shown in Figure 19 - on page 55. This format meets the requirements of the IETF SAToP standard (RFC 4553).

The Packet Transmit (PTX) circuit adds Layer 2 and Layer 3 protocol headers. The chosen protocol header combination for addition by the PTX must not exceed 64 bytes. The exception is context 127 (the 128th context), which must not exceed 56 bytes.

Contexts in the TDM to PKT direction are placed in the UPDATE state when they are opened, pending the local clock source generation. If there is no local clock source to generate packets, the context will remain in the UPDATE state and cannot be closed. ZL5011x Design Manual section “13.1 Understanding forceDelete” describes the procedure to close transmit contexts in the UPDATE state.

When the payload has been assembled it is written into the centrally managed memory, and a task message is passed to the Task Manager.

### 6.5.1 Structured Payload Operation

In structured mode a context may contain any number of 64 kbps channels. These channels need not be contiguous and they may be selected from any input stream.

Channels may be added or deleted dynamically from a context. This feature can be used to optimize bandwidth utilisation. Modifications to the context are synchronised with the start of a new packet.

The fixed header at the start of each packet is added by the Packet Transmit block. This consists of up to 64 bytes, containing the Ethernet header, any upper layer protocol headers, and the two byte context descriptor field (see section below). The header is entirely user programmable, enabling the use of any protocol.

The payload header and size must be chosen so that the overall packet size is not less than 64 bytes, the Ethernet standard minimum packet size. Where this is likely to be the case, the header or data must be padded (as shown in Figure 17 and Figure 19) to ensure the packet is large enough. This padding is added by the ZL5011x for most applications.

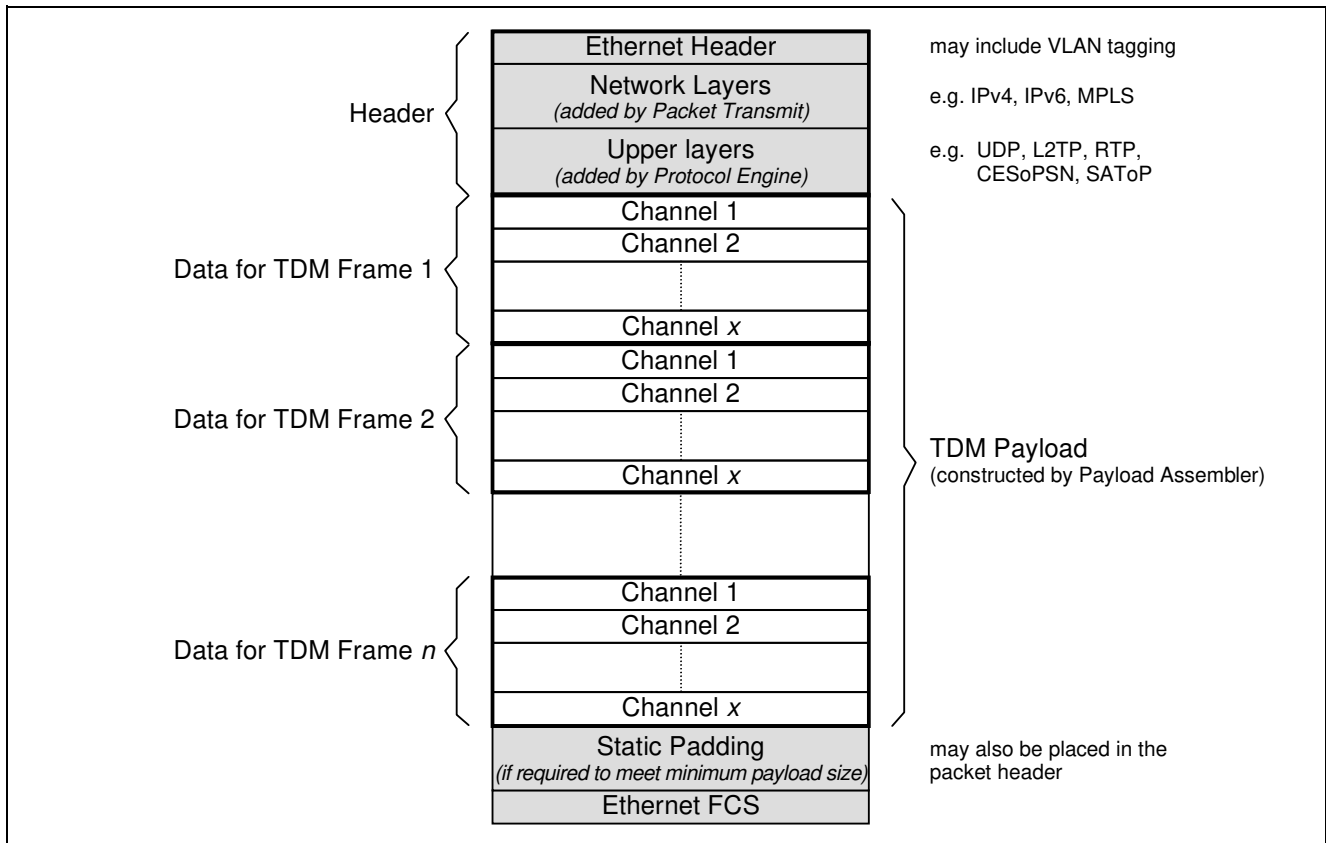
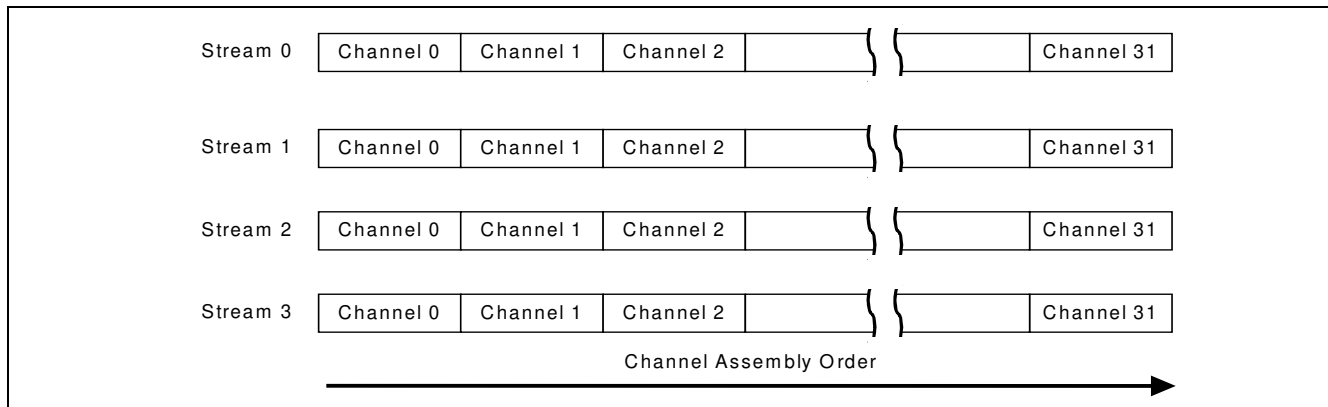


Figure 17 - ZL50115/16/17/18/19/20 Packet Format - Structured Mode

In applications where large payloads are being used, the payload size must be chosen such that the overall packet size does not exceed the maximum Ethernet packet size of 1518 bytes (1522 bytes with VLAN tags). Figure 17 shows the packet format for structured TDM data, where the payload is split into frames, and each frame concatenated to form the packet.

### 6.5.1.1 Payload Order

Packets are assembled sequentially, with each channel placed into the packet as it arrives at the TDM Interface. A fixed order of channels is maintained (see Figure 18), with channel 0 placed before channel 1, which is placed before channel 2. It is this order that allows the packet to be correctly disassembled at the far end. A context must contain only unique channel numbers. As such a context that contains the same channel from different streams, for example channel 1 from stream 2 and channel 1 from stream 3, would not be permitted.



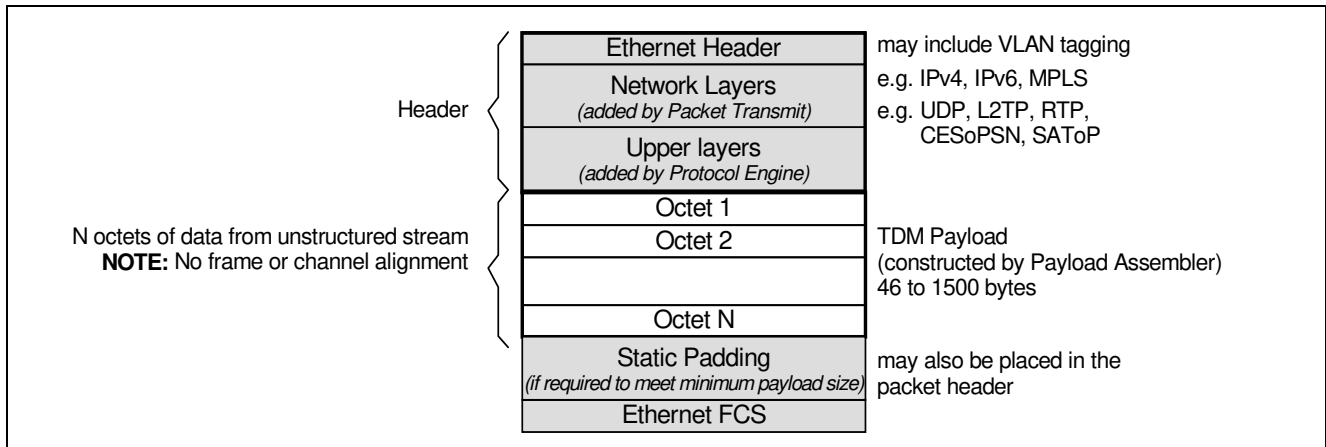
**Figure 18 - Channel Order for Packet Formation**

Each packet contains one or more frames of TDM data, in sequential order. This groups the selected channels for the first frame, followed by the same set of channels for the subsequent frame, and so on.

### 6.5.2 Unstructured Payload Operation

In unstructured mode, the payload is not split by defined frames or timeslots, so the packet consists of a continuous stream of data. Each packet consists of a number of octets, as shown in Figure 19. The number of octets in a packet need not be an integer number of frames. A typical value for N may be 192, as defined in the IETF PWE3 RFC.

For example, consider mapping the unstructured data of a 25 timeslot DS0 stream. The data for each T1 frame would normally consist of 193 bits, 192 data bits and 1 framing bit. If the payload consists of 24 octets it will be 1 bit short of a complete frames worth of data, if the payload consists of 25 octets it will be 7 bits over a complete frames worth of data. **NOTE:** No alignment of the octets with the T1 framing structure can be assumed.



**Figure 19 - ZL50115/16/17/18/19/20 Packet Format - Unstructured Mode**

Note: To change the packet size of a context, first close the context and then re-open the context with a new packet size.

### 6.6 Protocol Engine

In general, the next processing block for TDM packets is the Protocol Engine. This handles the data-plane requirements of the main higher level protocols (layers 4 and 5) expected to be used in typical applications of the ZL5011x family: UDP, RTP, L2TP, CESoPSN, SAToP and CDP. The Protocol Engine can add a header to the datagram containing up to 24 bytes. This header is largely static information, and is programmed directly by the CPU. It may contain a number of dynamic fields, including a length field, checksum, sequence number and a timestamp. The location, and in some cases the length of these fields is also programmable, allowing the various protocols to be placed at variable locations within the header.

### 6.7 Packet Transmission

Packets ready for transmission are queued to the switch fabric interface by the Queue Manager. Four classes of service are provided, allowing some packet streams to be prioritized over others. On transmission, the Packet Transmit block appends a programmable header, which has been set up in advance by the control processor. Typically this contains the data-link and network layer headers (layers 2 and 3), such as Ethernet, IP (versions 4 and 6) and MPLS.

### 6.8 Packet Reception

Incoming data traffic on the packet interface is received by the MACs. The well-formed packets are forwarded to a packet classifier to determine the destination. When a packet is successfully classified the destination can be the TDM interface, the LAN interface or the host interface. TDM traffic is then further classified to determine the context it is intended for.

Each TDM interface context has an individual queue, and the TDM re-formatting process re-creates the TDM streams from the incoming packet streams. This queue is used as a jitter buffer, to absorb variation in packet delay across the network. The size of the jitter buffer can be programmed in units of TDM frames (i.e., steps of 125 μs).

There is also a queue to the host interface, allowing a traffic flow to the host CPU for processing. The host's DMA controller can be used to retrieve packet data and write it out into the CPU's own memory.

## 6.9 TDM Formatter

At the receiving end of the packet network, the original TDM data must be re-constructed from the packets received. This is known as re-formatting, and follows the reverse process from the Payload Assembler. The TDM Formatter plays out the packets in the correct sequence, directing each octet to the selected timeslot on the output TDM interface.

When lost or late packets are detected, the TDM Formatter plays out underrun data for the same number of TDM frames as were included in the missing packet. Underrun data can either be the last value played out on that timeslot, or a pre-programmed value (e.g., 0xFF). If the packet subsequently turns up it is discarded. In this way, the end-to-end latency through the system is maintained at a constant value.

Contexts in the Packet to TDM direction are placed in the UPDATE state when they are opened, pending first packet arrival. If a packet never arrives the context will remain in the UPDATE state. ZL5011x Design Manual section “13.1 Understanding forceDelete” describes the procedure to close receive contexts in the UPDATE state.

## 6.10 Ethernet Traffic Aggregation (ZL50118/19/20 only)

The ZL5011x allows native Ethernet traffic received on the customer side Fast Ethernet port to be aggregated with the CESoP traffic from the TDM interface to the provider side Gigabit Ethernet port. Likewise, traffic from the provider side Gigabit Ethernet port may be split between CESoP traffic destined towards the TDM interface and native Ethernet traffic destined towards the customer side Fast Ethernet port. This functionality is achieved by correctly programming the task manager and packet classifiers for flow 11.

From the provider side Gigabit Ethernet port to the customer side TDM and Fast Ethernet interfaces there is sufficient internal bandwidth to avoid any prioritization issues. From the customer side TDM and Fast Ethernet interfaces towards the Gigabit Ethernet ports the TDM CESoP traffic may be sent to a higher priority output queue (there are four output queues total) than the native Fast Ethernet traffic. In this way the access to the provider side Gigabit Ethernet port is prioritized for TDM traffic over native Ethernet traffic.



## 7.0 Clock Recovery

One of the main issues with circuit emulation is that the clock used to drive the TDM link is not necessarily linked into the central office reference clock, and hence may be any value within the tolerance defined for that service. The reverse link may also be independently timed, and operating at a slightly different frequency. In the plesiochronous digital hierarchy the difference in clock frequencies between TDM links is compensated for using bit stuffing techniques, allowing the clock to be accurately regenerated at the remote end of the carrier network.

With a packet network, that connection between the ingress and egress frequency is broken, since packets are discontinuous in time. From Figure 8, the TDM service frequency  $f_{service}$  at the customer premises must be exactly reproduced at the egress of the packet network. The consequence of a long-term mismatch in frequency is that the queue at the egress of the packet network will either fill up or empty, depending on whether the regenerated clock is slower or faster than the original. This will cause loss of data and degradation of the service.

The ZL5011x provides a per-stream clock recovery function to reproduce the TDM service frequency at the egress of the packet network. There are two schemes employed, depending on the availability of a common reference clock at each provider edge unit, within the ZL5011x - differential and adaptive.

The adaptive and differential algorithms assume that there are no bit errors in the received packet header sequence number or timestamp fields. If there are bit errors in the sequence number or timestamp fields, especially in the most significant bits, then it is likely to cause a temporary degradation of the recovered clock performance. It is advised to protect packets end-to-end (e.g., by using Ethernet FCS) such that packets with bit errors are discarded and do not impact the recovered clock performance.

The clock recovery itself is performed by software in the host processor, with support from on-chip hardware to gather the required statistics.

### 7.1 Differential Clock Recovery

For applications where the wander characteristics of the recovered clock are very important, such as when the emulated circuit must be connected into the plesiochronous digital hierarchy (PDH), the ZL5011x also offers a differential clock recovery technique. This relies on having a common reference clock available at each provider edge point. Figure 20 illustrates this concept with a common Primary Reference Source (PRS) clock being present at both the source and destination equipment.

The differential algorithm assumes that the common clock is always present. There is no internal holdover capability for the common clock source (e.g., TDM\_CLKiP). If the availability of the common clock can not be guaranteed, then it is recommended to use an external DPLL with holdover capability to provide a clock source at all times. The external DPLL may enter holdover while the common clock is absent to maintain a relatively close frequency to the original common clock.

In a differential technique, the timing of the TDM service clock is sent relative to the common reference clock. Since the same reference is available at the packet egress point and the packet size is fixed, the original service clock frequency can be recovered. This technique is unaffected by any low frequency components in the packet delay variation. The disadvantage is the requirement for a common reference clock at each end of the packet network, which could either be the central office TDM clock, or provided by a global position system (GPS) receiver.

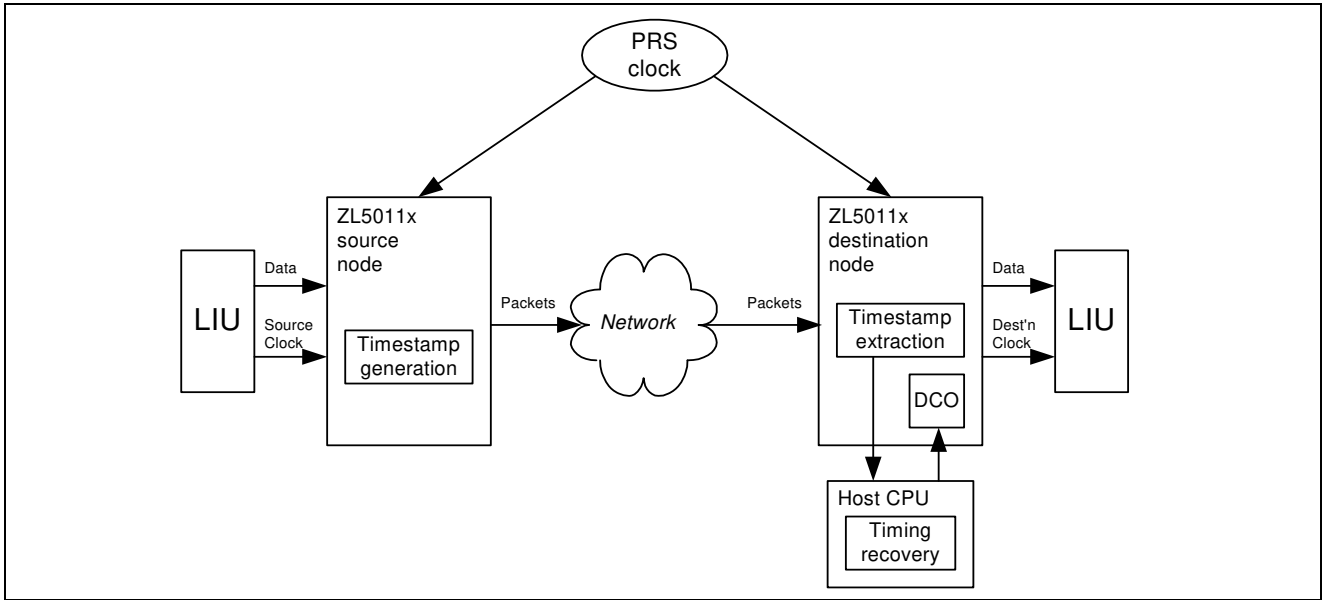


Figure 20 - Differential Clock Recovery

**7.2 Adaptive Clock Recovery**

For applications where there is no common reference clock between provider edge units, an adaptive clock recovery technique is provided. The Adaptive clock recovery solution provided in the Zarlink CESoP products is a combination hardware and software. The chip contains a DCO per TDM port in unstructured mode, that enables the recovery of up to 32 independent clocks. The timing algorithm resides in the API and runs out of the host processor. The basic information is transmitted using timestamps. Current CES standards allow for using of timestamps. Timestamps may be implied by the value of the sequence numbers, or it can be formatted as RTP timestamps. When a packet containing TDM data is sent, an RTP timestamp and/or sequence number is placed into the packet header. On arrival at the receiving device, the arrival time is noted in the form of a local timestamp, driven by the output clock of the TDM port it is destined for.

The recovered clock at the egress point of the ZL5011x is based on non-linear filtering of the timestamps that are carried in the CESoP packets. The performance of the clock recovery is greatly improved by applying these non-linear filtering techniques. The adaptive clock recovery performance is dependent on the network configuration and operation, if the loading of the network is constrained, then the wander of the recovered clock will not exceed the specified limits.

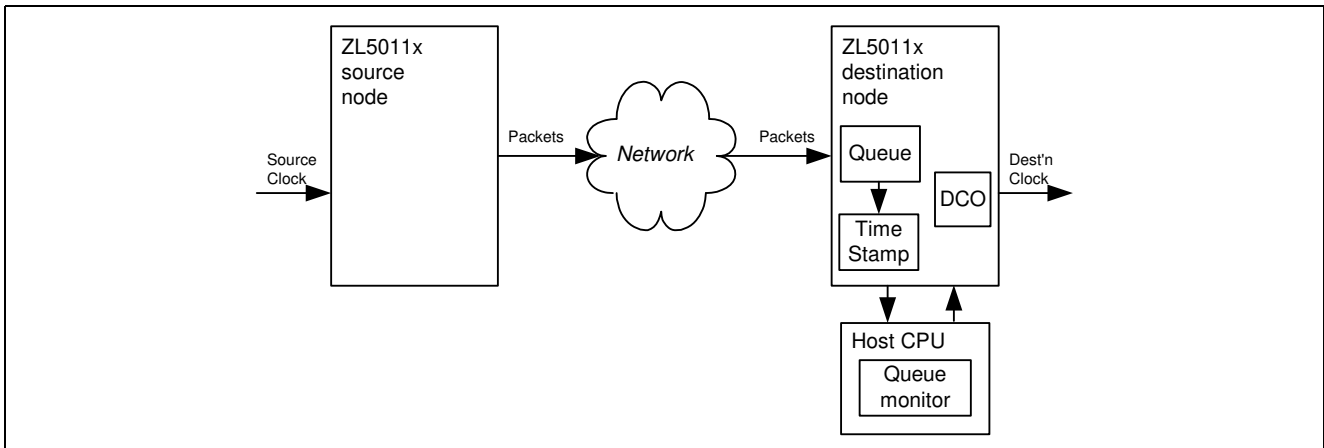


Figure 21 - Adaptive Clock Recovery

## 8.0 System Features

### 8.1 Latency

The following lists the intrinsic processing latency of the ZL5011x. The intrinsic processing latency is dependent on the number of channels in a context for structured operation, as detailed below. However, the intrinsic processing latency is not dependent on the total number of contexts opened or the total number of channels being processed by the device.

- TDM to Packet transmission processing latency less than 125  $\mu$ s
- Packet to TDM transmission processing latency less than 250  $\mu$ s (unstructured)
- Packet to TDM transmission processing latency less than 250  $\mu$ s (structured, more than 16 channels in context)
- Packet to TDM transmission processing latency less than 375  $\mu$ s (structured, 16 or less channels in context)

End-to-end latency may be estimated as the transmit latency + packet network latency + receive latency. The transmit latency is the sum of the transmit processing and the number of frames per packet x 125  $\mu$ s. The receive latency is the sum of the receive processing and the delay through the jitter buffer which is programmed to compensate for packet network PDV.

The ZL5011x is capable of creating an extremely low latency connection, with end to end delays of less than 0.5 ms, depending on user configuration.

### 8.2 Loopback Modes

The ZL5011x devices support loopback of the TDM circuits and the circuit emulation packets.

TDM loopback is achieved by first packetizing the TDM circuit as normal via the TDM Interface and Payload Assembly blocks. The packetized data is then routed by the Task Manager back to the same TDM port via the TDM Formatter and TDM Interface.

Loopback of the emulated services is achieved by redirecting classified packets from the Packet Receive blocks, back to the packet network. The Packet Transmit blocks are setup to strip the original header and add a new header directing the packets back to the source.

### 8.3 Host Packet Generation

The control processor can generate packets directly, allowing it to use the network for out-of-band communications. This can be used for transmission of control data or network setup information, e.g., routing information. The host interface can also be used by a local resource for network transmission of processed data.

The device supports dual address DMA transfers of packets to and from the CPU memory, using the host's own DMA controller. Table 22 illustrates the maximum bandwidths achievable by an external DMA master.

DMA Path	Packet Size	Max Bandwidth Mbps <sup>1</sup>
ZL5011x to CPU only	>1000 bytes	50
ZL5011x to CPU only	60 bytes	6.7
CPU to ZL5011x only	>1000 bytes	60
CPU to ZL5011x only	60 bytes	43
Combined <sup>2</sup>	>1000 bytes	58 (29 each way)
Combined <sup>2</sup>	60 bytes	11 (5.5 each way)

**Table 22 - DMA Maximum Bandwidths**

Note 1: Maximum bandwidths are the maximum the ZL5011x devices can transfer under host control, and assumes only minimal packet processing by the host.

Note 2: Combined figures assume the same amount of data is to be transferred each way.

## 8.4 Loss of Service (LOS)

During normal operation, a situation may arise where a Loss of Service occurs. This may be caused by a disruption in the transmission line due to engineering works or cable disconnection, for example. The locally detected LOS should be transferred across the emulated T1/E1 to the far end. The far end, in turn, should propagate AIS downstream.

The handling of LOS over a CESoP connection is typically performed using (setting/clearing) the L bit in the CESoPSN or SAToP control word of the packet header.

Refer to ZL5011x Design Manual section “3.1.1 Connection to LIU” for details on a variety of different ways that LOS may be handled in an application.

## 8.5 Power Up Sequence

To power up the ZL5011x the following procedure must be used:

- The I/O supply should lead the Core supply, or both can be brought up together
- The I/O supply must never exceed the Core supply by more than 2.0VDC
- The Core supply must never exceed the I/O supply by more than 0.5V<sub>DC</sub>
- Both the Core supply and the I/O supply must be brought up together
- The System Reset and, if used, the JTAG Reset must remain low until at least 100 μs after the 100 MHz system clock has stabilised. Note that if JTAG Reset is not used it must be tied low

This is illustrated in the diagram shown in Figure 22.

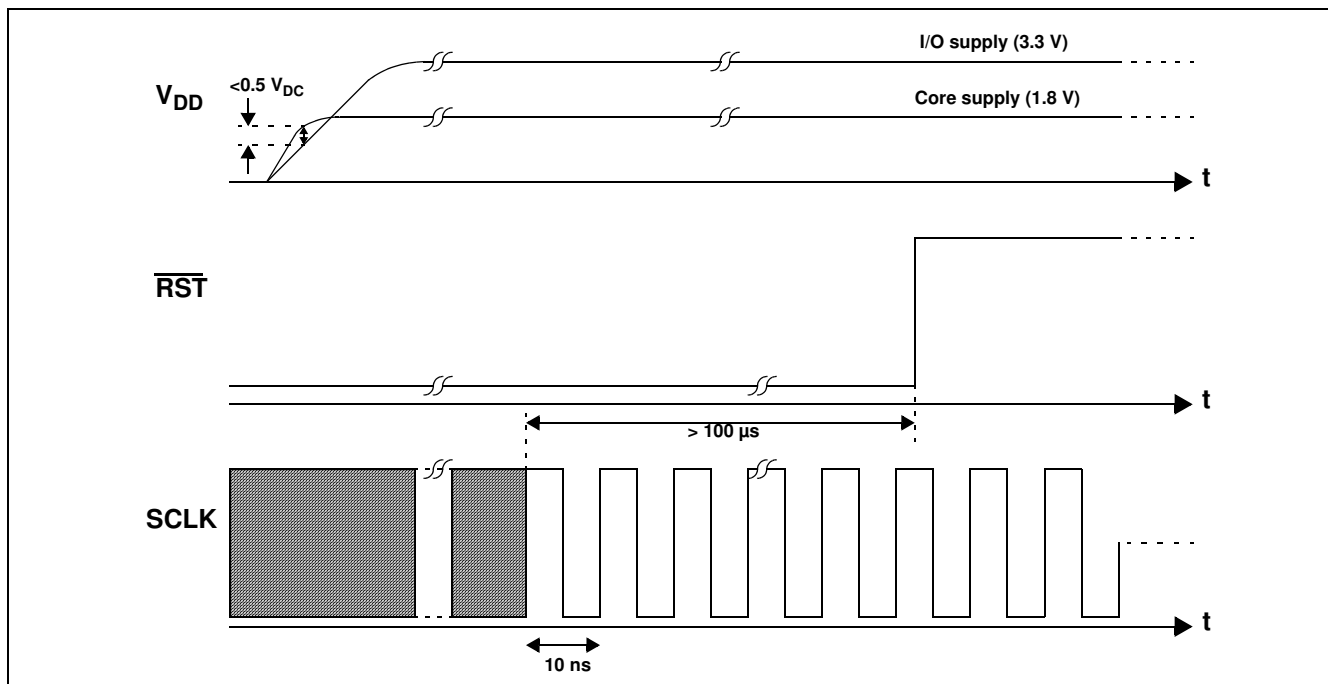


Figure 22 - Powering Up the ZL5011x

## 8.6 JTAG Interface and Board Level Test Features

The JTAG interface is used to access the boundary scan logic for board level production testing.

## 8.7 External Component Requirements

- Direct connection to PowerQUICC™ II (MPC8260) host processor and associated memory, but can support other processors with appropriate interface logic
- TDM Framers and/or Line Interface Units
- Ethernet PHY for each MAC port

## 8.8 Miscellaneous Features

- System clock speed of 100 MHz
- Host clock speed of up to 66 MHz
- Debug option to freeze all internal state machines
- JTAG (IEEE1149) Test Access Port
- Fully compatible with MT90880/1/2/3 and ZL50110/11/12/14 Zarlink products

## 8.9 Test Modes Operation

### 8.9.1 Overview

The ZL5011x family supports the following modes of operation.

#### 8.9.1.1 System Normal Mode

This mode is the device's normal operating mode. Boundary scan testing of the peripheral ring is accessible in this mode via the dedicated JTAG pins. The JTAG interface is compliant with the IEEE Std. 1149.1-2001; Test Access Port and Boundary Scan Architecture.

Each variant has its own dedicated .bsdl file which fully describes its boundary scan architecture.

#### 8.9.1.2 System Tri-State Mode

All output and I/O output drivers are tri-stated allowing the device to be isolated when testing or debugging the development board.

### 8.9.2 Test Mode Control

The System Test Mode is selected using the dedicated device input bus TEST\_MODE[2:0] as follows in Table 23.

System Test Mode	test_mode[2:0]
SYS_NORMAL_MODE	3'b000
SYS_TRI_STATE_MODE	3'b011

**Table 23 - Test Mode Control**

### 8.9.3 System Normal Mode

Selected by TEST\_MODE[2:0] = 3'b000. As the test\_mode[2:0] inputs have internal pull-downs this is the default mode of operation if no external pull-up/downs are connected. The GPIO[15:0] bus is captured on the rising edge of the external reset to provide internal bootstrap options. After the internal reset has been de-asserted the GPIO pins may be configured by the ADM module as either inputs or outputs.

### 8.9.4 System Tri-state Mode

Selected by TEST\_MODE[2:0] = 3'b011. All device output and I/O output drivers are tri-stated.

## 9.0 DPLL Specification

The ZL5011x family incorporates an internal DPLL that meets Telcordia GR-1244-CORE Stratum 4/E requirements, assuming an appropriate clock oscillator is connected to the system clock pin. It will meet the jitter/wander tolerance, jitter/wander transfer, intrinsic jitter/wander, frequency accuracy, capture range, phase change slope, holdover frequency and MTIE requirements for these specifications. In structured mode with the ZL5011x device operating as a master the DPLL is used to provide clock and frame reference signals to the internal and external TDM infrastructure. In structured mode, with the ZL5011x device operating as a slave, the DPLL is not used. All TDM clock generation is performed externally and the input streams are synchronised to the system clock by the TDM interface. The DPLL is not required in unstructured mode, hence it is not available, because TDM clocks and frame signals are generated by internal DCO's assigned to each individual stream.

### 9.1 Modes of Operation

It can be set into one of four operating modes: Locking mode, Holdover mode, Freerun mode and Powerdown mode.

#### 9.1.1 Locking Mode (normal operation)

The DPLL accepts a reference signal from either a primary or secondary source, providing redundancy in the event of a failure. These references should have the same nominal frequencies but do not need to be identical as long as their frequency offsets meet the appropriate Stratum requirements. Each source is selected from any one of the available TDM input stream clocks (up to 4 on the ZL50117/20 variants), or from the external TDM\_CLKiP (primary) or TDM\_CLKiS (secondary) input pins, as illustrated in Figure 16 - on page 52. It is possible to supply a range of input frequencies as the DPLL reference source, depicted in Table 24. The PRD register Value is the number (in hexadecimal) that must be programmed into the PRD register within the DPLL to obtain the divided down frequency at PLL\_PRI or PLL\_SEC.

Source Input Frequency (MHz)	Tolerance ( $\pm$ ppm)	Divider Ratio	PRD/SRD Register Value (Hex) (Note 1)	Frequency at PLL_PRI or PLL_SEC (MHz)	Maximum Acceptable Input Wander tolerance (UI) (Note 2)
0.008	30	1	1	0.008	$\pm$ 1
1.544	130	1	1	1.544	$\pm$ 1023
2.048	50	1	1	2.048	$\pm$ 1023
4.096	50	1	1	4.096	$\pm$ 1023
8.192	50	1	1	8.192	$\pm$ 1023

Table 24 - DPLL Input Reference Frequencies

Source Input Frequency (MHz)	Tolerance ( $\pm$ ppm)	Divider Ratio	PRD/SRD Register Value (Hex) (Note 1)	Frequency at PLL_PRI or PLL_SEC (MHz)	Maximum Acceptable Input Wander tolerance (UI) (Note 2)
16.384	50	1	1	16.384	$\pm$ 1023
6.312	30	1	1	6.312	$\pm$ 1023
22.368	20	2796	AEC	0.008	$\pm$ 1 (on 64k Hz)
34.368	20	537	219	0.064	$\pm$ 1 (on 64 kHz)
44.736 (Note 3)	20	699	2BB	0.064	$\pm$ 1 (on 64 kHz)

**Table 24 - DPLL Input Reference Frequencies**

Note 1: A PRD/SRD value of 0 will suppress the clock, and prevent it from reaching the DPLL.

Note 2: UI means Unit Interval - in this case periods of the time signal. So  $\pm$ 1UI on a 64 kHz signal means  $\pm$ 15.625  $\mu$ s, the period of the reference frequency. Similarly  $\pm$ 1023UI on a 4.096 MHz signal means  $\pm$ 250  $\mu$ s.

Note 3: This input frequency is supported with the use of an external divide by 2.

The maximum lock-in range can be programmed up to  $\pm$ 372 ppm regardless of the input frequency. The DPLL will fail to lock if the source input frequency is absent, if it is not of approximately the correct frequency or if it is too jittery. See Section 9.7 for further details. The Application Program Interface (API) software that accompanies the ZL5011x family can be used to automatically set up the DPLL for the appropriate standard requirement.

The DPLL lock-in range can be programmed using the Lock Range register (see ZL50115/16/17/18/19/20 Programmers Model document) in order to extend or reduce the capture envelope. The DPLL provides bit-error-free reference switching, meeting the specification limits in the Telcordia GR-1244-CORE standard. If Stratum 4/4E accuracy is not required, it is possible to use a more relaxed system clock tolerance.

The DPLL output consists of three signals; a common clock (comclk), a double-rate common clock (comclkx2) and a frame reference (8 kHz). These are used to time the internal TDM Interface, and hence the corresponding TDM infrastructure attached to the interface. The output clock options are either 2.048 Mbps (comclkx2 at 4.096 Mbps) or 8.192 Mbps (comclkx2 at 16.384 Mbps), determined by setup in the DPLL control register. The frame pulse is programmable for polarity and width.

### 9.1.2 Holdover Mode

In the event of a reference failure resulting in an absence of both the primary and secondary source, the DPLL automatically reverts to Holdover mode. The last valid frequency value recorded before failure can be maintained within the Stratum 3 limits of  $\pm$ 0.05 ppm. The hold value is wholly dependent on the drift and temperature performance of the system clock. For example, a  $\pm$ 32 ppm oscillator may have a temperature coefficient of  $\pm$ 0.1 ppm/ $^{\circ}$ C. Thus a 10 $^{\circ}$ C ambient change since the DPLL was last in the Locking mode will change the holdover frequency by an additional  $\pm$ 1 ppm, which is much greater than the  $\pm$ 0.05 ppm Stratum 3 specification. If the strict target of Stratum 3 holdover accuracy is not required, a less restrictive oscillator can be used for the system clock.

Holdover mode is typically used for a short period of time until network synchronisation is re-established.

### 9.1.3 Freerun Mode

In freerun mode the DPLL is programmed with a centre frequency, and can output that frequency within the Stratum 3 limits of  $\pm$ 4.6 ppm. To achieve this the 100 MHz system clock must have an absolute frequency accuracy of  $\pm$ 4.6 ppm. The centre frequency is programmed as a fraction of the system clock frequency.

### 9.1.4 Powerdown Mode

It is possible to “power down” the DPLL when it is not in use. For example, an unstructured TDM system, or use of an external DPLL would mean the internal DPLL could be switched off, saving power. The internal registers can still be accessed while the DPLL is powered down.

## 9.2 Reference Monitor Circuit

There are two identical reference monitor circuits, one for the primary and one for the secondary source. Each circuit will continually monitor its reference, and report the references validity. The validity criteria depends on the frequency programmed for the reference. A reference must meet all the following criteria to maintain validity:

- The “period in specified range” check is performed regardless of the programmed frequency. Each period must be within a range, which is programmable for the application. Refer to the ZL50115/16/17/18/19/20 Programmers Model for details.
- If the programmed frequency is 1.544 MHz or 2.048 MHz, the “n periods in specified range” check will be performed. The time taken for n cycles must be within a programmed range, typically with n at 64, the time taken for consecutive cycles must be between 62 and 66 periods of the programmed frequency.

The fail flags are independent of the preferred option for primary or secondary operation, will be asserted in the event of an invalid signal regardless of mode.

## 9.3 Locking Mode Reference Switching

When the reference source the DPLL is currently locking to becomes invalid, the DPLL’s response depends on which one of the failure detect modes has been chosen: autodetect, forced primary, or forced secondary. One of these failure detect modes must be chosen via the FDM1:0 bits of the DOM register. After a device reset via the SYSTEM\_RESET pin, the autodetect mode is selected.

In autodetect mode (automatic reference switching) if both references are valid the DPLL will synchronise to the preferred reference. If the preferred reference becomes unreliable, the DPLL continues driving its output clock in a stable holdover state until it makes a switch to the backup reference. If the preferred reference recovers, the DPLL makes a switch back to the preferred reference. If necessary, the switch back can be prevented by changing the preferred reference using the REFSEL bit in the DOM register, after the switch to the backup reference has occurred.

If both references are unreliable, the DPLL will drive its output clock using the stable holdover values until one of the references becomes valid.

In forced primary mode, the DPLL will synchronise to the primary reference only. The DPLL will not switch to the secondary reference under any circumstances including the loss of the primary reference. In this condition, the DPLL remains in holdover mode until the primary reference recovers. Similarly in forced secondary mode, the DPLL will synchronise to the secondary reference only, and will not switch to the primary reference. Again, a failure of the secondary reference will cause the DPLL to enter holdover mode, until such time as the secondary reference recovers. The choice of preferred reference has no effect in these modes.

When a conventional PLL is locked to its reference, there is no phase difference between the input reference and the PLL output. For the DPLL, the input references can have any phase relationship between them. During a reference switch, if the DPLL output follows the phase of the new reference, a large phase jump could occur. The phase jump would be transferred to the TDM outputs. The DPLL’s MTIE (Maximum Time Interval Error) feature preserves the continuity of the DPLL output so that it appears no reference switch had occurred. The MTIE circuit is not perfect however, and a small Time Interval Error is still incurred per reference switch. To align the DPLL output clock to the nearest edge of the selected input reference, the MTIE reset bit (MRST bit in the DOM register) can be used.

Unlike some designs, switching between references which are at different nominal frequencies do not require intervention such as a system reset.



## 9.4 Locking Range

The locking range is the input frequency range over which the DPLL must be able to pull into synchronization and to maintain the synchronization. The locking range is programmable up to  $\pm 372$  ppm.

Note that the locking range relates to the system clock frequency. If the external oscillator has a tolerance of -100 ppm, and the locking range is programmed to  $\pm 200$  ppm, the actual locking range is the programmed value shifted by the system clock tolerance to become -300 ppm to +100 ppm.

## 9.5 Locking Time

The Locking Time is the time it takes the synchroniser to phase lock to the input signal. Phase lock occurs when the input and output signals are not changing in phase with respect to each other (not including jitter).

Locking time is very difficult to determine because it is affected by many factors including:

- initial input to output phase difference
- initial input to output frequency difference
- DPLL Loop Filter
- DPLL Limiter (phase slope)

Although a short phase lock time is desirable, it is not always achievable due to other synchroniser requirements. For instance, better jitter transfer performance is obtained with a lower frequency loop filter which increases locking time; and a better (smaller) phase slope performance will increase locking time. Additionally, the locking time is dependent on the `p_shift` value.

The DPLL Loop Filter and Limiter have been optimised to meet the Telcordia GR-1244-CORE jitter transfer and phase alignment speed requirements. The phase lock time is guaranteed to be no greater than 30 seconds when using the recommended Stratum 3 and Stratum 4/4E register settings.

## 9.6 Lock Status

The DPLL has a Lock Status Indicator and a corresponding Lock Change Interrupt. The response of the Lock Status Indicator is a function of the programmed Lock Detect Interval (LDI) and Lock Detect Threshold (LDT) values in the `dpll_idetect` register. The LDT register can be programmed to set the jitter tolerance level of the Lock Status Indicator. To determine if the DPLL has achieved lock the Lock Status Indicator must be high for a period of at least 30 seconds. When the DPLL loses lock the Lock Status Indicator will go low after  $LDI \times 125 \mu s$ .

## 9.7 Jitter

The DPLL is designed to withstand, and improve inherent jitter in the TDM clock domain.

### 9.7.1 Acceptance of Input Wander

For T1(1.544 MHz), E1(2.048 MHz) and J2(6.312 MHz) input frequencies, the DPLL will accept a wander of up to  $\pm 1023 UI_{pp}$  at 0.1 Hz to conform with the relevant specifications. For the 8 kHz (frame rate) and 64 kHz (the divided down output for T3/E3) input frequencies, the wander acceptance is limited to  $\pm 1 UI$  (0.1 Hz). This principle is illustrated in Table 24.

### 9.7.2 Intrinsic Jitter

Intrinsic jitter is the jitter produced by a synchronizer and measured at its output. It is measured by applying a jitter free reference signal to the input of the device, and measuring its output jitter. Intrinsic jitter may also be measured when the device is in a non synchronizing mode such as free running or holdover, by measuring the output jitter of the device. Intrinsic jitter is usually measured with various band-limiting filters, depending on the applicable standards.

The intrinsic jitter in the DPLL is reduced to less than 1 ns p-p<sup>1</sup> by an internal Tapped Delay Line (TDL).

### 9.7.3 Jitter Tolerance

Jitter tolerance is a measure of the ability of a PLL to operate properly without cycle slips (i.e., remain in lock and/or regain lock in the presence of large jitter magnitudes at various jitter frequencies) when jitter is applied to its reference. The applied jitter magnitude and the jitter frequency depends on the applicable standards.

The DPLL's jitter tolerance can be programmed to meet Telcordia GR-1244-CORE DS1 reference input jitter tolerance requirements.

### 9.7.4 Jitter Transfer

Jitter transfer or jitter attenuation refers to the magnitude of jitter at the output of a device for a given amount of jitter at the input of the device. Input jitter is applied at various amplitudes and frequencies, and output jitter is measured with various filters depending on the applicable standards.

Since intrinsic jitter is always present, jitter attenuation will appear to be lower for small input jitter signals than larger ones. Consequently, accurate jitter transfer function measurements are usually made with large input jitter signals (e.g., 75% of the specified maximum jitter tolerance).

The internal DPLL is a first order type 2 component, so a frequency offset doesn't result in a phase offset. Stratum 3 requires a -3 dB frequency of less than 3 Hz. The nature of the filter results in some peaking, resulting in a -3 dB frequency of 1.9 Hz and a 0.08 dB peak with a system clock frequency of 100 MHz assuming a p\_shift value of 2. The transfer function is illustrated in Figure 23 and in more detail in Figure 24. Increasing the p\_shift value increases the speed the DPLL will lock to the required frequency and reduces the peak, but also reduces the tolerance to jitter - so the p\_shift value must be programmed correctly to meet Stratum 3 or Stratum 4/4E jitter transfer characteristics. This is done automatically in the API.

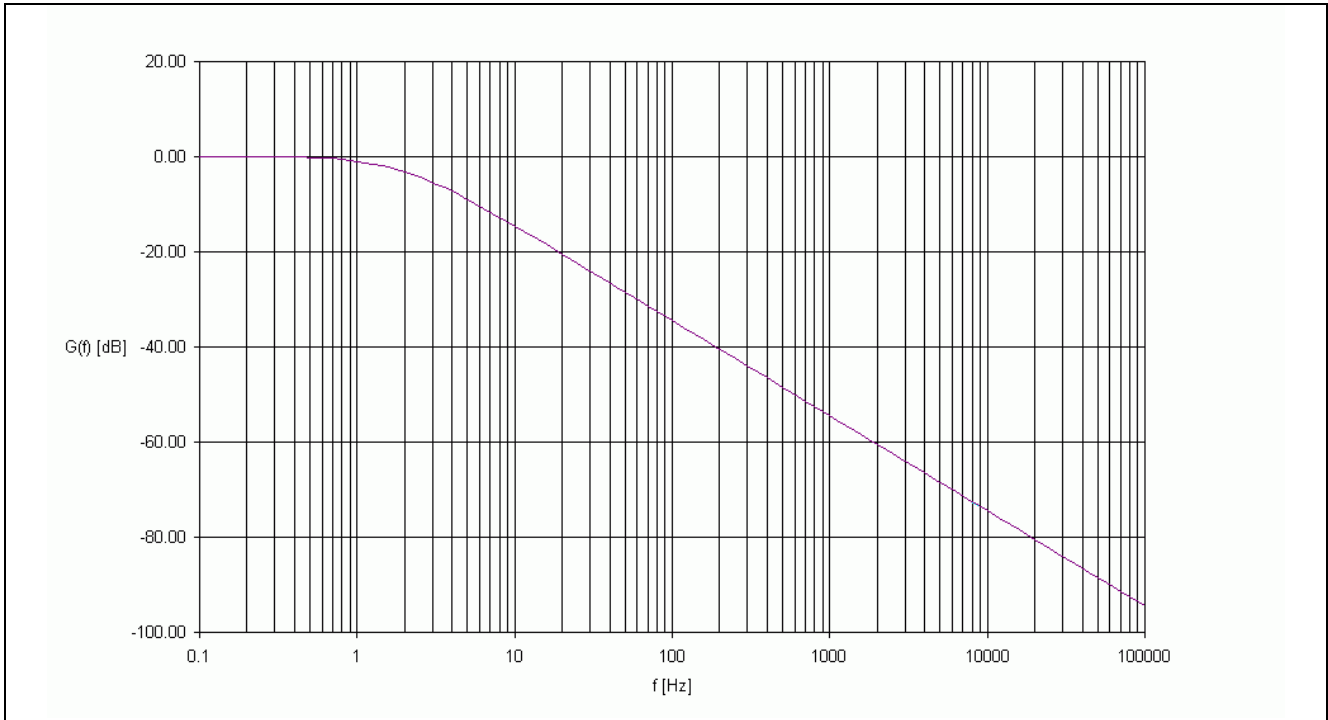
## 9.8 Maximum Time Interval Error (MTIE)

In order to meet several standards requirements, the phase shift of the DPLL output must be controlled. A potential phase shift occurs every time the DPLL is re-arranged by changing reference source signal, or the mode. In order to meet the requirements of Stratum 3, the DPLL will shift phase by no more than 20 ns per re-arrangement.

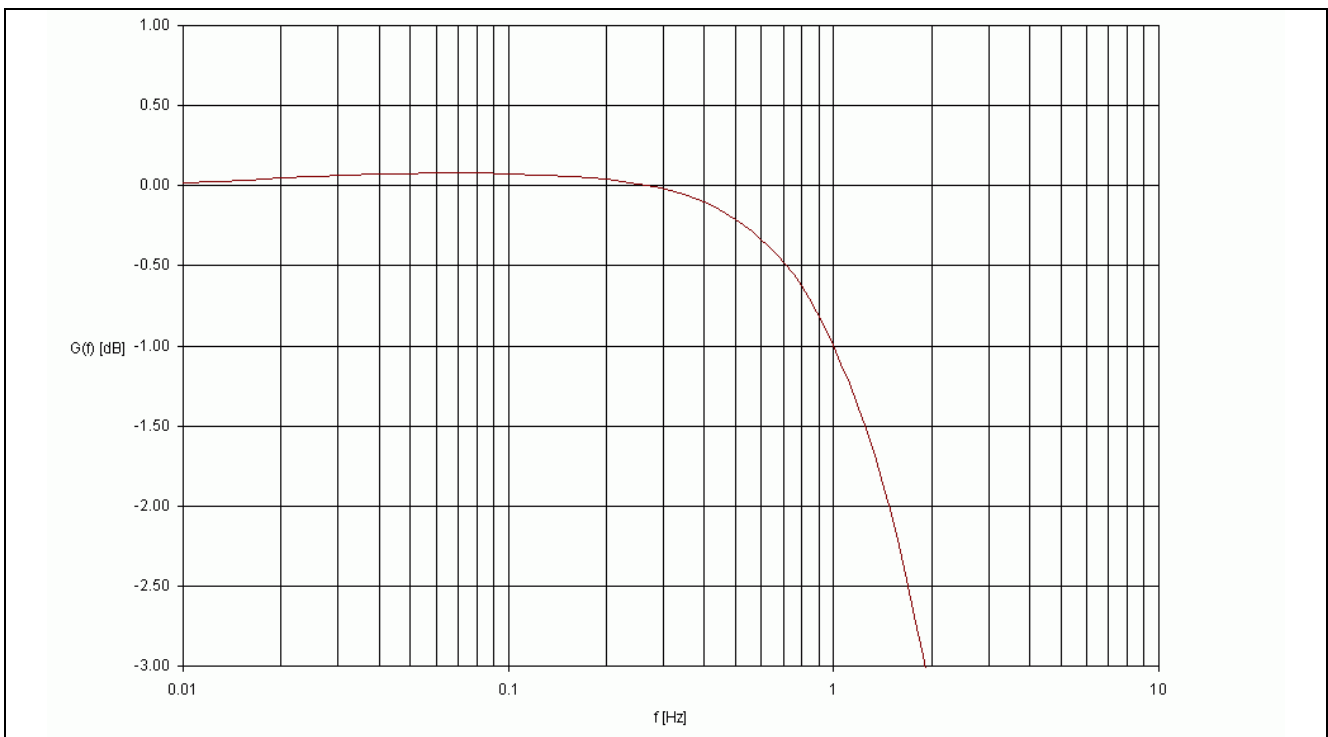
Additionally the speed at which the change occurs is also critical. A large step change in output frequency is undesirable. The rate of change is programmable using the skew register, up to a maximum of 15.4 ns / 125  $\mu$ s (124 ppm).

---

1. There are 2 exceptions to this. a) When reference is 8 kHz, and reference frequency offset relative to the master is small, jitter up to 1 master clock period is possible, i.e. 10 ns p-p. b) In holdover mode, if a huge amount of jitter had been present prior to entering holdover, then an additional 2 ns p-p is possible.



**Figure 23 - Jitter Transfer Function**



**Figure 24 - Jitter Transfer Function - Detail**

## 10.0 Memory Map and Register Definitions

All memory map and register definitions are included in the ZL50115/16/17/18/19/20 Programmers Model document.

## 11.0 DC Characteristics

### Absolute Maximum Ratings\*

Parameter	Symbol	Min.	Max.	Units
I/O Supply Voltage	$V_{DD\_IO}$	-0.5	5.0	V
Core Supply Voltage	$V_{DD\_CORE}$	-0.5	2.5	V
PLL Supply Voltage	$V_{DD\_PLL}$	-0.5	2.5	V
Input Voltage	$V_I$	-0.5	$V_{DD} + 0.5$	V
Input Voltage (5 V tolerant inputs)	$V_{I\_5V}$	-0.5	7.0	V
Continuous current at digital inputs	$I_{IN}$	-	$\pm 10$	mA
Continuous current at digital outputs	$I_O$	-	$\pm 15$	mA
Package power dissipation	PD	-	2.38	W
Storage Temperature	TS	-55	+125	°C

\* Exceeding these figures may cause permanent damage. Functional operation under these conditions is not guaranteed. Voltage measurements are with respect to ground ( $V_{SS}$ ) unless otherwise stated.

\* The core and PLL supply voltages must never be allowed to exceed the I/O supply voltage by more than 0.5 V during power-up. Failure to observe this rule could lead to a high-current latch-up state, possibly leading to chip failure, if sufficient cross-supply current is available. To be safe ensure the I/O supply voltage supply always rises earlier than the core and PLL supply voltages.

### Recommended Operating Conditions

Characteristics	Symbol	Min.	Typ.	Max.	Units	Test Condition
Operating Temperature	$T_{OP}$	-40	25	+85	°C	
Junction temperature	$T_J$	-40	-	125	°C	
Positive Supply Voltage, I/O	$V_{DD\_IO}$	3.0	3.3	3.6	V	
Positive Supply Voltage, Core	$V_{DD\_CORE}$	1.65	1.8	1.95	V	
Positive Supply Voltage, Core	$V_{DD\_PLL}$	1.65	1.8	1.95	V	
Input Voltage Low - all inputs	$V_{IL}$	-	-	0.8	V	
Input Voltage High	$V_{IH}$	2.0	-	$V_{DD\_IO}$	V	
Input Voltage High, 5 V tolerant inputs	$V_{IH\_5V}$	2.0	-	5.5	V	

Typical figures are at 25°C and are for design aid only, they are not guaranteed and not subject to production testing. Voltage measurements are with respect to ground ( $V_{SS}$ ) unless otherwise stated.

**DC Electrical Characteristics** - Typical characteristics are at 1.8 V core, 3.3 V I/O, 25°C and typical processing. The min. and max. values are defined over all process conditions, from -40 to 125°C junction temperature, core voltage 1.65 to 1.95 V and I/O voltage 3.0 and 3.6 V unless otherwise stated.

Characteristics	Symbol	Min.	Typ.	Max.	Units.	Test Condition
Input Leakage	$I_{LEIP}$			±1	μA	No pull up/down $V_{DD\_IO} = 3.6 V$
Output (High impedance) Leakage	$I_{LEOP}$			2	μA	No pull up/down $V_{DD\_IO} = 3.6 V$
Input Capacitance	$C_{IP}$		1		pF	
Output Capacitance	$C_{OP}$		4		pF	
Pullup Current	$I_{PU}$		-27		μA	Input at 0 V
Pullup Current, 5 V tolerant inputs	$I_{PU\_5V}$		-110		μA	Input at 0 V
Pulldown Current	$I_{PD}$		27		μA	Input at $V_{DD\_IO}$
Pulldown Current, 5 V tolerant inputs	$I_{PD\_5V}$		110		μA	Input at $V_{DD\_IO}$
Core 1.8 V supply current	$I_{DD\_CORE}$			950	mA	Note 1,2
PLL 1.8 V supply current	$I_{DD\_PLL}$			1.30	mA	
I/O 3.3 V supply current	$I_{DD\_IO}$			120	mA	Note 1,2

Note 1: The IO and Core supply current worst case figures apply to different scenarios and can not simply be summed for a total figure. For a clearer indication of power consumption, please refer to Section 13.0.

Note 2: Worst case assumes the maximum number of active contexts and channels. Figures are for the ZL50120. For an indication of typical power consumption, please refer to Section 13.0.

### Input Levels

Characteristics	Symbol	Min.	Typ.	Max.	Units	Test Condition
Input Low Voltage	$V_{IL}$			0.8	V	
Input High Voltage	$V_{IH}$	2.0			V	
Positive Schmitt Threshold	$V_{T+}$		1.6		V	
Negative Schmitt Threshold	$V_{T-}$		1.2		V	

### Output Levels

Characteristics	Symbol	Min.	Typ.	Max.	Units	Test Condition
Output Low Voltage	$V_{OL}$			0.4	V	$I_{OL} = 6 mA$ . $I_{OL} = 12 mA$ for packet interface (m*) pins and GPIO pins. $I_{OL} = 24 mA$ for LED pins.
Output High Voltage	$V_{OH}$	2.4			V	$I_{OH} = 6 mA$ . $I_{OH} = 12 mA$ for packet interface (m*) pins and GPIO pins. $I_{OH} = 24 mA$ for LED pins.

## 12.0 AC Characteristics

### 12.1 TDM Interface Timing - ST-BUS

The TDM Bus either operates in Slave mode, where the TDM clocks for each stream are provided by the device sourcing the data, or Master mode, where the TDM clocks are generated from the ZL5011x.

#### 12.1.1 ST-BUS Slave Clock Mode

##### TDM ST-BUS Slave Timing Specification

Data Format	Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
ST-BUS 8.192 Mbps mode	TDM_CLKi Period	$t_{C16iP}$	54	60	66	ns	
	TDM_CLKi High	$t_{C16iH}$	27	-	33	ns	
	TDM_CLKi Low	$t_{C16iL}$	27	-	33	ns	
ST-BUS 2.048 Mbps mode	TDM_CLKi Period	$t_{C4iP}$	-	244.1	-	ns	
	TDM_CLKi High	$t_{C4iH}$	110	-	134	ns	
	TDM_CLKi Low	$t_{C4iL}$	110	-	134	ns	
All Modes	$\overline{\text{TDM\_FOi}}$ Width 8.192 Mbps 2.048 Mbps	$t_{FOiW}$	50 200	- -	- 300	ns	
	$\overline{\text{TDM\_FOi}}$ Setup Time	$t_{FOiS}$	5	-	-	ns	With respect to TDM_CLKi falling edge
	$\overline{\text{TDM\_FOi}}$ Hold Time	$t_{FOiH}$	5	-	-	ns	With respect to TDM_CLKi falling edge
	TDM_STo Delay	$t_{SToD}$	1	-	20	ns	With respect to TDM_CLKi Load $C_L = 50$ pF
	TDM_STi Setup Time	$t_{STiS}$	5	-	-	ns	With respect to TDM_CLKi
	TDM_STi Hold Time	$t_{STiH}$	5	-	-	ns	With respect to TDM_CLKi

In synchronous mode the clock must be within the locking range of the DPLL to function correctly ( $\pm 245$  ppm). In asynchronous mode, the clock may be any frequency.

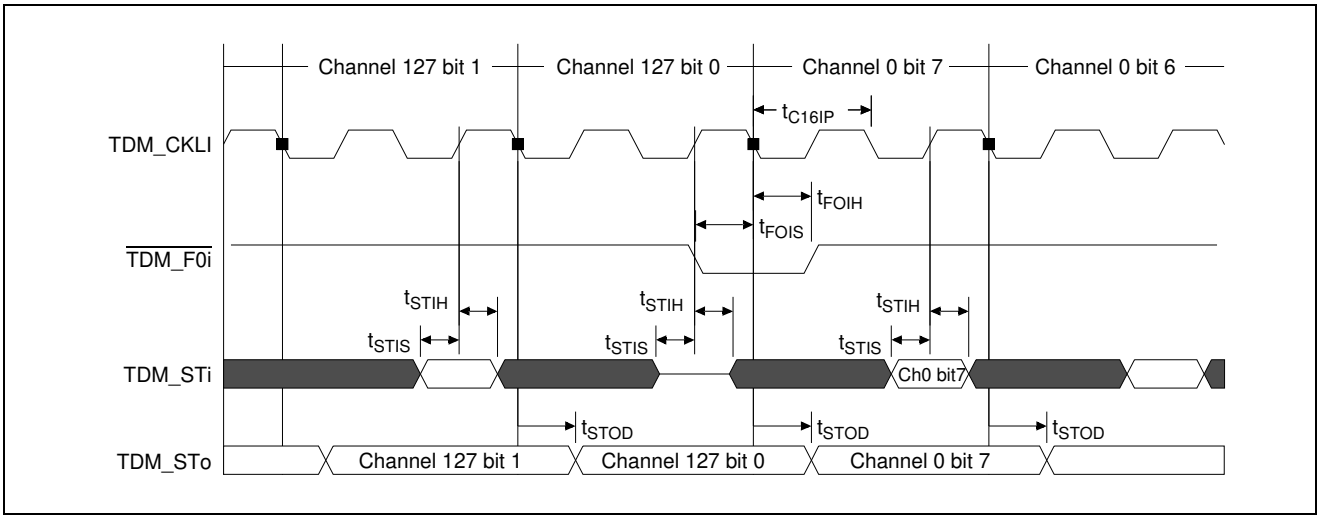


Figure 25 - TDM ST-BUS Slave Mode Timing at 8.192 Mbps

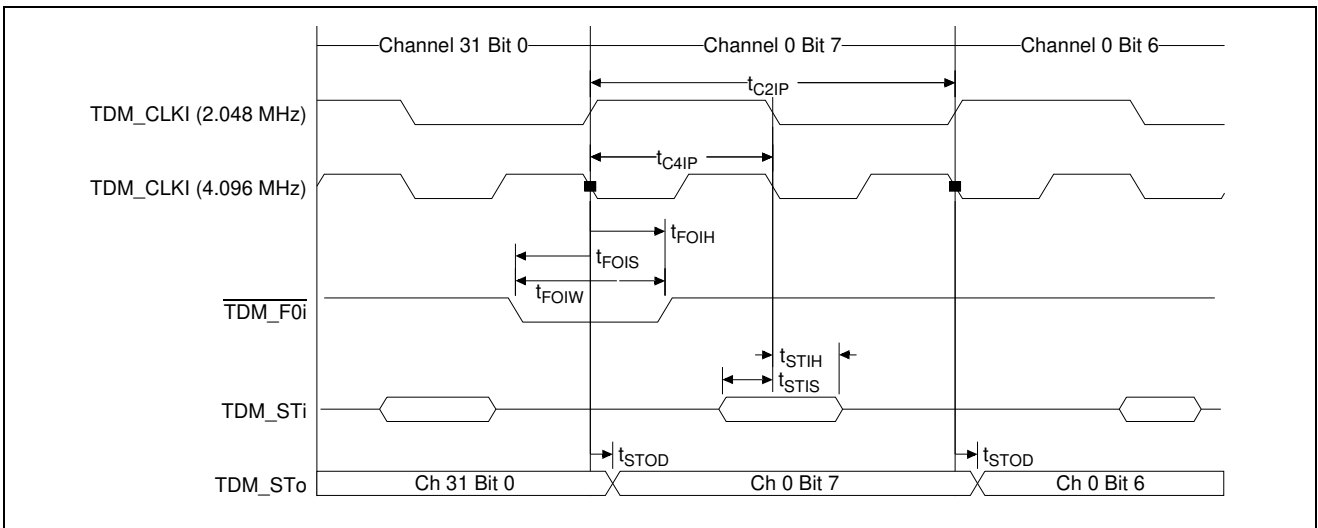


Figure 26 - TDM ST-BUS Slave Mode Timing at 2.048 Mbps

12.1.2 ST-BUS Master Clock Mode

Data Format	Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
ST-BUS 8.192 Mbps mode	TDM_CLKo Period	$t_{C16OP}$	54.0	61.0	68.0	ns	
	TDM_CLKo High	$t_{C16OH}$	23.0	-	37.0	ns	
	TDM_CLKo Low	$t_{C16OL}$	23.0	-	37.0	ns	
ST-BUS 2.048 Mbps mode	TDM_CLKo Period	$t_{C4OP}$	237.0	244.1	251.0	ns	
	TDM_CLKo High	$t_{C4OH}$	115.0	-	129.0	ns	
	TDM_CLKo Low	$t_{C4OL}$	115.0	-	129.0	ns	
All Modes	TDM_F0o Delay	$t_{FOD}$	-	-	25	ns	With respect to TDM_CLKo falling edge
	TDM_STo Delay Active-Active	$t_{STOD}$	-	-	5	ns	With respect to TDM_CLKo falling edge
	TDM_STo Delay Active to HiZ and HiZ to Active	$t_{DZ}, t_{ZD}$	-	-	33	ns	With respect to TDM_CLKo falling edge
	TDM_STi Setup Time	$t_{STIS}$	5	-	-	ns	With respect to TDM_CLKo
	TDM_STi Hold Time	$t_{STIH}$	5	-	-	ns	With respect to TDM_CLKo

Table 25 - TDM ST-BUS Master Timing Specification

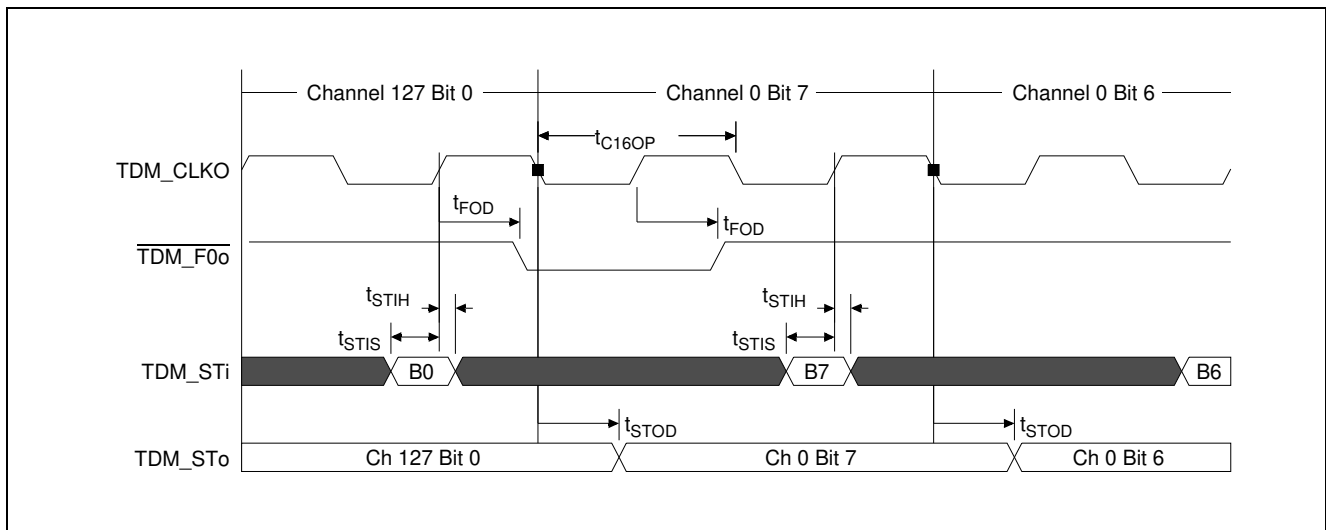


Figure 27 - TDM Bus Master Mode Timing at 8.192 Mbps



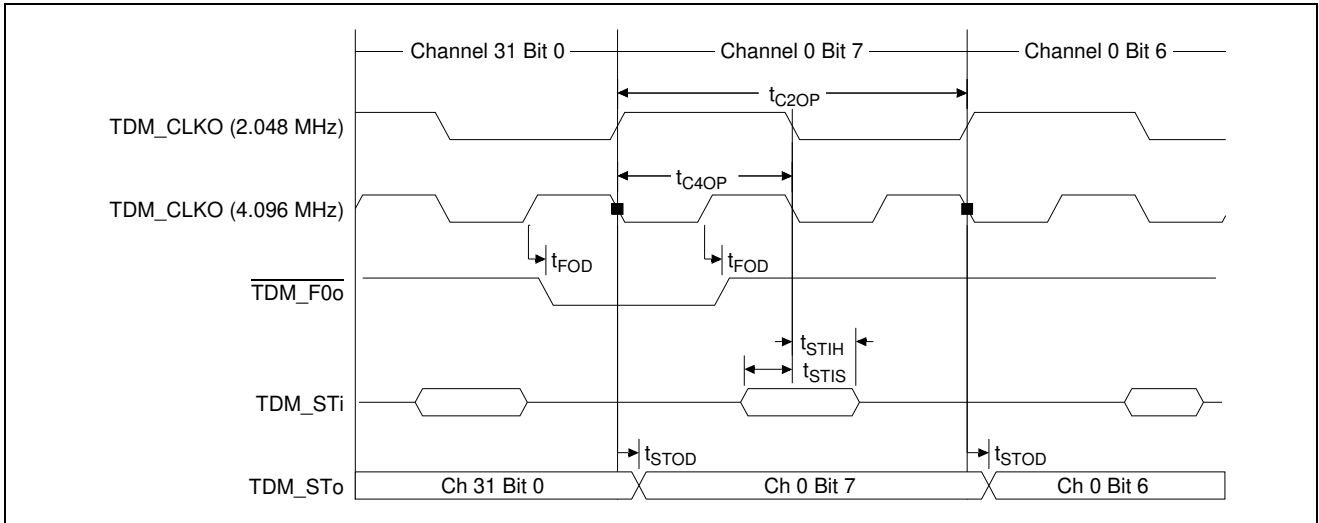


Figure 28 - TDM Bus Master Mode Timing at 2.048 Mbps

12.2 TDM Interface Timing - H.110 Mode

These parameters are based on the H.110 Specification from the Enterprise Computer Telephony Forum (ECTF) 1997.

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
TDM_C8 Period	t <sub>C8P</sub>	122.066-Φ	122	122.074+Φ	ns	Note 1 Note 2
TDM_C8 High	t <sub>C8H</sub>	63-Φ	-	69+Φ	ns	
TDM_C8 Low	t <sub>C8L</sub>	63-Φ	-	69+Φ	ns	
TDM_D Output Delay	t <sub>DOD</sub>	0	-	11	ns	Load - 12 pF
TDM_D Output to HiZ	t <sub>DOZ</sub>	-	-	33	ns	Load - 12 pF Note 3
TDM_D HiZ to Output	t <sub>ZDO</sub>	0	-	11	ns	Load - 12 pF Note 3
TDM_D Input Delay to Valid	t <sub>DV</sub>	0	-	83	ns	Note 4
TDM_D Input Delay to Invalid	t <sub>DIV</sub>	102	-	112	ns	Note 4
TDM_FRAME width	t <sub>FP</sub>	90	122	180	ns	Note 5
TDM_FRAME setup	t <sub>FS</sub>	45	-	90	ns	
TDM_FRAME hold	t <sub>FH</sub>	45	-	90	ns	
Phase Correction	F	0	-	10	ns	Note 6

Table 26 - TDM H.110 Timing Specification

- Note 1: TDM\_C8 and TDM\_FRAME signals are required to meet the same timing standards and so are not defined independently.
- Note 2: TDM\_C8 corresponds to pin TDM\_CLKi.
- Note 3: t<sub>DOZ</sub> and t<sub>ZDO</sub> apply at every time-slot boundary.
- Note 4: Refer to H.110 Standard from Enterprise Computer Telephony Forum (ECTF) for the source of these numbers.
- Note 5: The TDM\_FRAME signal is centred on the rising edge of TDM\_C8. All timing measurements are based on this rising edge point; TDM\_FRAME corresponds to pin TDM\_F0i.
- Note 6: Phase correction (Φ) results from DPLL timing corrections.

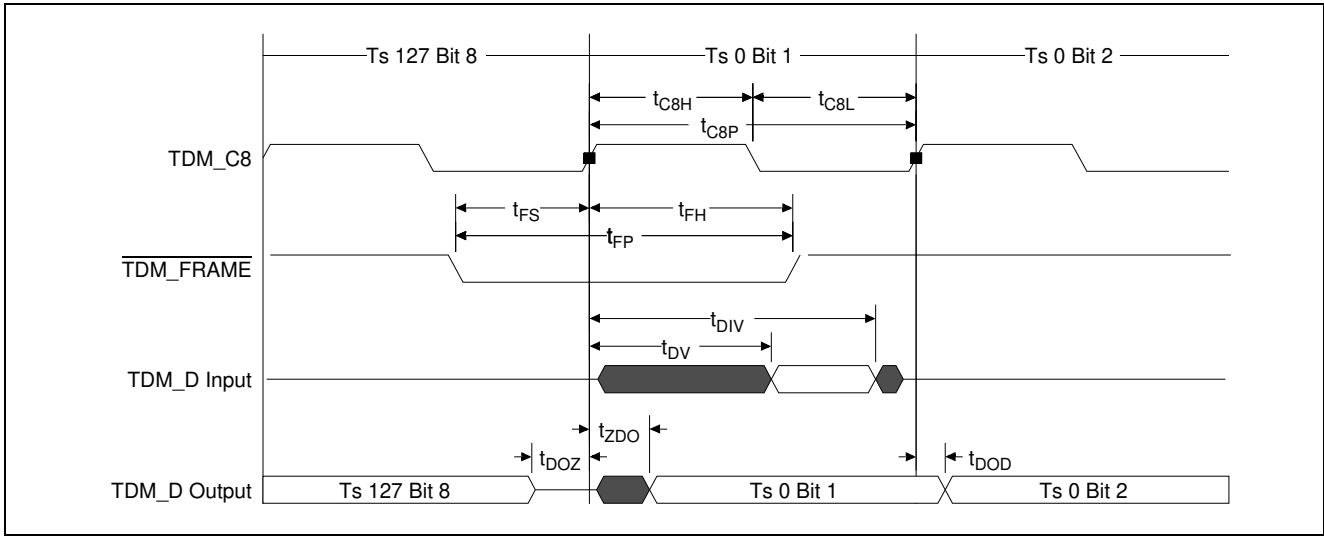


Figure 29 - H.110 Timing Diagram

### 12.3 TDM Interface Timing - H-MVIP

These parameters are based on the Multi-Vendor Integration Protocol (MVIP) specification for an H-MVIP Bus, Release 1.1a (1997).

Positive transitions of TDM\_C2 are synchronous with the falling edges of  $\overline{TDM\_C4}$  and  $\overline{TDM\_C16}$ . The signals TDM\_C2,  $\overline{TDM\_C4}$  and  $\overline{TDM\_C16}$  correspond with pins TDM\_CLKi. The signals  $\overline{TDM\_F0}$  correspond with pins TDM\_F0i. The signals TDM\_HDS correspond with pins TDM\_STi and TDM\_STo.

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
TDM_C2 Period	$t_{C2P}$	487.8	488.3	488.8	ns	
TDM_C2 High	$t_{C2H}$	220	-	268	ns	
TDM_C2 Low	$t_{C2L}$	220	-	268	ns	
$\overline{TDM\_C4}$ Period	$t_{C4P}$	243.9	244.1	244.4	ns	
$\overline{TDM\_C4}$ High	$t_{C4H}$	110	-	134	ns	
$\overline{TDM\_C4}$ Low	$t_{C4L}$	110	-	134	ns	
$\overline{TDM\_C16}$ Period	$t_{C16P}$	60.9	61.0	61.1	ns	
$\overline{TDM\_C16}$ High	$t_{C16H}$	30	-	31	ns	
$\overline{TDM\_C16}$ Low	$t_{C16L}$	30	-	31	ns	
TDM_HDS Output Delay	$t_{PD}$	-	-	30	ns	At 8.192 Mbps
TDM_HDS Output Delay	$t_{PD}$	-	-	100	ns	At 2.048 Mbps
TDM_HDS Output to HiZ	$t_{HZD}$	-	-	30	ns	
TDM_HDS Input Setup	$t_S$	30	-	0	ns	
TDM_HDS Input Hold	$t_H$	30	-	0	ns	
$\overline{TDM\_F0}$ width	$t_{FW}$	200	244	300	ns	

Table 27 - TDM H-MVIP Timing Specification

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
TDM_F0 setup	$t_{FS}$	50	-	150	ns	
TDM_F0 hold	$t_{FH}$	50	-	150	ns	

Table 27 - TDM H-MVIP Timing Specification (continued)

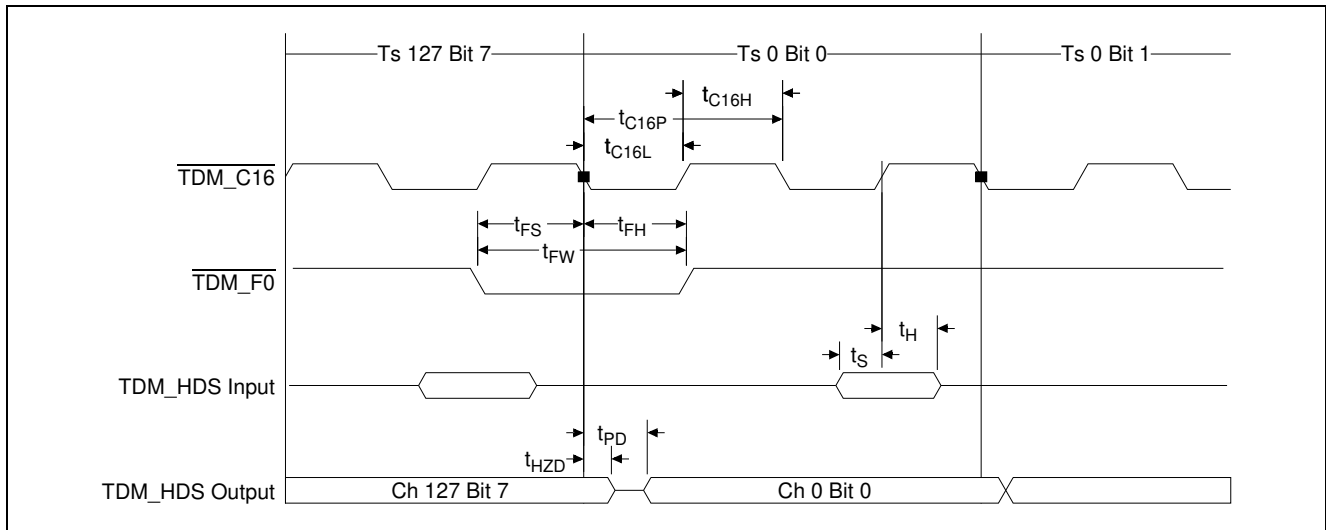


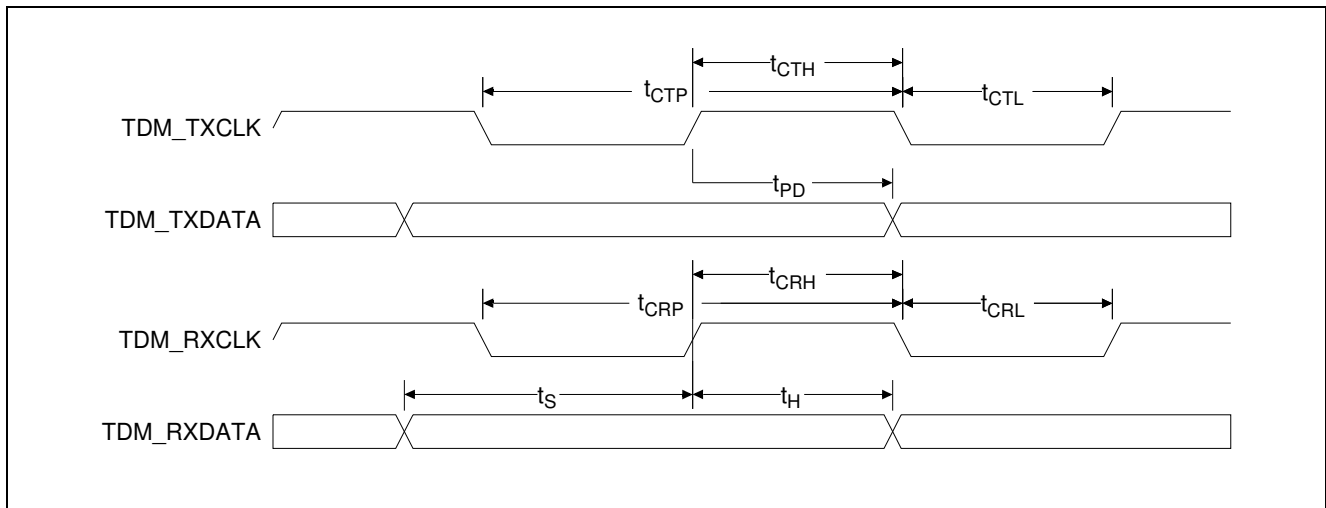
Figure 30 - TDM - H-MVIP Timing Diagram for 16 MHz Clock (8.192 Mbps)

### 12.4 TDM LIU Interface Timing

The TDM Interface can be used to directly drive into a Line Interface Unit (LIU). The interface can work in this mode with E1, DS1, J2, E3 and DS3. The frame pulse is not present, just data and clock is transmitted and received. Table 28 shows timing for DS3, which would be the most stringent requirement.

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
TDM_TXCLK Period	$t_{CTP}$		22.353		ns	DS3 clock
TDM_TXCLK High	$t_{CTH}$	6.7			ns	
TDM_TXCLK Low	$t_{CTL}$	6.7			ns	
TDM_RXCLK Period	$t_{CRP}$		22.353		ns	DS3 clock
TDM_RXCLK High	$t_{CRH}$	9.0			ns	
TDM_RXCLK Low	$t_{CRL}$	9.0			ns	
TDM_TXDATA Output Delay	$t_{PD}$	3	-	10	ns	
TDM_RXDATA Input Setup	$t_S$	6			ns	
TDM_RXDATA Input Hold	$t_H$	3			ns	

**Table 28 - TDM - LIU Structured Transmission/Reception**



**Figure 31 - TDM-LIU Structured Transmission/Reception**

**12.5 PAC Interface Timing**

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
TDM_CLKiP High / Low Pulsewidth	$t_{CPP}$	10	-	-	ns	
TDM_CLKiS High / Low Pulsewidth	$t_{CSP}$	10	-	-	ns	

**Table 29 - PAC Timing Specification**

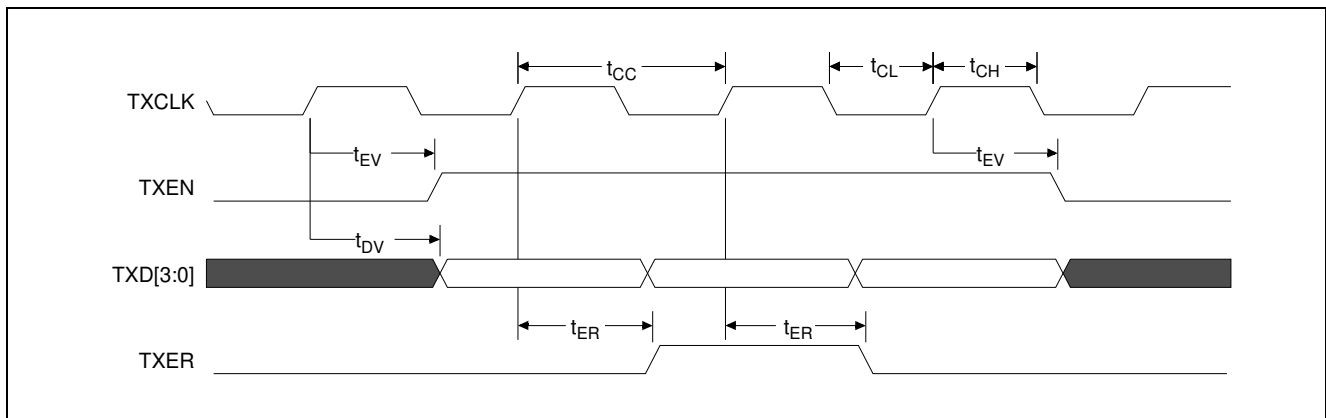
**12.6 Packet Interface Timing**

Data for the MII/GMII/TBI packet switching is based on Specification IEEE Std. 802.3 - 2000.

**12.6.1 MII Transmit Timing**

Parameter	Symbol	100 Mbps			Units	Notes
		Min.	Typ.	Max.		
TXCLK period	$t_{CC}$	-	40	-	ns	
TXCLK high time	$t_{CHI}$	14	-	26	ns	
TXCLK low time	$t_{CLO}$	14	-	26	ns	
TXCLK rise time	$t_{CR}$	-	-	5	ns	
TXCLK fall time	$t_{CF}$	-	-	5	ns	
TXCLK rise to TXD[3:0] active delay (TXCLK rising edge)	$t_{DV}$	1	-	25	ns	Load = 25 pF
TXCLK to TXEN active delay (TXCLK rising edge)	$t_{EV}$	1	-	25	ns	Load = 25 pF
TXCLK to TXER active delay (TXCLK rising edge)	$t_{ER}$	1	-	25	ns	Load = 25 pF

**Table 30 - MII Transmit Timing - 100 Mbps**



**Figure 32 - MII Transmit Timing Diagram**

12.6.2 MII Receive Timing

Parameter	Symbol	100 Mbps			Units	Notes
		Min.	Typ.	Max.		
RXCLK period	$t_{CC}$	-	40	-	ns	
RXCLK high wide time	$t_{CH}$	14	20	26	ns	
RXCLK low wide time	$t_{CL}$	14	20	26	ns	
RXCLK rise time	$t_{CR}$	-	-	5	ns	
RXCLK fall time	$t_{CF}$	-	-	5	ns	
RXD[3:0] setup time (RXCLK rising edge)	$t_{DS}$	10	-	-	ns	
RXD[3:0] hold time (RXCLK rising edge)	$t_{DH}$	5	-	-	ns	
RXDV input setup time (RXCLK rising edge)	$t_{DVS}$	10	-	-	ns	
RXDV input hold time (RXCLK rising edge)	$t_{DVH}$	5	-	-	ns	
RXER input setup time (RXCLK edge)	$t_{ERS}$	10	-	-	ns	
RXER input hold time (RXCLK rising edge)	$t_{ERH}$	5	-	-	ns	

Table 31 - MII Receive Timing - 100 Mbps

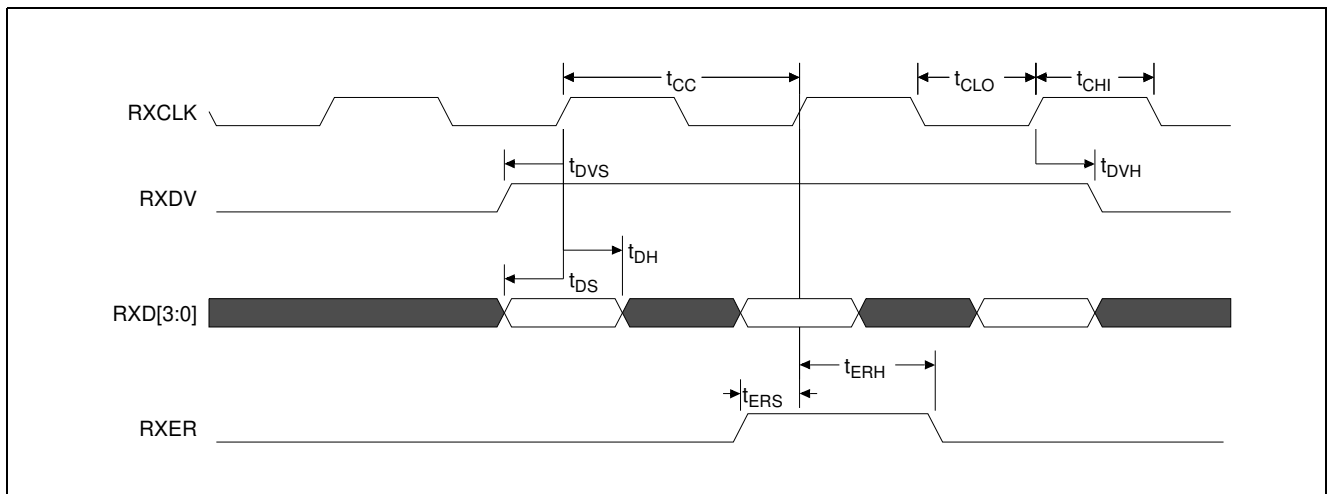


Figure 33 - MII Receive Timing Diagram

12.6.3 GMII Transmit Timing

Parameter	Symbol	1000 Mbps			Units	Notes
		Min.	Typ.	Max.		
GTXCLK period	$t_{GC}$	7.5	-	8.5	ns	
GTXCLK high time	$t_{GCH}$	2.5	-	-	ns	
GTXCLK low time	$t_{GCL}$	2.5	-	-	ns	
GTXCLK rise time	$t_{GCR}$	-	-	1	ns	
GTXCLK fall time	$t_{GCF}$	-	-	1	ns	
GTXCLK rise to TXD[7:0] active delay	$t_{DV}$	0.5	-	5	ns	Load = 25 pF
GTXCLK rise to TXEN active delay	$t_{EV}$	0.5	-	5	ns	Load = 25 pF
GTXCLK rise to TXER active delay	$t_{ER}$	0.5	-	5	ns	Load = 25 pF

Table 32 - GMII Transmit Timing - 1000 Mbps

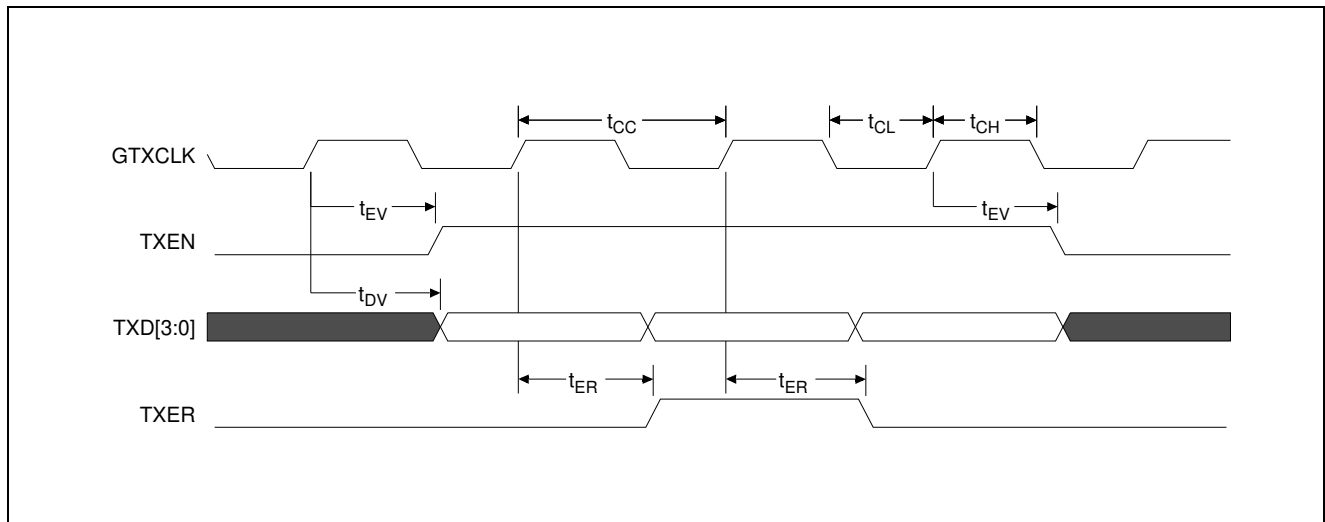


Figure 34 - GMII Transmit Timing Diagram

12.6.4 GMII Receive Timing

Parameter	Symbol	1000 Mbps			Units	Notes
		Min.	Typ.	Max.		
RXCLK period	$t_{CC}$	7.5	-	8.5	ns	
RXCLK high wide time	$t_{CH}$	2.5	-	-	ns	
RXCLK low wide time	$t_{CL}$	2.5	-	-	ns	
RXCLK rise time	$t_{CR}$	-	-	1	ns	
RXCLK fall time	$t_{CF}$	-	-	1	ns	
RXD[7:0] setup time (RXCLK rising edge)	$t_{DS}$	2	-	-	ns	
RXD[7:0] hold time (RXCLK rising edge)	$t_{DH}$	1	-	-	ns	
RXDV setup time (RXCLK rising edge)	$t_{DVS}$	2	-	-	ns	
RXDV hold time (RXCLK rising edge)	$t_{DVH}$	1	-	-	ns	
RXER setup time (RXCLK rising edge)	$t_{ERS}$	2	-	-	ns	
RXER hold time (RXCLK rising edge)	$t_{ERH}$	1	-	-	ns	

Table 33 - GMII Receive Timing - 1000 Mbps

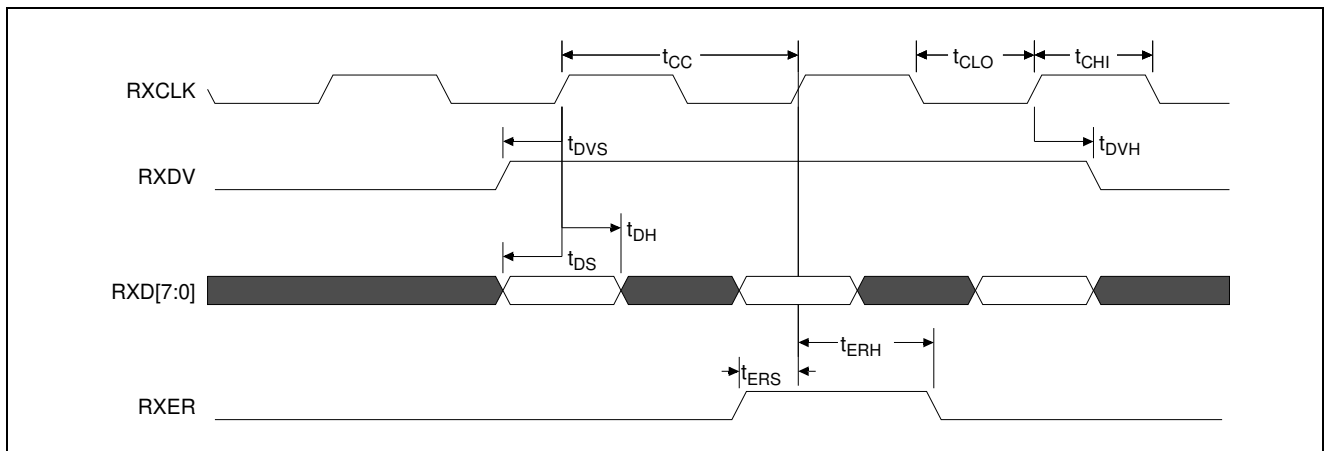


Figure 35 - GMII Receive Timing Diagram



12.6.5 TBI Interface Timing

Parameter	Symbol	1000 Mbps			Units	Notes
		Min.	Typ.	Max.		
GTXCLK period	$t_{GC}$	7.5	-	8.5	ns	
GTXCLK high wide time	$t_{GH}$	2.5	-	-	ns	
GTXCLK low wide time	$t_{GL}$	2.5	-	-	ns	
TXD[9:0] Output Delay (GTXCLK rising edge)	$t_{DV}$	0.1	-	2.4		Load = 10 pF Note 1
RCB0/RBC1 period	$t_{RC}$	15	16	17	ns	
RCB0/RBC1 high wide time	$t_{RH}$	5	-	-	ns	
RCB0/RBC1 low wide time	$t_{RL}$	5	-	-	ns	
RCB0/RBC1 rise time	$t_{RR}$	-	-	2	ns	
RCB0/RBC1 fall time	$t_{RF}$	-	-	2	ns	
RXD[9:0] setup time (RCB0 rising edge)	$t_{DS}$	2	-	-	ns	
RXD[9:0] hold time (RCB0 rising edge)	$t_{DH}$	1	-	-	ns	
REFCLK period	$t_{FC}$	7.5	-	8.5	ns	
REFCLK high wide time	$t_{FH}$	2.5	-	-	ns	
REFCLK low wide time	$t_{FL}$	2.5	-	-	ns	

Table 34 - TBI Timing - 1000 Mbps

Note1: These measurements were obtained through simulation and lab measurement using a 10 pF load. See ZL5011x Design Manual section "7.1.3.1 TBI Interface Timing" for workaround when using the TBI interface.

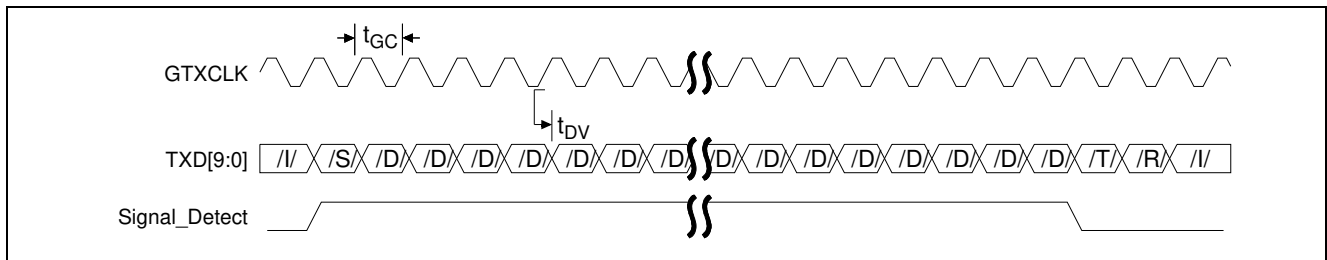


Figure 36 - TBI Transmit Timing Diagram

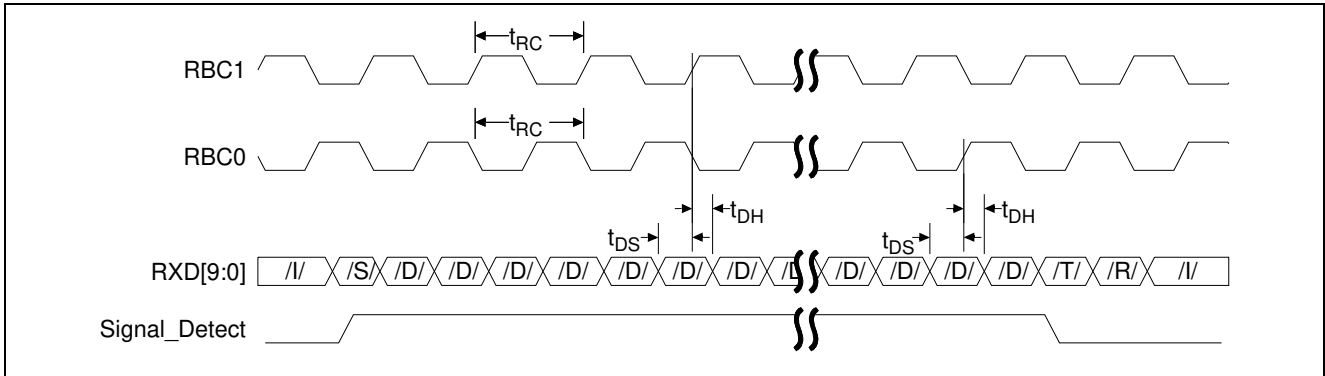


Figure 37 - TBI Receive Timing Diagram

12.6.6 Management Interface Timing

The management interface is common for all inputs and consists of a serial data I/O line and a clock line.

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
M_MDC Clock Output period	$t_{MP}$	1990	2000	2010	ns	Note 1
M_MDC high	$t_{MHI}$	900	1000	1100	ns	
M_MDC low	$t_{MLO}$	900	1000	1100	ns	
M_MDC rise time	$t_{MR}$	-	-	5	ns	
M_MDC fall time	$t_{MF}$	-	-	5	ns	
M_MDIO setup time (MDC rising edge)	$t_{MS}$	10	-	-	ns	Note 1
M_MDIO hold time (M_MDC rising edge)	$t_{MH}$	0	-	-	ns	Note 1
M_MDIO Output Delay (M_MDC rising edge)	$t_{MD}$	1	-	300	ns	Note 2

Table 35 - MAC Management Timing Specification

Note 1: Refer to Clause 22 in IEEE802.3 (2000) Standard for input/output signal timing characteristics.

Note 2: Refer to Clause 22C.4 in IEEE802.3 (2000) Standard for output load description of MDIO.

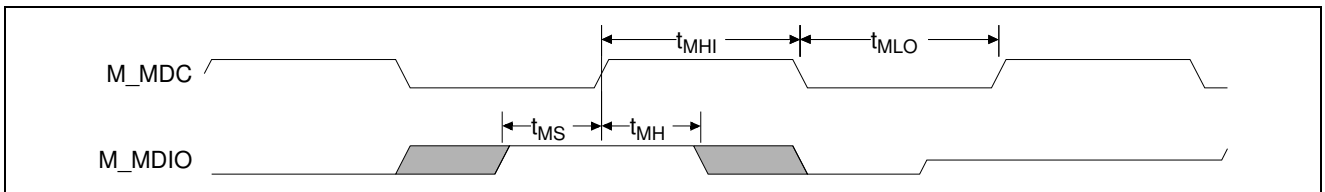


Figure 38 - Management Interface Timing for Ethernet Port - Read

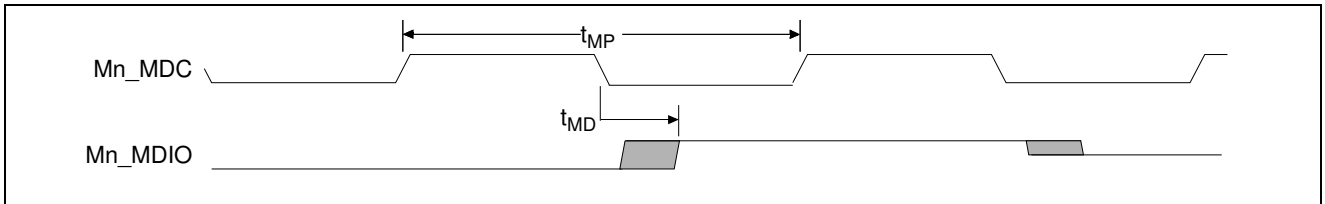


Figure 39 - Management Interface Timing for Ethernet Port - Write

## 12.7 CPU Interface Timing

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
CPU_CLK Period	$t_{CC}$		15.152		ns	
CPU_CLK High Time	$t_{CCH}$	6			ns	
CPU_CLK Low Time	$t_{CCL}$	6			ns	
CPU_CLK Rise Time	$t_{CCR}$			4	ns	
CPU_CLK Fall Time	$t_{CCF}$			4	ns	
CPU_ADDR[23:2] Setup Time	$t_{CAS}$	4			ns	
CPU_ADDR[23:2] Hold Time	$t_{CAH}$	2			ns	
CPU_DATA[31:0] Setup Time	$t_{CDS}$	4			ns	
CPU_DATA[31:0] Hold Time	$t_{CDH}$	2			ns	
$\overline{\text{CPU\_CS}}$ Setup Time	$t_{CSS}$	4			ns	
$\overline{\text{CPU\_CS}}$ Hold Time	$t_{CSH}$	2			ns	
$\overline{\text{CPU\_WE}}/\overline{\text{CPU\_OE}}$ Setup Time	$t_{CES}$	5			ns	
$\overline{\text{CPU\_WE}}/\overline{\text{CPU\_OE}}$ Hold Time	$t_{CEH}$	2			ns	
CPU_ $\overline{\text{TS}}$ _ALE Setup Time	$t_{CTS}$	4			ns	
CPU_ $\overline{\text{TS}}$ _ALE Hold Time	$t_{CTH}$	2			ns	
CPU_SDACK1/CPU_SDACK2 Setup Time	$t_{CKS}$	2			ns	
CPU_SDACK1/CPU_SDACK2 Hold Time	$t_{CKH}$	2			ns	Note 1
CPU_ $\overline{\text{TA}}$ Output Valid Delay	$t_{CTV}$	2		11.3	ns	Note 1, 2
CPU_DREQ0/CPU_DREQ1 Output Valid Delay	$t_{CWV}$	2		6	ns	Note 1
CPU_IREQ0/CPU_IREQ1 Output Valid Delay	$t_{CRV}$	2		6	ns	Note 1
CPU_DATA[31:0] Output Valid Delay	$t_{CDV}$	2		7	ns	Note 1
$\overline{\text{CPU\_CS}}$ to Output Data Valid	$t_{SDV}$	3.2		10.4	ns	
$\overline{\text{CPU\_OE}}$ to Output Data Valid	$t_{ODV}$	3.3		10.4	ns	
CPU_CLK(falling) to $\overline{\text{CPU\_TA}}$ Valid	$t_{OTV}$	3.2		9.5	ns	

**Table 36 - CPU Timing Specification**

Note 1: Load = 50 pF maximum

Note 2: The maximum value of  $t_{CTV}$  may cause setup violations if directly connected to the MPC8260. See Section 14.2 for details of how to accommodate this during board design.

The actual point where read/write data is transferred occurs at the positive clock edge following the assertion of CPU\_ $\overline{\text{TA}}$ , not at the positive clock edge during the assertion of CPU\_ $\overline{\text{TA}}$ .

The CPU\_ $\overline{\text{TA}}$  maximum assertion time is 4  $\mu\text{s}$ .

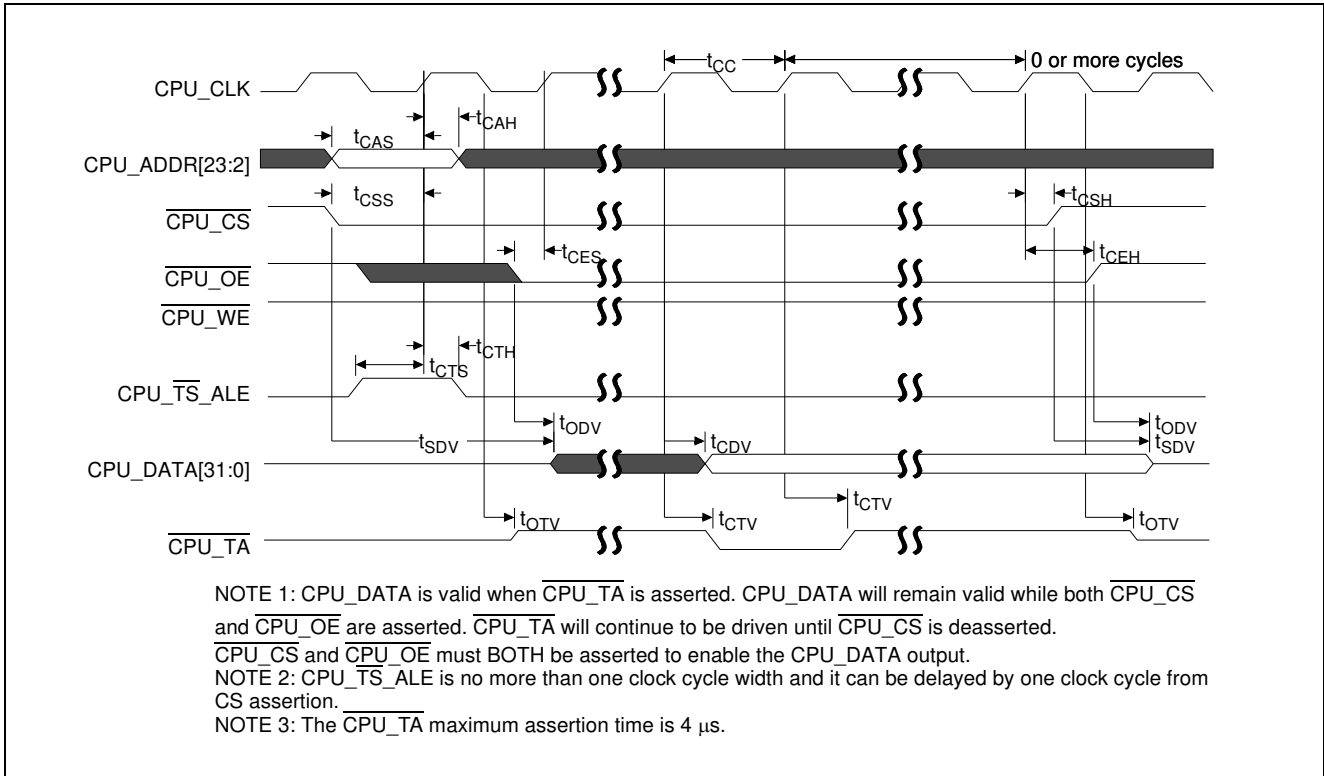


Figure 40 - CPU Read - MPC8260

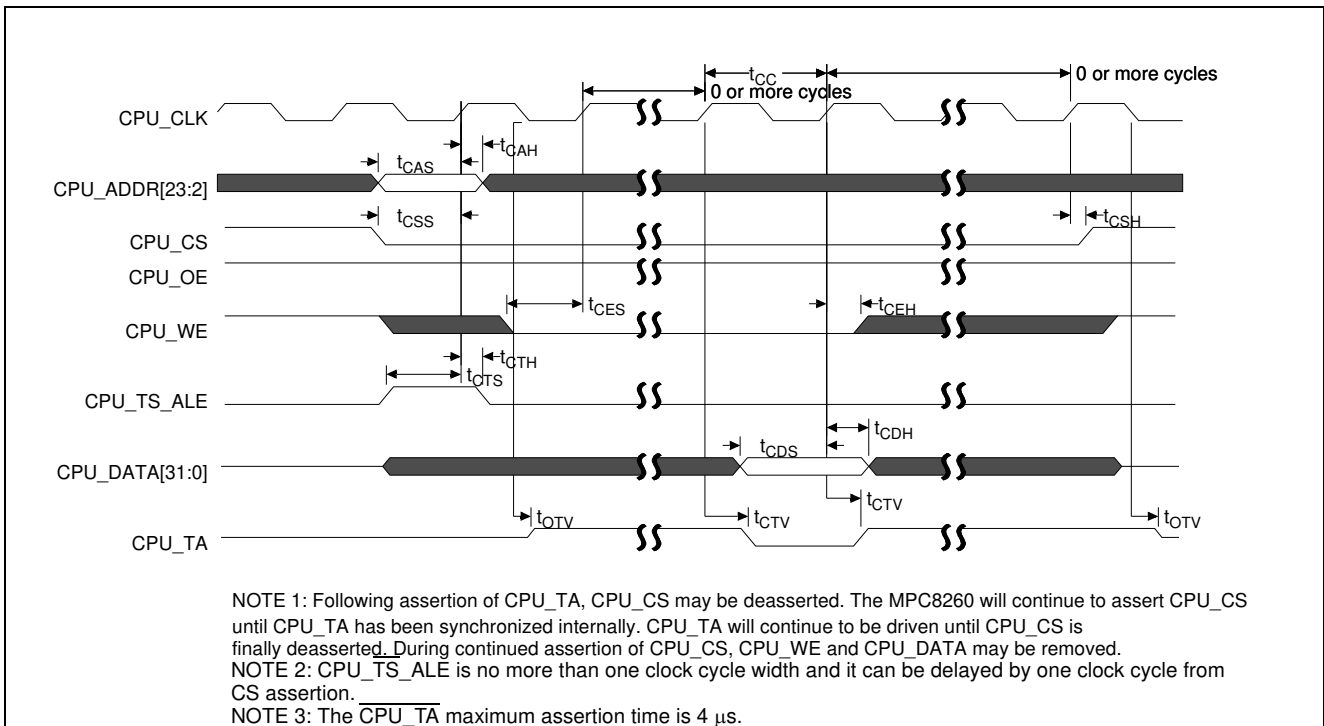


Figure 41 - CPU Write - MPC8260

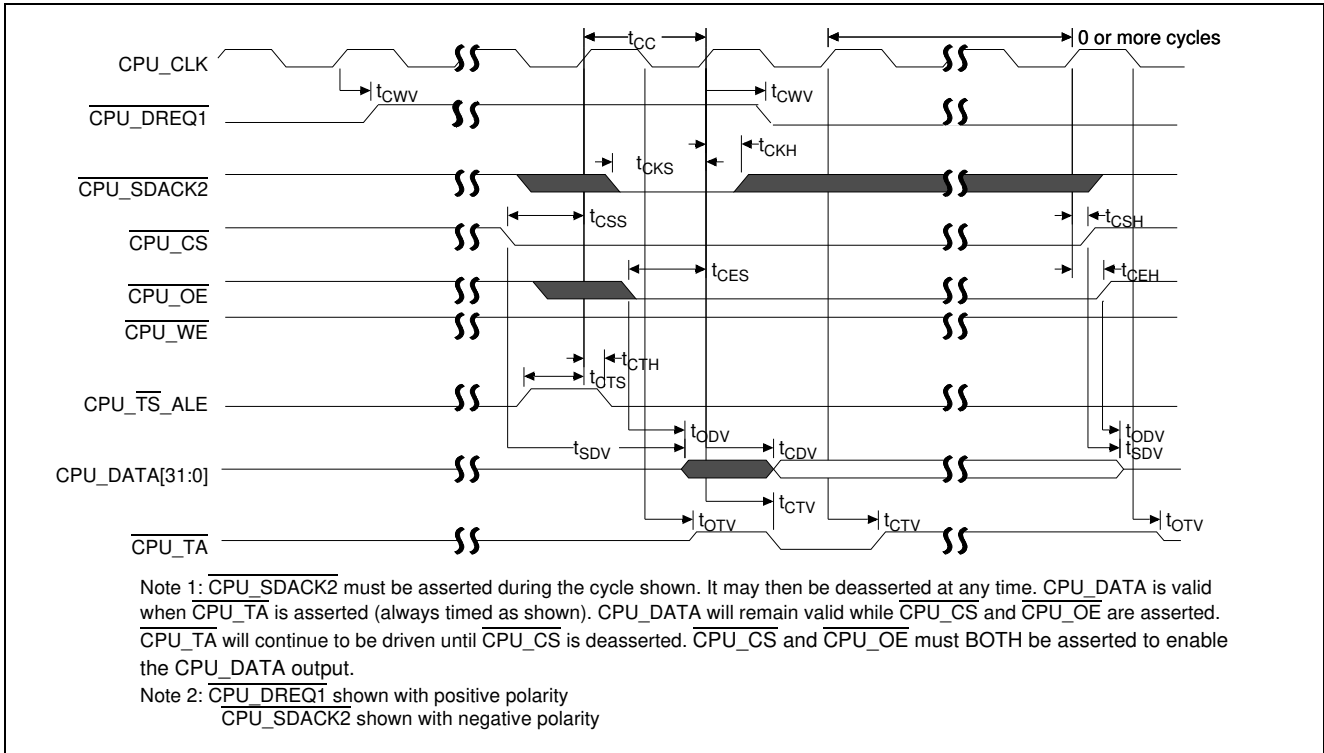


Figure 42 - CPU DMA Read - MPC8260

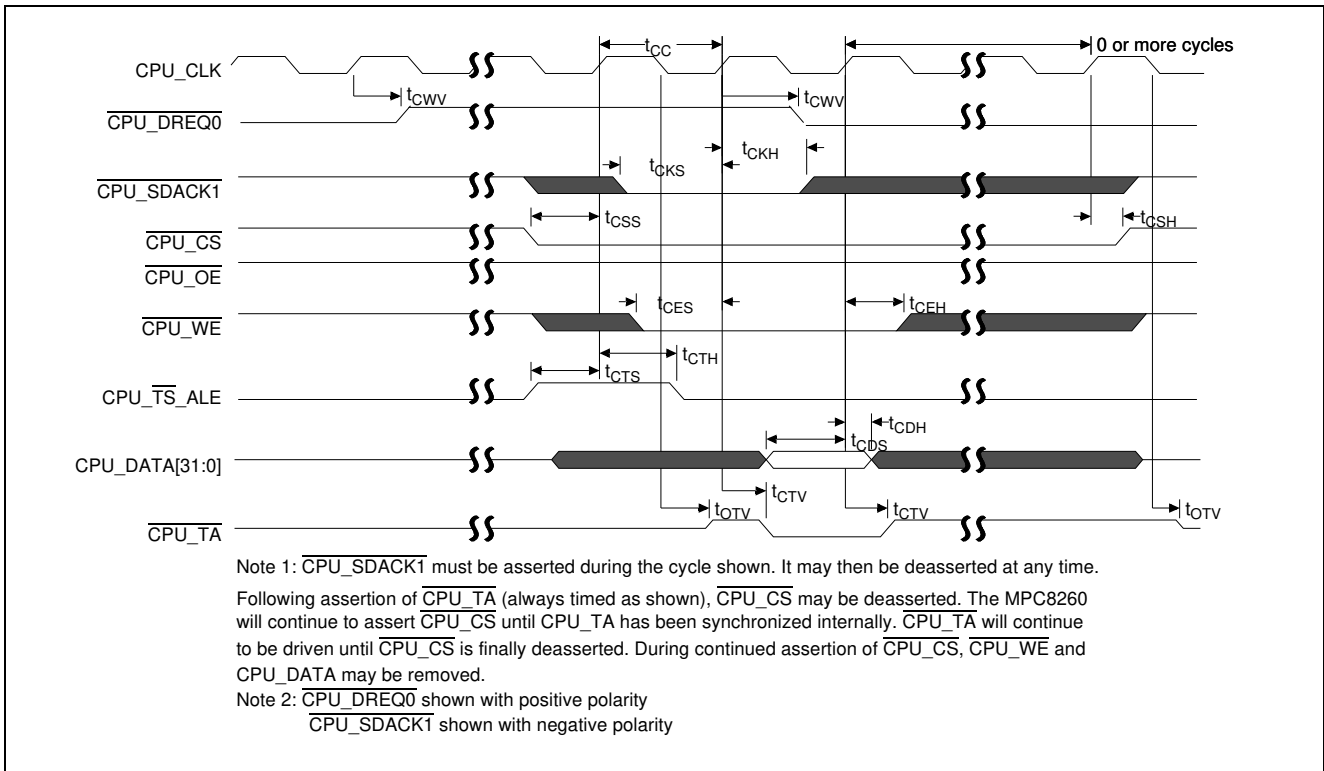


Figure 43 - CPU DMA Write - MPC8260

## 12.8 System Function Port

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
SYSTEM_CLK Frequency	CLK <sub>FR</sub>	-	100	-	MHz	Note 1, Note 2 and Note 5
SYSTEM_CLK accuracy (synchronous master mode)	CLK <sub>ACS</sub>	-	-	±30	ppm	Note 3
SYSTEM_CLK accuracy (synchronous slave mode and asynchronous mode)	CLK <sub>ACA</sub>	-	-	±200	ppm	Note 4

**Table 37 - System Clock Timing**

- Note 1: The system clock frequency stability affects the holdover-operating mode of the DPLL. Holdover Mode is typically used for a short duration while network synchronisation is temporarily disrupted. Drift on the system clock directly affects the Holdover Mode accuracy. Note that the absolute system clock accuracy does not affect the Holdover accuracy, only the change in the system clock (SYSTEM\_CLK) accuracy while in Holdover. For example, if the system clock oscillator has a temperature coefficient of 0.1 ppm/°C, a 10°C change in temperature while the DPLL is in will result in a frequency accuracy offset of 1 ppm. The intrinsic frequency accuracy of the DPLL Holdover Mode is 0.06 ppm, excluding the system clock drift.
- Note 2: The system clock frequency affects the operation of the DPLL in free-run mode. In this mode, the DPLL provides timing and synchronisation signals which are based on the frequency of the accuracy of the master clock (i.e., frequency of clock output equals 8.192 MHz ± SYSTEM\_CLK accuracy ± 0.005 ppm).
- Note 3: The absolute SYSTEM\_CLK accuracy must be controlled to ± 30 ppm in synchronous master mode to enable the internal DPLL to function correctly.
- Note 4: In asynchronous mode and in synchronous slave mode the DPLL is not used. Therefore the tolerance on SYSTEM\_CLK may be relaxed slightly.
- Note 5: The quality of SYSTEM\_CLK, or the oscillator that drives SYSTEM\_CLK directly impacts the adaptive clock recovery performance. See Section 6.3.

12.9 JTAG Interface Timing

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
JTAG_CLK period	$t_{JCP}$	40	100		ns	
JTAG_CLK clock pulse width	$t_{LOW}$ , $t_{HIGH}$	20	-	-	ns	
JTAG_CLK rise and fall time	$t_{JRF}$	0	-	3	ns	
JTAG_TRST setup time	$t_{RSTSU}$	10	-	-	ns	With respect to JTAG_CLK falling edge. Note 1
JTAG_TRST assert time	$t_{RST}$	10	-	-	ns	
Input data setup time	$t_{JSU}$	5	-	-	ns	Note 2
Input Data hold time	$t_{JH}$	15	-	-	ns	Note 2
JTAG_CLK to Output data valid	$t_{JDV}$	0	-	20	ns	Note 3
JTAG_CLK to Output data high impedance	$t_{JZ}$	0	-	20	ns	Note 3
JTAG_TMS, JTAG_TDI setup time	$t_{TPSU}$	5	-	-	ns	
JTAG_TMS, JTAG_TDI hold time	$t_{TPH}$	15	-	-	ns	
JTAG_TDO delay	$t_{TOPDV}$	0	-	15	ns	
JTAG_TDO delay to high impedance	$t_{TPZ}$	0	-	15	ns	

Table 38 - JTAG Interface Timing

Note 1:  $\overline{\text{JTAG\_TRST}}$  is an asynchronous signal. The setup time is for test purposes only.

Note 2: Non Test (other than JTAG\_TDI and JTAG\_TMS) signal input timing with respect to JTAG\_CLK.

Note 3: Non Test (other than JTAG\_TDO) signal output with respect to JTAG\_CLK.

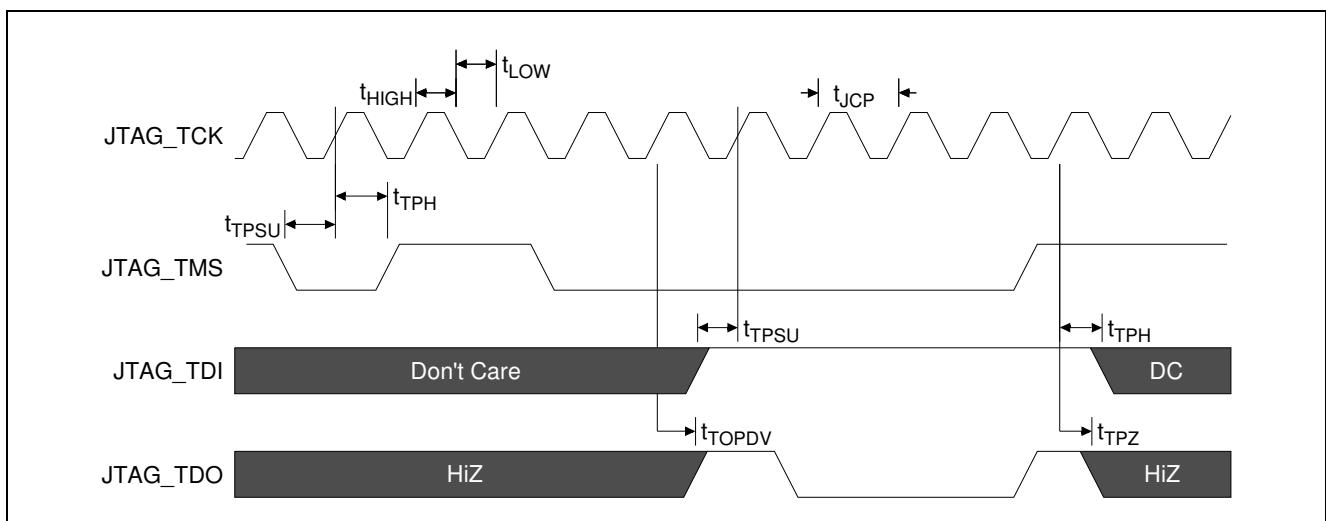


Figure 44 - JTAG Signal Timing

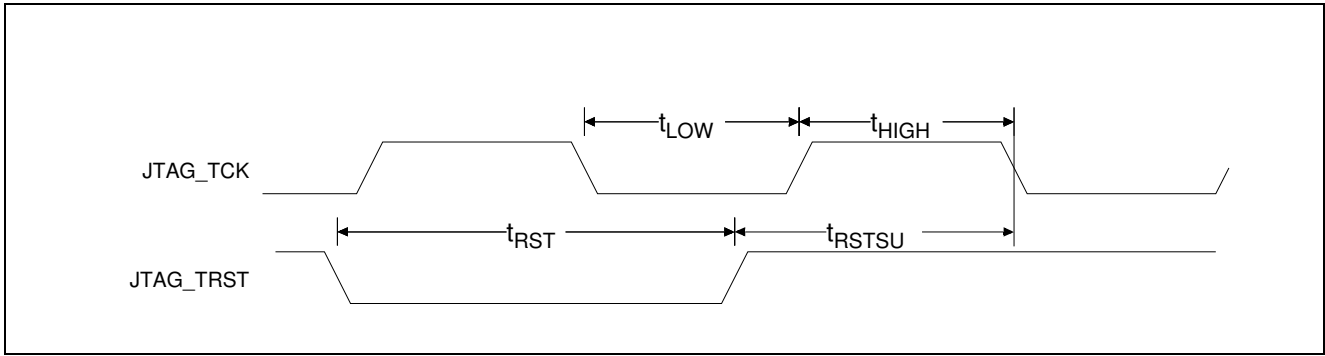


Figure 45 - JTAG Clock and Reset Timing

### 13.0 Power Characteristics

The following graph in Figure 46 illustrates typical power consumption figures for the ZL5011x family. Typical characteristics are at 1.8 V core, 3.3V I/O, 25°C and typical processing.

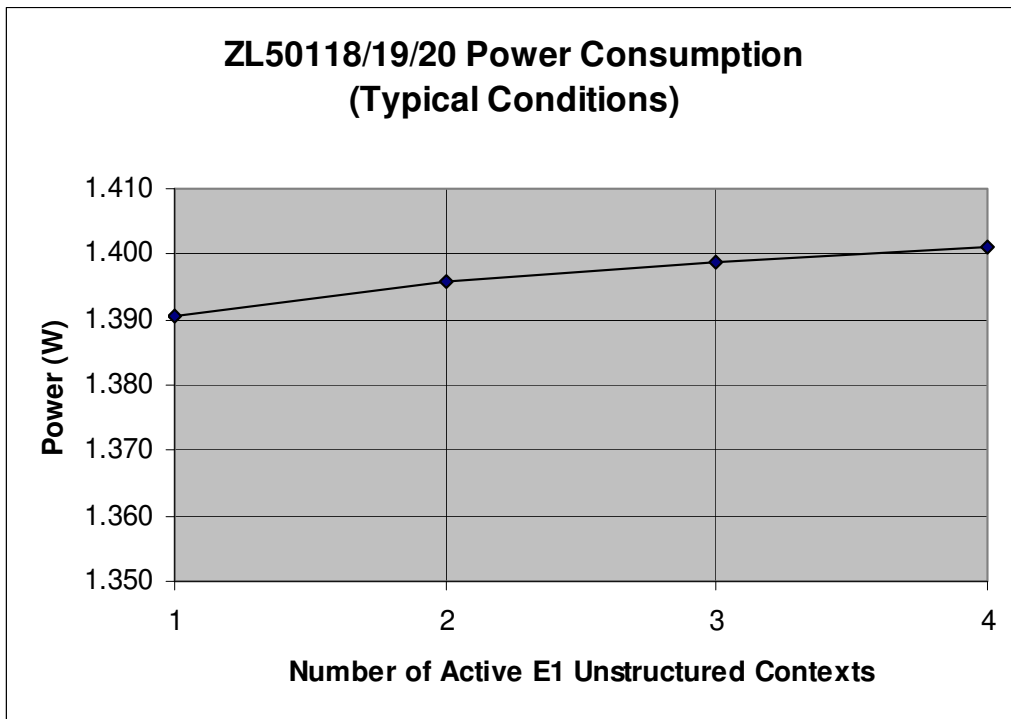


Figure 46 - ZL50115/16/17/18/19/20 Power Consumption Plot



## 14.0 Design and Layout Guidelines

This guide will provide information and guidance for PCB layouts when using the ZL5011x. Specific areas of guidance are:

- High Speed Clock and Data, Outputs and Inputs
- CPU\_TA Output

### 14.1 High Speed Clock & Data Interfaces

On the ZL5011x series of devices there are four high-speed data interfaces that need consideration when laying out a PCB to ensure correct termination of traces and the reduction of crosstalk noise. The interfaces being:

- GMAC Interfaces
- TDM Interface
- CPU Interface

It is recommended that the outputs are suitably terminated using a series termination through a resistor as close to the output pin as possible. The purpose of the series termination resistor is to reduce reflections on the line. The value of the series termination and the length of trace the output can drive will depend on the driver output impedance, the characteristic impedance of the PCB trace (recommend 50 ohm), the distributed trace capacitance and the load capacitance. As a general rule of thumb, if the trace length is less than 1/6th of the equivalent length of the rise and fall times, then a series termination may not be required.

$$\text{the equivalent length of rise time} = \text{rise time (ps)} / \text{delay (ps/mm)}$$

For example:

Typical FR4 board delay = 6.8 ps/mm

Typical rise/fall time for a ZL5011x output = 2.5 ns

$$\text{critical track length} = (1/6) \times (2500/6.8) = 61 \text{ mm}$$

Therefore tracks longer than 61 mm will require termination.

As a signal travels along a trace it creates a magnetic field, which induces noise voltages in adjacent traces, this is crosstalk. If the crosstalk is of sufficiently strong amplitude, false data can be induced in the trace and therefore it should be minimized in the layout. The voltage that the external fields cause is proportional to the strength of the field and the length of the trace exposed to the field. Therefore to minimize the effect of crosstalk some basic guidelines should be followed.

First, increase separation of sensitive signals, a rough rule of thumb is that doubling the separation reduces the coupling by a factor of four. Alternatively, shield the victim traces from the aggressor by either routing on another layer separated by a power plane (in a correctly decoupled design the power planes have the same AC potential) or by placing guard traces between the signals usually held ground potential.

Particular effort should be made to minimize crosstalk from ZL5011x outputs and ensuring fast rise time to these inputs.

In Summary:

- Place series termination resistors as close to the pins as possible
- minimize output capacitance
- Keep common interface traces close to the same length to avoid skew
- Protect input clocks and signals from crosstalk

#### **14.1.1 GMAC Interface - Special Considerations During Layout**

The GMII interface passes data to and from the ZL5011x with their related transmit and receive clocks. It is therefore recommended that the trace lengths for transmit related signals and their clock and the receive related signals and their clock are kept to the same length. By doing this the skew between individual signals and their related clock will be minimized.

#### **14.1.2 TDM Interface - Special Considerations During Layout**

Although the data rate of this interface is low the outputs edge speeds share the characteristics of the higher data rate outputs and therefore must be treated with the same care extended to the other interfaces with particular reference to the lower stream numbers which support the higher data rates. The TDM interface has numerous clocking schemes and as a result of this the input clock traces to the ZL5011x devices should be treated with care.

#### **14.1.3 Summary**

Particular effort should be made to minimize crosstalk from ZL5011x outputs and ensuring fast rise time to these inputs.

In Summary:

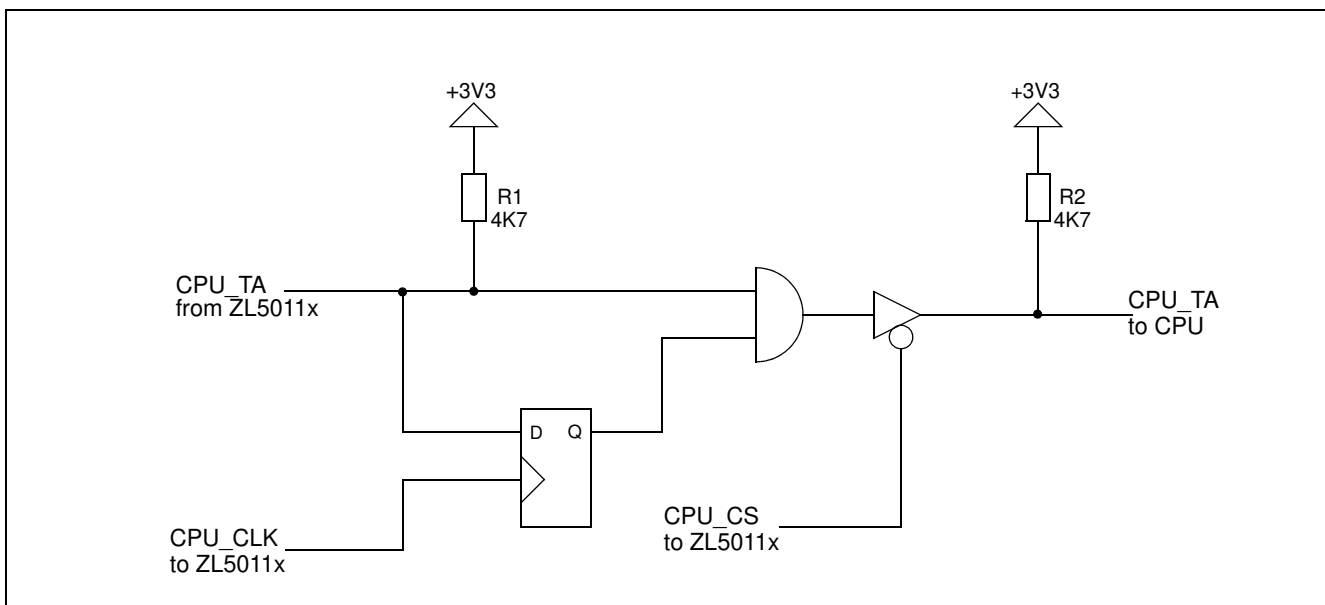
- Place series termination resistors as close to the pins as possible
- minimize output capacitance
- Keep common interface traces close to the same length to avoid skew
- Protect input clocks and signals from crosstalk

## 14.2 CPU TA Output

The CPU\_TA output signal from the ZL5011x is a critical handshake signal to the CPU that ensures the correct completion of a bus transaction between the two devices. As the signal is critical, it is recommended that the circuit shown in Figure 46 is implemented in systems operating above 40 MHz bus frequency to ensure robust operation under all conditions.

The following external logic is required to implement the circuit:

- 74LCX74 dual D-type flip-flop (one section of two)
- 74LCX08 quad AND gate (one section of four)
- 74LCX125 quad tri-state buffer (one section of four)
- 4K7 resistor x2



**Figure 47 - CPU\_TA Board Circuit**

The function of the circuit is to extend the TA signal, to ensure the CPU correctly registers it. Resistor R2 must be fitted to ensure correct operation of the TA input to the processor. It is recommended that the logic is fitted close to the ZL5011x and that the clock to the 74LCX74 is derived from the same clock source as that input to the ZL5011x.

---

## 15.0 Reference Documents

### 15.1 External Standards/Specifications

- IEEE Standard 1149.1-2001; Test Access Port and Boundary Scan Architecture
- IEEE Standard 802.3-2000; Local and Metropolitan Networks CSMA/CD Access Method and Physical Layer
- ECTF H.110 Revision 1.0; Hardware Compatibility Specification
- H-MVIP (GO-MVIP) Standard Release 1.1a; Multi-Vendor Integration Protocol
- MPC8260AEC/D Revision 0.7; Motorola MPC8260 Family Hardware Specification
- RFC 768; UDP
- RFC 791; IPv4
- RFC2460; IPv6
- RFC 1889; RTP
- RFC 2661; L2TP
- RFC 1213; MIB II
- RFC 1757; Remote Network Monitoring MIB (for SMIv1)
- RFC 2819; Remote Network Monitoring MIB (for SMIv2)
- RFC 2863; Interfaces Group MIB
- G.823; Control of Jitter/Wander with digital networks based on the 2.048 Mbps hierarchy
- G.824; Control of Jitter/Wander with digital networks based on the 1.544 Mbps hierarchy
- G.8261; Timing and Synchronization aspects in Packet Networks
- ANSI T1.101 Stratum 3/4
- Telcordia GR-1244-CORE Stratum 3/4/4e
- RFC5086; Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation - Service over Packet Switched Network (CESoPSN)
- RFC4553; Structure-Agnostic TDM over Packet (SAToP)
- ITU-T Y.1413 TDM-MPLS Network Interworking

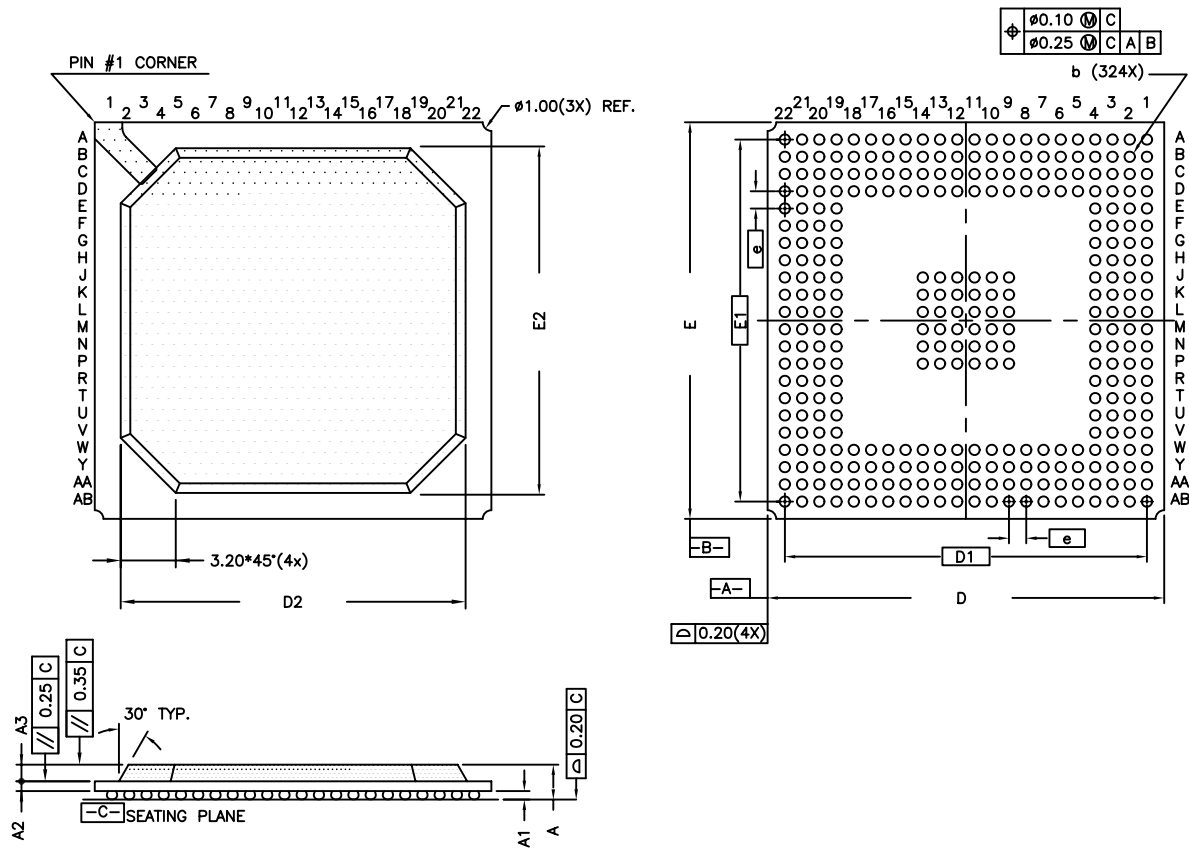
### 15.2 Zarlink Standards

- MSAN-126 Revision B, Issue 4; ST-BUS Generic Device Specification

## 16.0 Glossary

<b>API</b>	Application Program Interface
<b>ATM</b>	Asynchronous Transfer Mode
<b>CDP</b>	Context Descriptor Protocol (the protocol used by Zarlink's MT9088x family of TDM-Packet devices)
<b>CESoP</b>	Circuit Emulation Services over Packet
<b>CESoPSN</b>	Circuit Emulation Services over Packet Switched Networks
<b>CONTEXT</b>	A programmed connection of a number of TDM timeslots assembled into a unique packet stream.
<b>CPU</b>	Central Processing Unit
<b>DMA</b>	Direct Memory Access
<b>DPLL</b>	Digital Phase Locked Loop
<b>DSP</b>	Digital Signal Processor
<b>GMII</b>	Gigabit Media Independent Interface
<b>H.100/H.110</b>	High capacity TDM backplane standards
<b>H-MVIP</b>	High-performance Multi-Vendor Integration Protocol (a TDM bus standard)
<b>IETF</b>	Internet Engineering Task Force
<b>IA</b>	Implementation Agreement
<b>IP</b>	Internet Protocol (version 4, RFC 791, version 6, RFC 2460)
<b>JTAG</b>	Joint Test Algorithms Group (generally used to refer to a standard way of providing a board-level test facility)
<b>L2TP</b>	Layer 2 Tunneling Protocol (RFC 2661)
<b>LAN</b>	Local Area Network
<b>LIU</b>	Line Interface Unit
<b>MAC</b>	Media Access Control
<b>MEF</b>	Metro Ethernet Forum
<b>MFA</b>	MPLS and Frame Relay Alliance
<b>MII</b>	Media Independent Interface
<b>MIB</b>	Management Information Base
<b>MPLS</b>	Multi Protocol Label Switching
<b>MTIE</b>	Maximum Time Interval Error
<b>MVIP</b>	Multi-Vendor Integration Protocol (a TDM bus standard)
<b>PDH</b>	Plesiochronous Digital Hierarchy
<b>PLL</b>	Phase Locked Loop
<b>PRS</b>	Primary Reference Source
<b>PRX</b>	Packet Receive
<b>PSTN</b>	Public Switched Telephone Circuit

<b>PTX</b>	Packet Transmit
<b>PWE3</b>	Pseudo-Wire Emulation Edge to Edge (a working group of the IETF)
<b>QoS</b>	Quality of Service
<b>RTP</b>	Real Time Protocol (RFC 1889)
<b>PE</b>	Protocol Engine
<b>SAToP</b>	Structure-Agnostic TDM over Packet
<b>ST BUS</b>	Standard Telecom Bus, a standard interface for TDM data streams
<b>TDL</b>	Tapped Delay Line
<b>TDM</b>	Time Division Multiplexing
<b>UDP</b>	User Datagram Protocol (RFC 768)
<b>UI</b>	Unit Interval
<b>VLAN</b>	Virtual Local Area Network
<b>WFQ</b>	Weighted Fair Queuing



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	1.90	2.03	2.16
A1	0.40	0.50	0.60
A2	0.56 Ref.		
A3	0.97 Ref.		
b	0.50	0.60	0.70
D	22.80	23.00	23.20
D1	21.00 Ref.		
D2	20.00 Ref.		
E	22.80	23.00	23.20
E1	21.00 Ref.		
E2	20.00 Ref.		
e	1.00 Ref.		

Confirms to JEDEC MS-034  
iss. A

**NOTE:**

- ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
- DIMENSION "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER
- PRIMARY DATUM **-C-** AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
- ALL DIMENSIONS ARE IN MILLIMETERS.
- NOT TO SCALE.
- DETAILS OF A1 CORNER ARE OPTIONAL, AND MAY CONSIST OF INK DOT, LASER MARK OR METALISED MARKING, BUT MUST BE LOCATED WITHIN ZONE INDICATED.

© Zarlink Semiconductor 2006 All rights reserved.			
ISSUE	1	2	
ACN	CDCA	CDCA	
DATE	02 June 04	08 June 06	
APPRD.			



Previous package codes
------------------------

Package Code	GA
Package Outline for 324, PBGA (23x23x2.03)mm, 1.00 Pitch	
105072	



**For more information about all Zarlink products  
visit our Web Site at  
[www.zarlink.com](http://www.zarlink.com)**

Information relating to products and services furnished herein by Zarlink Semiconductor Inc. or its subsidiaries (collectively "Zarlink") is believed to be reliable. However, Zarlink assumes no liability for errors that may appear in this publication, or for liability otherwise arising from the application or use of any such information, product or service or for any infringement of patents or other intellectual property rights owned by third parties which may result from such application or use. Neither the supply of such information or purchase of product or service conveys any license, either express or implied, under patents or other intellectual property rights owned by Zarlink or licensed from third parties by Zarlink, whatsoever. Purchasers of products are also hereby notified that the use of product in certain ways or in combination with Zarlink, or non-Zarlink furnished goods or services may infringe patents or other intellectual property rights owned by Zarlink.

This publication is issued to provide information only and (unless agreed by Zarlink in writing) may not be used, applied or reproduced for any purpose nor form part of any order or contract nor to be regarded as a representation relating to the products or services concerned. The products, their specifications, services and other information appearing in this publication are subject to change by Zarlink without notice. No warranty or guarantee express or implied is made regarding the capability, performance or suitability of any product or service. Information concerning possible methods of use is provided as a guide only and does not constitute any guarantee that such methods of use will be satisfactory in a specific piece of equipment. It is the user's responsibility to fully determine the performance and suitability of any equipment using such information and to ensure that any publication or data used is up to date and has not been superseded. Manufacturing does not necessarily include testing of all functions or parameters. These products are not suitable for use in any medical products whose failure to perform may result in significant injury or death to the user. All products and materials are sold and services provided subject to Zarlink's conditions of sale which are available on request.

Purchase of Zarlink's I2C components conveys a license under the Philips I2C Patent rights to use these components in an I2C System, provided that the system conforms to the I2C Standard Specification as defined by Philips.

Zarlink, ZL, the Zarlink Semiconductor logo and the Legerity logo and combinations thereof, VoiceEdge, VoicePort, SLAC, ISLIC, ISLAC and VoicePath are trademarks of Zarlink Semiconductor Inc.

**TECHNICAL DOCUMENTATION - NOT FOR RESALE**

---