ZL50130

Ethernet Pseudo-Wires across a PSN

Data Sheet

January 2004

Applications

- Ethernet Pseudo-Wires across a Packet Switch Network

Ordering Information

ZL50130 PBGA

-40°C to +85°C

Features

Ethernet Pseudo-Wire Emulation Functions

Supports the following functions for Ethernet Pseudo-Wire emulation over the packet domain:

- Transports the complete Ethernet frame (less preamble and FCS) across the PSN
- Supports up to 127 point-to-point pseudo-wire links across the PSN
- VLAN priority field may be used to determine class of service on the PSN
- complies with the standards for Ethernet pseudo-wires proposed in the IETF's PWE3 working group

Network Interfaces

- 3 x 100 Mbit/s MII interfaces

System Interfaces

- Flexible 32-bit host CPU interface (Motorola PowerQUICC™ II compatible)
- Dual address DMA transfer of packets to or from the CPU
- On-chip packet memory for self-contained operation

Packet Processing Functions

- Flexible, multi-protocol packet encapsulation, with support for IPv4/6, MPLS, L2TP, PWE3
- Wire speed processing and forwarding of packets
- Packet sequencing and re-ordering where required
- Four classes of service with programmable priority mechanisms (WFQ and SP)
- Flexible classification of incoming packets at layers 2, 3, 4 and 5

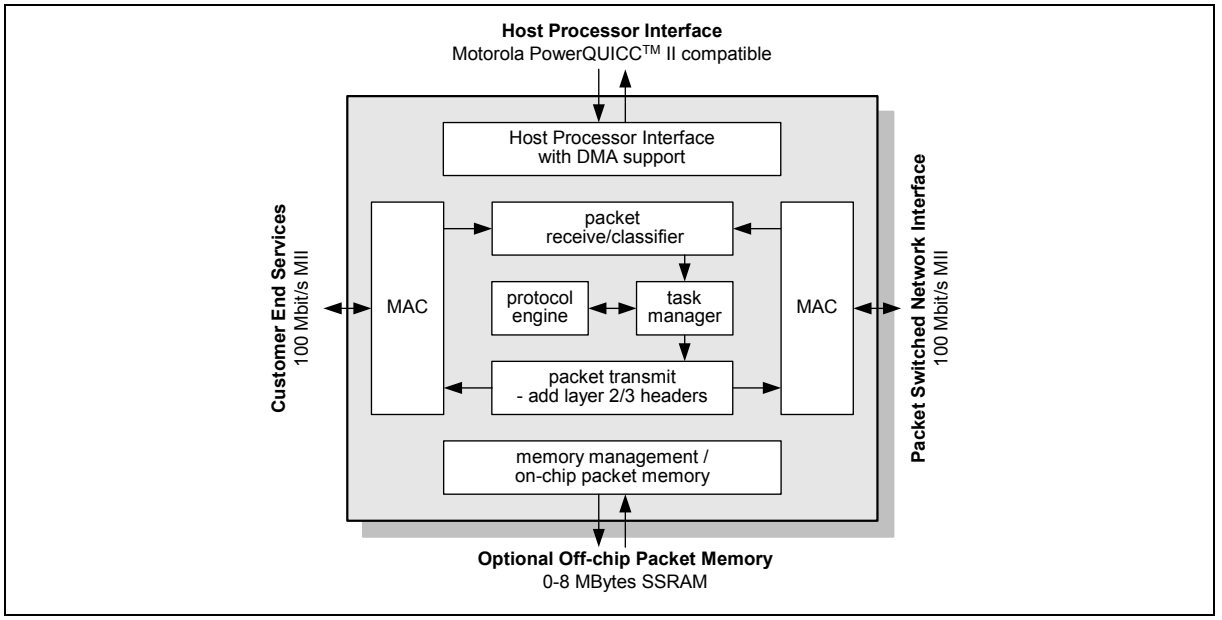


Figure 1 - High Level Overview

Description

The ZL50130 is part of a range of highly functional bridging devices. It provides the capability to extend a Local Area Network based on Ethernet across a service provider's packet switched network. This allows a company with multiple sites to manage its network as though it was a single LAN.

In conjunction with an Ethernet aggregation network, the ZL50130 provides the dataplane requirements of the inter-working function between the customer end services and the provider's packet switched network (see Figure 2). It can support wire-speed processing of up to 100 Mbit/s of traffic in each direction, and provides up to 127 separate pseudo-wire connections across the PSN. Packets arriving from a single customer end service may all be directed onto a single pseudo-wire, or split across multiple connections based on source and destination addresses. This is useful in the case where the customer is using the Ethernet pseudo-wire service to connect multiple sites.

On packet egress the device includes four different classes of service, allowing priority treatment of customer traffic, depending on the service level agreement between the customer and provider. The user priority field in the packet's VLAN tag (if any) may be used to determine the appropriate class of service to be used on the PSN.

Packets received from the Ethernet interfaces are parsed to determine the egress destination, and are appropriately queued to the customer end service, passed up to the host processor, or sent back toward the packet interface. Again there are four different classes of service to allow differentiation between customers with different service level agreements, or based on the use of VLAN tag priority.

The ZL50130 includes sufficient on-chip memory to allow completely self-contained operation, reducing system costs and simplifying the design. For applications that do require more memory (e.g. where the network has a very high packet delay variation), the device supports up to 8 Mbytes of external synchronous ZBT SRAM.

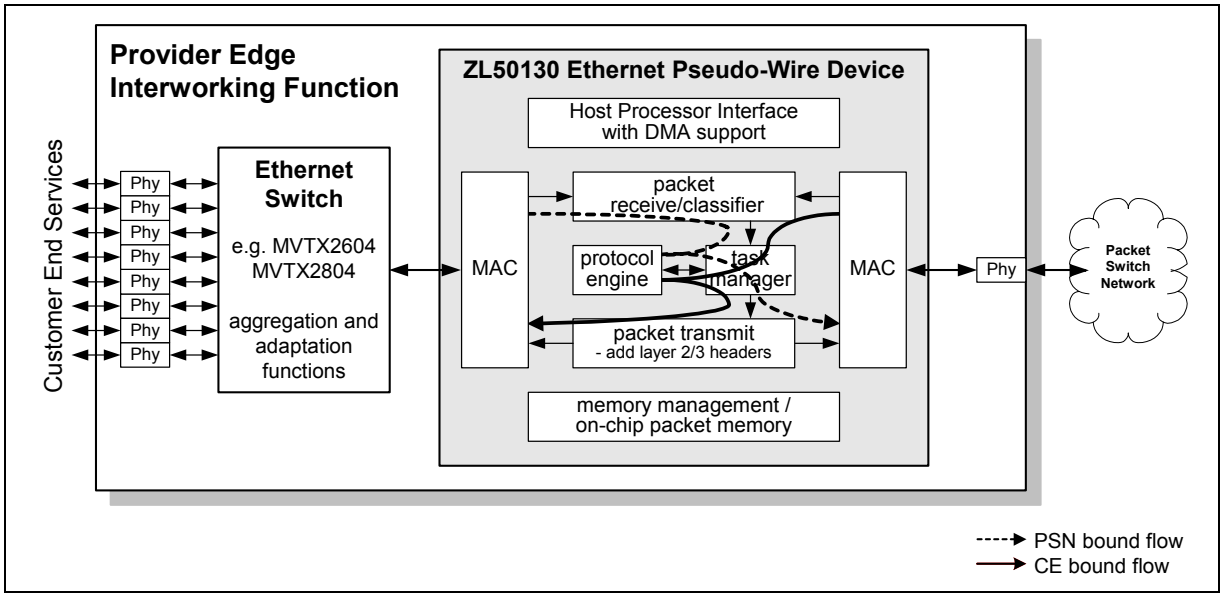


Figure 2 - Provider Edge Inter-working Function using the ZL50130

Table of Contents

Table of Contents

List of Figures

List of Tables

1.0 Introduction

1.1 Overview

The ZL50130 provides the data-plane processing to enable layer 2 Ethernet service to be extended over a packet switched network, such as an IP or MPLS system. The device encapsulates the Ethernet frames into IP or MPLS packets, and forwards them into the packet network for re-construction at the far end. This has a number of applications, including layer 2 VPNs (Virtual Private Networks).

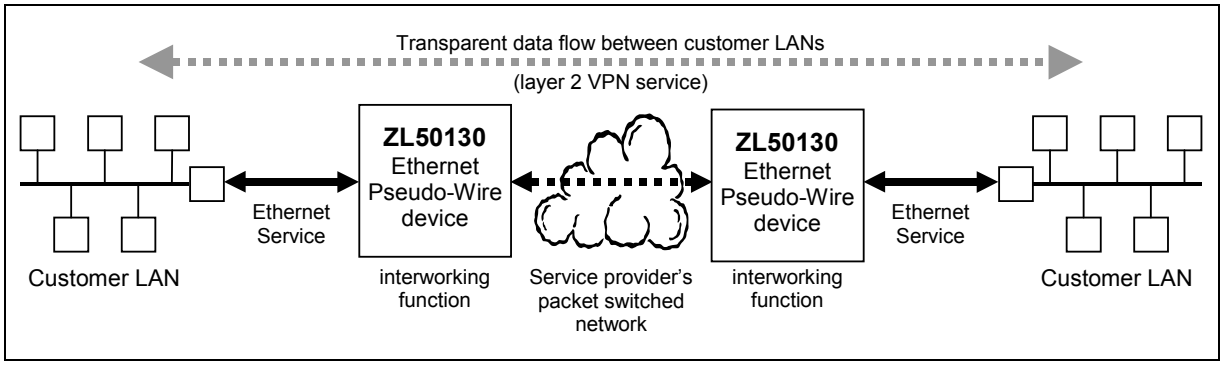


Figure 3 - ZL50130 Operation

The ZL50130 is capable of wire speed processing and forwarding of packets, and includes support for the "Martini-style" layer 2 pseudo-wire protocols currently in development by the IETF's PWE3 (Pseudo-Wire Edge-to-Edge Emulation) working group (draft-ietf-pwe3-ethernet-encap).

1.2 Basic Operation

A diagram of the ZL50130 device is given in Figure 4, which shows the major data flows between functional components.

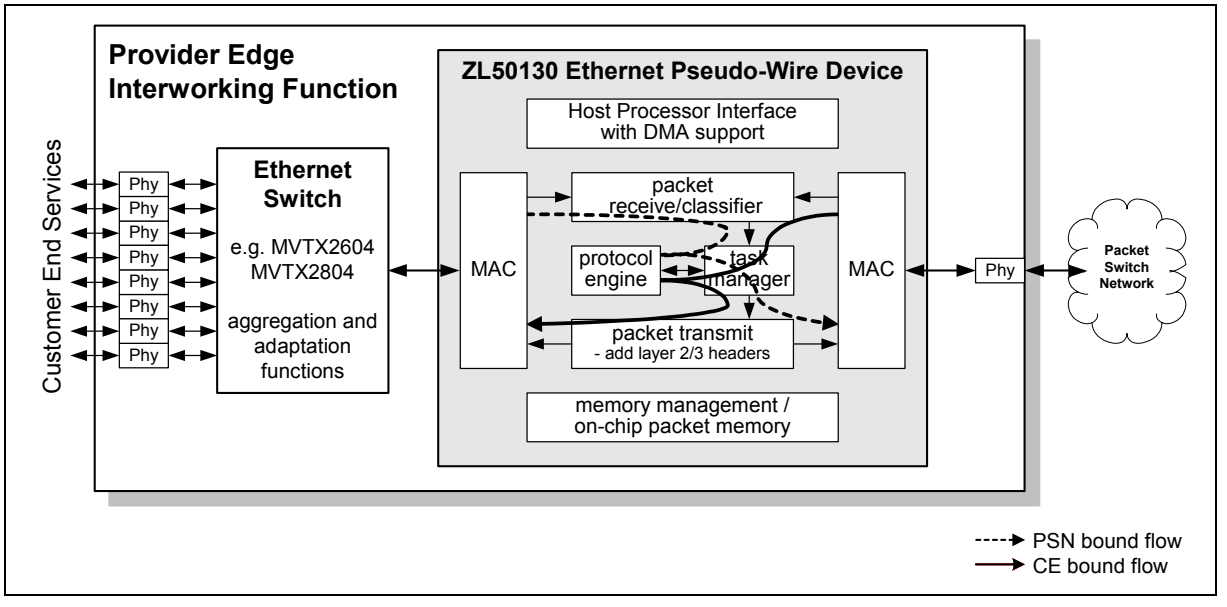


Figure 4 - ZL50130 Data Flows

1.2.1 PSN-Bound Flow

The Ethernet switch device aggregates Ethernet frames received from each customer into a single Ethernet connection. Packets are forwarded on this connection to the ZL50130, and received by its customer-facing MAC interface. Valid packets are passed to the Packet Classifier to determine the destination.

The Protocol Engine handles the data-plane requirements of the main higher level protocols (layers 4 and 5) used in typical applications of the ZL50130. These include the Ethernet pseudo-wire control word (basically a 16-bit sequence number), L2TPv3 connection ID, L2TP version 2 and UDP. 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 to the CPU. The header 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.

Packets ready for transmission are queued to the switch fabric interface by the Packet Transmit block. 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, or the MPLS tunnel and the VC labels. Finally, packets are sent out to the packet switched network by the PSN-facing MAC.

1.2.2 CE-Bound Flow

The flow in the reverse direction is essentially similar to the PSN-bound flow. Packets from the PSN are received by its PSN-facing MAC interface. Valid packets are passed to the Packet Classifier to determine the destination. Once this has been determined, the packets are passed to the Packet Transmit block for forwarding onto the customer. This time, the Packet Transmit block strips the tunnel header appended on original transmission into the PSN. The packets are then queued for transmission by the customer-facing MAC.

1.2.3 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 out-of-band 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 DMA transfers of packets to and from the CPU memory, using the host's own DMA controller.

1.2.4 External Memory Requirement

The ZL50130 includes a large amount of on-chip memory, such that for most applications, external memory will not be required. However, for some applications there may be a requirement for external memory. Therefore, the device allows the external connection of up to 8 Mbytes of synchronous ZBT SRAM.

2.0 Functional Block Descriptions

2.1 Task Manager

Conceptually, the task manager performs the function of a router in the centre of the chip, directing packets to the appropriate processing blocks. The architecture is based on the task-oriented approach derived from computer science, in which each functional block is considered a service provider or “subroutine". As a packet is processed in the chip, it receives a specific service from a block, returns the flow of control to the task manager and is then forwarded to the next block for service. The process is carried on until it reaches the egress port, as shown in Figure 5. The solid arrows illustrate the actual flow of control, while the dotted lines represent the equivalent point-to-point path.

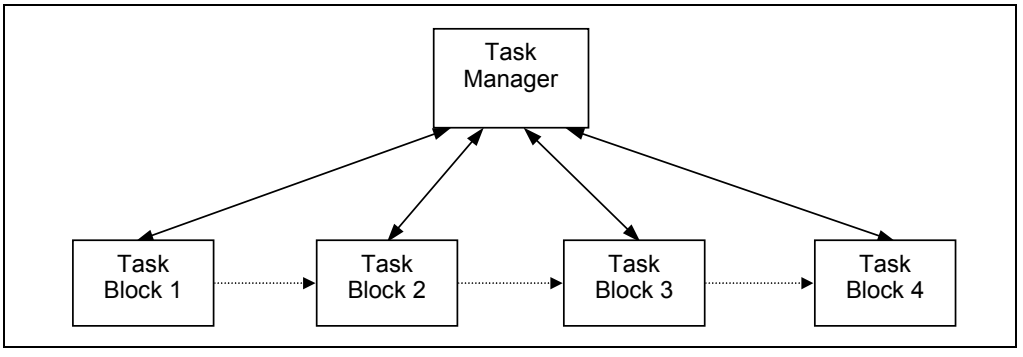


Figure 5 - Task Manager Routing Concept

The main function of the task manager is to dispatch task messages to the task blocks. Each task block interfaces only with the task manager, not with other task blocks. The task messages passed between blocks contain pointers 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.

Features include:

- Flexible routing of packets through the device
- Common interface to all functional blocks
- Communication of key parameters between blocks

2.2 Protocol Engine

The Protocol Engine handles the data-plane requirements of the upper level protocols at layers 4 and 5. It has been designed to handle the requirements of the specific protocols expected to be used in typical applications: UDP, L2TP versions 2 and 3, and Ethernet pseudo-wire. However, it is not exclusively limited to these protocols, since it works by providing a number of dynamic fields that can be updated as required. Therefore other protocol headers containing the same types of dynamic fields can also be constructed.

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. This header 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.

Features include:

- Up to 24 bytes of higher layer protocol headers (layers 2 to 5)
- Specific support for the following protocols:
  - UDP (RFC 768)
  - L2TP versions 2 and 3 (RFC 2661, draft-ietf-l2tpext-l2tp-base-02)
  - Ethernet PW (draft-ietf-pwe3-ethernet-encap)
- Support for the following dynamic fields:
  - 16 bit checksum over header and data
  - 16 bit datagram length field
  - 8 and 16 bit sequence numbers

2.3 Packet Transmit

The Packet Transmit block serves two main functions. Firstly, it directs the packets to the appropriate Ethernet MAC. Four separate queues are directed at each MAC, allowing different classes of service to be established. For instance, packets from the host can be given lower priority service than normal transmission packets. The queues can be managed either on a strict priority basis, or using Weighted Fair Queuing (WFQ). There is also a limit on the maximum number of packets in an individual queue. Beyond the limit packets are dropped, avoiding a large build-up in packet delay variation, and preventing the memory becoming full and thereby "crashing" the device.

Secondly, the Packet Transmit block adds a header to the packet before it is sent out for transmission. Typically this contains the data-link and network layer headers (layers 2 and 3), such as Ethernet, IP and MPLS. While the main contents of this header are static and programmed by the user in a block of memory reserved for each context, certain fields can be dynamically adjusted by the block. These are the Ethernet, IPv4 and IPv6 length fields and the IPv4 identification field (effectively a sequence number). Once the packet header has been appended, any packets that are still smaller than the minimum Ethernet packet size (64 bytes) are padded.

Features include:

- Four separate queues to each Ethernet MAC, with different classes of service
- Queue disciplines on packet interface are:
  - Weighted Fair Queuing with programmable weights
  - Strict Priority
- Programmable drop threshold when queues get too big
- Adds up to 64 bytes of layer 2 and layer 3 headers
- Automatically adjusts Ethernet length, IP length and IP identification fields
- Pads small packets to meet the 64 byte minimum Ethernet packet size
- Packets can be directed to any of the four Ethernet MACs

2.4 Packet Receive

The Packet Receive block handles two main functions: writing the packet into memory, and identifying the packet received. Packets are written into memory as they arrive at the device. A small buffer is used to cope with simultaneous requests for memory access. The device is capable of accepting packets on 3 ports in MII mode, either ports 0, 1 and 2, or ports 0, 1 and 3.

2.4.1 Packet Classification

The ZL50130 contains an extremely flexible packet classifier, capable of operating on layers 2 to 5, and identifying 272 separate flows. For instance, it can identify a separate data and control flow for each context, and a number of other separate flows for sending to the CPU. In addition, the classifier can quickly identify certain specific types of control traffic intended for the CPU, such as ARP, RARP and multicast messages.

The classifier is designed to handle all the protocols likely to occur in networks using the standards developed by the IETF's PWE3, either alone or in combination. These include Ethernet (with VLAN and SNAP), IPv4, IPv6, MPLS, UDP, L2TP versions 2 and 3, and the Ethernet pseudo-wire control word. Typical protocol stacks which are expected to be used and can be handled by the ZL50130 include (but are not limited to) the following:

Ethernet	Ethernet	Ethernet	Ethernet
IPv4 / IPv6	IPv4 / IPv6	MPLS (tunnel label)	Stacked VLAN (tunnel tag)
L2TPv3	UDP	MPLS (VC label)	Stacked VLAN (VC tag)
	L2TPv2		
Ethernet PW	Ethernet PW	Ethernet PW	Ethernet PW

The classifier is also used to detect traffic directed to the host CPU.

Features include:

- Automatic forwarding of the following control traffic types to the CPU:
  - 802.1 control packets
  - Ethernet broadcast packets
  - IP multicast packets
  - ARP packets
  - RARP packets
- Four separate traffic classes
- Traffic class identified on any combination fields within the first 64 bytes
- 272 traffic flows (e.g. 128 data, 128 control, 16 host traffic and loopback)
- Flow identification based on up to 96 bits, extracted from any field in the first 96 bytes
- Mis-connection check, based on up to 64 bits, extracted from any field in the first 96 bytes
- Host traffic can be directed to its queue
- IPv4 checksum verification
- Support for IP and UDP MIBS in combination with the CPU

2.4.2 Classifier Operation

The classifier works in four steps. Firstly, the classifier checks the packet header to see if it is one of a number of fixed traffic types. These include 802.1 control packets, Ethernet broadcast packets, IP multicast packets, ARP and RARP packets. These types of packets are automatically forwarded to the CPU at high priority.

Next the traffic is sorted into one of four pre-determined traffic classes. This is done by a comparison across the first 64 bytes of the packet. This will generally check fields such as Ethertype and the IP protocol field. In addition, the Ethernet and IP destination address fields can be checked, to ensure that the packet is intended for this device.

Once the class has been determined, a template is applied, extracting up to 96 bits from any field in the first 96 bytes. These are used to determine the individual flow. For example, this could be used to check the cookie value in the L2TPv3 header. The checksum fields can also be verified now, since the protocol stack in use has been determined.

When the flow has been identified, up to 64 further bits may be compared to a pre-programmed value as a mis-connection check. For example, the SSRC field in the RTP header could be checked, or the cookie value in the L2TPv3 header. It could also be used to check Ethernet or IP source addresses, to check the packet came from the expected source. These 64 bits may again be extracted from any field in the first 96 bytes. The use of the mis-connection check helps to protect against denial of service attacks, since the cookie or SSRC values are usually hard to guess.

At any stage, a failure to match results in the packet being directed to the CPU queue. This enables the host to view the packet and take appropriate action.

2.5 Ethernet MAC

The ZL50130 device contains four separate, IEEE standard 802.3 compliant, 100 Mbit/s Ethernet MACs. Each MAC is connected to a Physical Layer (PHY) device via a Media Independent Interface (MII).

The MAC is responsible for data encapsulation/decapsulation. This includes frame alignment and synchronization, and detection of physical medium transmission errors. The MAC is capable of both full and half-duplex operation. In half-duplex mode it manages the collision avoidance and contention resolution process. In the event of a collision, the MAC will back off and attempt to re-send the packet up to 16 times.

Packets for transmission are forwarded to the MAC by the Packet Formatter block. The MAC appends the frame check sequence, and generates the preamble and start of frame delimiter before transmitting out of the MII port.

During packet reception, the MAC receive section verifies that the frame check sequence is correct, and that the packet is a valid length. Packets with an invalid frame check sequence, and data packets longer than a pre-programmed threshold and shorter than 64 bytes are dropped.

For Ethernet pseudo-wire operation, the thresholds on the customer-facing MACs are usually set shorter than on the PSN-facing MAC. This is to avoid creating over-sized packets when the additional tunnel headers are added to the packet.

The MAC also checks the destination field to determine if the packet is intended for the device. If the packet is accepted, it is forwarded on for packet classification, and to be entered into the appropriate destination queue. Illegal packets, or packets intended for a different destination are discarded.

The MAC also collects statistics on the different types of packets transmitted and received on the Ethernet. The statistics collected are sufficient to enable the CPU to support the Interfaces sections of some common MIBs.

Features include:

- IEEE 802.3 compliant operation at 100 Mbit/s
- Industry-standard MII interface to the physical layer devices
- Full and half-duplex operation
- Generates preamble, start-of-frame delimiter and frame check sequence
- Collision avoidance and contention resolution in half-duplex mode
- Verifies frame check sequence and frame length, discarding frames that contain errors
- Statistics collection for common MIB support:
  - 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

2.6 Memory Management Unit

The Memory Management Unit handles all access to the on- and off-chip packet memory, arbitrating between the different modules requiring access. Efficient use of memory is maintained by allocating memory in small blocks or “granules”.

The ZL50130 includes a large amount of on-chip memory, such that for most applications, external memory will not be required. However, for some applications there may be a requirement for external memory. Therefore the device allows the connection of up to 8Mbytes of synchronous ZBT SRAM.

Features include:

- On-chip packet memory for self-contained operation
- Up to 8 Mbytes of off-chip packet memory
- Interfaces to bandwidth efficient ZBT SRAM devices
- Operates at 100 MHz
- 64 bit wide data path
- Supports one memory bank consisting of up to 2x32 bit devices or 1x64 bit device.

2.7 Host Interface

The Host Interface is directly compatible with the Motorola PowerQUICC™ II microprocessor family. It provides the host CPU with read and write access to registers, internal memory blocks, and the main on- and off-chip data memory. The device supports dual address DMA transfers of packets to and from the CPU memory, using the host's own DMA controller.

Features include:

- Interfaces directly to PowerQUICC™ II (MPC8260) microprocessors using the GPCM.
- 32 bit wide data bus
- Allows target access to all on-chip registers and memory, and to external packet memory
- DMA support, for transfer of packets to and from the host CPU
- Flexible interrupt controller

Table illustrates the maximum bandwidths achievable by an external DMA master.

DMA Path	Packet Size	Max Bandwidth Mbps[1]
ZL50130 to CPU only	>1000 bytes	50
ZL50130 to CPU only	60 bytes	6.7
CPU to ZL50130 only	>1000 bytes	60
CPU to ZL50130 only	60 bytes	43
Combined[2]	>1000 bytes	58 (29 each way)
Combined[2]	60 bytes	11 (5.5 each way)

Table 1 - DMA Maximum Bandwidths

Note 1: Maximum bandwidths are the maximum the ZL50130 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.

2.8 JTAG Interface and Board Level Test Features.

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

3.0 External Interface Description

The following key applies to all tables:

- I Input
- O Output
- D Internal 100 kΩ pull-down resistor present
- U Internal 100 kΩ pull-up resistor present
- T Tristate Output

3.1 Packet Interfaces

The ZL50130 the packet interface is capable of 3 MII interfaces.

Data for packet switching is based on Specification IEEE Std. 802.3 - 2000. 3 ports can be used as 100 Mbit/s MII interfaces, either Ports 0, 1 and 2 or Ports 0, 1 and 3.

Note: Port 2 and Port 3 can not be used to receive data simultaneously, they are mutually exclusive. They may both be used for packet transmission if required.

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

Signal	I/O	Package Balls	Description
M_MDC	O	H23	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	G26	MII management data I/O. Common for all four MII ports at up to 2.5 MHz. It is bi-directional between the ZL50130 and the Ethernet station management entity. Data is passed synchronously with respect to M_MDC.

Table 2 - MII Management Interface Package Ball Definition

MII Port 0			
Signal	**I/O**	**Package Balls**	**Description**
M0_LINKUP_LED	O	G24	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	AC26	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_REFCLK	I D	AA24	**NOTE:** In **MII** mode this pin must be driven with the same clock as M0_RXCLK.
M0_RXCLK	I U	AB22	MII - M0_RXCLK. Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M0_COL	I D	Y25	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.

Table 3 - MII Port 0 Interface Package Ball Definition

MII Port 0			
Signal	**I/O**	**Package Balls**	**Description**
M0_RXD[3:0]	I U	[3] W26 [2] U22 [1] Y26 [0] AA26	Receive Data. Clocked on rising edge of M0_RXCLK.
M0_RXDV	I D	V25	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.
M0_RXER	I D	V26	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.
M0_CRS	I D	U25	MII - M0_CRS Carrier Sense. This asynchronous signal is asserted when either the transmission or reception device is non-idle. It is active high.
M0_TXCLK	I U	U24	MII Transmit Clock Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M0_TXD[3:0]	O	[3] AA23 [2] W21 [1] Y22 [0] AA22	Transmit Data. Clocked on rising edge of M0_TXCLK (MII).
M0_TXEN	O	V23	MII - M0_TXEN Transmit Enable. Asserted when the MAC has data to transmit, synchronously to M0_TXCLK with the first preamble of the packet to be sent. Remains asserted until the end of the packet transmission. Active high.

Table 3 - MII Port 0 Interface Package Ball Definition

MII Port 0			
Signal	**I/O**	**Package Balls**	**Description**
M0_TXER	O	V22	MII - M0_TXER Transmit Error. Transmitted synchronously with respect to M0_TXCLK, and active high. When asserted (with M0_TXEN also asserted) the ZL50130 will transmit a non-valid symbol, somewhere in the transmitted frame.

Table 3 - MII Port 0 Interface Package Ball Definition

MII Port 1			
Signal	**I/O**	**Package Balls**	**Description**
M1_LINKUP_LED	O	G23	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	AB25	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_REFCLK	I D	M22	**NOTE:** In **MII** mode this pin must be driven with the same clock as M1_RXCLK.
M1_RXCLK	I U	M23	MII - M1_RXCLK. Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M1_COL	I D	R25	MII - M1_COL. 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] N25 [2] N24 [1] R26 [0] T26	Receive Data. Clocked on rising edge of M1_RXCLK (MII).

Table 4 - MII Port 1 Interface Package Ball Definition

MII Port 1			
Signal	**I/O**	**Package Balls**	**Description**
M1_RXDV	I D	M26	MII - M1_RXDV 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	L21	MII - M1_RXER 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	L23	MII - M1_CRS 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	L22	MII Transmit Clock Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M1_TXD[3:0]	O	[3] R22 [2] P21 [1] T22 [0] R21	Transmit Data. Clocked on rising edge of M1_TXCLK (MII).
M1_TXEN	O	P23	MII - M1_TXEN 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 4 - MII Port 1 Interface Package Ball Definition

MII Port 1			
Signal	**I/O**	**Package Balls**	**Description**
M1_TXER	O	N23	MII - M1_TXER Transmit Error. Transmitted synchronously with respect to M1_TXCLK, and active high. When asserted (with M1_TXEN also asserted) the ZL50130 will transmit a non-valid symbol, somewhere in the transmitted frame.

Table 4 - MII Port 1 Interface Package Ball Definition

MII Port 2 Note: This port must not be used to receive data at the same time as port 3, they are mutually exclusive.			
Signal	**I/O**	**Package Balls**	**Description**
M2_LINKUP_LED	O	F26	LED drive for MAC 2 to indicate port is linked up. Logic 0 output = LED on Logic 1 output = LED off
M2_ACTIVE_LED	O	AB24	LED drive for MAC 2 to indicate port is transmitting or receiving packet data. Logic 0 output = LED on Logic 1 output = LED off
M2_RXCLK	I U	AA19	MII Receive Clock. Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M2_COL	I D	AE26	Collision Detection. This signal is independent of M2_TXCLK and M2_RXCLK, and is asserted when a collision is detected on an attempted transmission. It is active high, and only specified for half-duplex operation.
M2_RXD[3:0]	I U	[3] AD25 [1] AB21 [2] AC23 [0] AD24	Receive Data. Clocked on rising edge of M2_RXCLK.
M2_RXDV	I D	AA20	Receive Data Valid. Active high. This signal is clocked on the rising edge of M2_RXCLK. It is asserted when valid data is on the M2_RXD bus.

Table 5 - MII Port 2 Interface Package Ball Definition

MII Port 2 Note: This port must not be used to receive data at the same time as port 3, they are mutually exclusive.			
Signal	**I/O**	**Package Balls**	**Description**
M2_RXER	I D	AC24	Receive Error. Active high signal indicating an error has been detected. Normally valid when M2_RXDV is asserted. Can be used in conjunction with M2_RXD when M2_RXDV signal is de-asserted to indicate a False Carrier.
M2_CRS	I D	AC25	Carrier Sense. This asynchronous signal is asserted when either the transmission or reception device is non-idle. It is active high.
M2_TXCLK	I U	AD26	MII Transmit Clock Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M2_TXD[3:0]	O	[3] AE25 [1] AC21 [2] AD23 [0] AE24	Transmit Data. Clocked on rising edge of M2_TXCLK
M2_TXEN	O	AC22	Transmit Enable. Asserted when the MAC has data to transmit, synchronously to M2_TXCLK with the first pre-amble of the packet to be sent. Remains asserted until the end of the packet transmission. Active high.
M2_TXER	O	AB20	Transmit Error. Transmitted synchronously with respect to M2_TXCLK, and active high. When asserted (with M2_TXEN also asserted) the ZL50130 will transmit a non-valid symbol, somewhere in the transmitted frame.

Table 5 - MII Port 2 Interface Package Ball Definition

MII Port 3 Note: This port must not be used to receive data at the same time as port 2, they are mutually exclusive.			
Signal	**I/O**	**Package Balls**	**Description**
M3_LINKUP_LED	O	AB23	LED drive for MAC 3 to indicate port is linked up. Logic 0 output = LED on Logic 1 output = LED off
M3_ACTIVE_LED	O	AB26	LED drive for MAC 3 to indicate port is transmitting or receiving packet data. Logic 0 output = LED on Logic 1 output = LED off
M3_RXCLK	I U	K26	MII Receive Clock. Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M3_COL	I D	J26	Collision Detection. This signal is independent of M3_TXCLK and M3_RXCLK, and is asserted when a collision is detected on an attempted transmission. It is active high, and only specified for half-duplex operation.
M3_RXD[3:0]	I U	[3] J22 [1] J24 [2] J23 [0] J25	Receive Data. Clocked on rising edge of M3_RXCLK.
M3_RXDV	I D	J21	Receive Data Valid. Active high. This signal is clocked on the rising edge of M3_RXCLK. It is asserted when valid data is on the M3_RXD bus.
M3_RXER	I D	H26	Receive Error. Active high signal indicating an error has been detected. Normally valid when M3_RXDV is asserted. Can be used in conjunction with M3_RXD when M3_RXDV signal is de-asserted to indicate a False Carrier.
M3_CRS	I D	H24	Carrier Sense. This asynchronous signal is asserted when either the transmission or reception device is non-idle. It is active high.

Table 6 - MII Port 3 Interface Package Ball Definition

MII Port 3 Note: This port must not be used to receive data at the same time as port 2, they are mutually exclusive.			
Signal	**I/O**	**Package Balls**	**Description**
M3_TXCLK	I U	H25	MII only - Transmit Clock Accepts the following frequencies: 25.0 MHz MII 100 Mbit/s
M3_TXD[3:0]	O	[3] K23 [1] L25 [2] L26 [0] L24	Transmit Data. Clocked on rising edge of M3_TXCLK
M3_TXEN	O	K24	Transmit Enable. Asserted when the MAC has data to transmit, synchronously to M3_TXCLK with the first pre-amble of the packet to be sent. Remains asserted until the end of the packet transmission. Active high.
M3_TXER	O	K25	Transmit Error. Transmitted synchronously with respect to M3_TXCLK, and active high. When asserted (with M3_TXEN also asserted) the ZL50130 will transmit a non-valid symbol, somewhere in the transmitted frame.

Table 6 - MII Port 3 Interface Package Ball Definition

3.2 External Memory Interface

All External Memory Interface signals are 5 V tolerant.

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

If the External Memory Interface is unused, all input pins may be left unconnected.

Active low signals are designated by a # suffix, in accordance with the convention used in common memory data sheets.

Signal	I/O	Package Balls				Description
RAM_DATA[63:0]	IU/ OT	[63]	AD7	[31]	K3	Buffer memory data. Synchronous to rising edge of SYSTEM_CLK.
		[62]	AE6	[30]	K4	
		[61]	AF5	[29]	J1	
		[60]	AB8	[28]	J2	
		[59]	AC7	[27]	J3	
		[58]	AD6	[26]	J4	
		[57]	AE5	[25]	H1	
		[56]	AF4	[24]	H2	
		[55]	AF3	[23]	H3	
		[54]	AE4	[22]	J5	
		[53]	AD5	[21]	G1	
		[52]	AA8	[20]	J6	
		[51]	AB7	[19]	H4	
		[50]	AF2	[18]	G2	
		[49]	AC6	[17]	H5	
		[48]	AE3	[16]	G3	
		[47]	AD4	[15]	F1	
		[46]	AC5	[14]	G4	
		[45]	AA7	[13]	F2	
		[44]	AB6	[12]	F3	
		[43]	AB5	[11]	G5	
		[42]	AC4	[10]	E1	
		[41]	AD3	[9]	E2	
		[40]	AE2	[8]	G6	
		[39]	AA5	[7]	F5	
		[38]	AB4	[6]	F4	
		[37]	AC3	[5]	E3	
		[36]	AD2	[4]	E4	
		[35	AE1	[3]	D1	
		[34]	AD1	[2]	E5	
		[33]	W6	[1]	D2	
		[32]	Y5	[0]	D4	
RAM_PARITY[7:0]	IU/ OT	[7]	L1	[3]	L5	Buffer memory parity. Synchronous to rising edge of SYSTEM_CLK. Bit [7] is parity for data byte [63:56], bit [0] is parity for data byte [7:0].
		[6]	L2	[2]	L6	
		[5]	L3	[1]	K1	
		[4]	L4	[0]	K2	

Table 7 - External Memory Interface Package Ball Definition

Signal	I/O	Package Balls	Description
RAM_ADDR[19:0]	O	[19] R4 [9] P1 [18] T2 [8] N4 [17] T1 [7] N3 [16] P5 [6] N2 [15] R3 [5] M1 [14] R2 [4] M2 [13] P4 [3] M4 [12] R1 [2] M3 [11] P3 [1] M6 [10] P2 [0] M5	Buffer memory address output. Synchronous to rising edge of SYSTEM_CLK.
RAM_BW_A#	O	U2	Synchronous Byte Write Enable A (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[7:0].
RAM_BW_B#	O	T3	Synchronous Byte Write Enable B (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[15:8].
RAM_BW_C#	O	U3	Synchronous Byte Write Enable C (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[23:16].
RAM_BW_D#	O	V2	Synchronous Byte Write Enable D (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[31:24].
RAM_BW_E#	O	W1	Synchronous Byte Write Enable E (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[39:32].
RAM_BW_F#	O	V3	Synchronous Byte Write Enable F (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[47:40].
RAM_BW_G#	O	W2	Synchronous Byte Write Enable G (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[55:48]

Table 7 - External Memory Interface Package Ball Definition

Signal	I/O	Package Balls	Description
RAM_BW_H#	O	Y1	Synchronous Byte Write Enable H (Active Low). Must be asserted same clock cycle as RAM_ADDR. Enables RAM_DATA[63:56].
RAM_RW#	O	U4	Read/Write Enable output Read = high Write = low

Table 7 - External Memory Interface Package Ball Definition

3.3 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] AF25 [30] AB19 [29] AD22 [28] AE23 [27] AC20 [26] AF24 [25] AE22 [24] AD21 [23] AA17 [22] AB18 [21] AC19 [20] AD20 [19] AF23 [18] AE21 [17] AF22 [16] AC18 [15] AA16 [14] AD19 [13] AE20 [12] AB17 [11] AF21 [10] AC17 [9] AE19 [8] AA15 [7] AB16 [6] AD18 [5] AF19 [4] AE18 [3] AD17 [2] AF20 [1] AB15 [0] AF18	CPU Data Bus. Bi-directional data bus, synchronously transmitted with CPU_CLK rising edge. NOTE: as with all ports in the ZL50130 device, CPU_DATA[0] is the least significant bit (lsb).
CPU_ADDR[23:2]	I	[23] AB13 [22] AC13 [21] AD13 [20] AE13 [19] AF12 [18] AE12 [17] AD12 [16] AC12 [15] AF11 [14] AB12 [13] AE11 [12] AA12 [11] AD11 [10] AF10 [9] AC11 [8] AE10 [7] AD10 [6] AB11 [5] AF9 [4] AC10 [3] AE9 [2] AA11	CPU Address Bus. Address input from processor to ZL50130, synchronously transmitted with CPU_CLK rising edge. NOTE: as with all ports in the ZL50130 device, CPU_ADDR[2] is the least significant bit (lsb).

Table 8 - CPU Interface Package Ball Definition

Signal	I/O	Package Balls	Description
CPU_CS	I U	AF14	CPU Chip Select. Synchronous to rising edge of CPU_CLK and active low. Is asserted with CPU_TS_ALE. Must be asserted with CPU_OE to asynchronously enable the CPU_DATA output during a read, including DMA read.
CPU_WE	I	AD14	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 ZL50130. Asserted one clock cycle after CPU_TS_ALE
CPU_OE	I	AE14	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 ZL50130. Asserted one clock cycle after CPU_TS_ALE. Must be asserted with CPU_CS to asynchronously enable the CPU_DATA output during a read, including DMA read.
CPU_TS_ALE	I	AE15	Synchronous input with rising edge of CPU_CLK. Latch Enable (ALE), active high signal. Asserted with CPU_CS, for a single clock cycle.
CPU_SDACK1	I	AF15	CPU/DMA 1 Acknowledge Input. Active low synchronous to CPU_CLK rising edge. Used to acknowledge request from ZL50130 for a DMA write transaction. Only used for DMA transfers, not for normal register access.
CPU_SDACK2	I	AD15	CPU/DMA 2 Acknowledge Input Active low synchronous to CPU_CLK rising edge. Used to acknowledge request from ZL50130 for a DMA read transaction. Only used for DMA transfers, not for normal register access.

Table 8 - CPU Interface Package Ball Definition

Signal	I/O	Package Balls	Description
CPU_CLK	I	AC14	CPU PowerQUICC™ II Bus Interface clock input. 66 MHz clock, with minimum of 6ns high/low time. Used to time all host interface signals into and out of ZL50130 device.
$\overline{\text{CPU\_TA}}$	OT	AB14	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 ZL50130 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	AC15	CPU DMA 0 Request Output Active low synchronous to CPU_CLK rising edge. Asserted by ZL50130 to request the host initiates a DMA write. Only used for DMA transfers, not for normal register access.
$\overline{\text{CPU\_DREQ1}}$	OT	AE16	CPU DMA 1 Request Active low synchronous to CPU_CLK rising edge. Asserted by ZL50130 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.
$\overline{\text{CPU\_IREQO}}$	O	AF17	CPU Interrupt 0 Request (Active Low)
$\overline{\text{CPU\_IREQ1}}$	O	AD16	CPU Interrupt 1 Request (Active Low)

Table 8 - CPU Interface Package Ball Definition

3.4 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.

Signal	I/O	Package Balls	Description
SYSTEM_CLK	I	U6	System Clock Input. The system clock frequency is 100 MHz. The frequency must be accurate to within ±32 ppm in synchronous mode.
SYSTEM_RST	I	V4	System Reset Input. Active low. The system reset is asynchronous, and causes all registers within the ZL50130 to be reset to their default state.
SYSTEM_DEBUG	I	U5	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.

Table 9 - System Function Interface Package Ball Definition

3.5 Test Facilities

3.5.1 Administration and Control Interface

All Administration and Control Interface signals are 5 V tolerant.

Signal	I/O	Package Balls	Description
GPIO[15:0]	ID/ OT	[15] AA4 [14] AB3 [13] AC2 [12] AC1 [11] AB2 [10] Y4 [9] W5 [8] AA3 [7] AA2 [6] Y3 [5] AB1 [4] Y2 [3] W4 [2] V5 [1] AA1 [0] W3	General Purpose I/O pins. Connected to an internal register, so customer can set user-defined parameters. Bits [4:0] reserved at startup or reset for memory TDL setup. See the ZL50130 Programmers Model for more details.
TEST_MODE[2:0]	I D	[2] AF6 [1] AB9 [0] AC8	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 10 - Administration/Control Interface Package Ball Definition

3.5.2 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
$\overline{\text{JTAG\_TRST}}$	I U	AE7	JTAG Reset. Asynchronous reset. In normal operation this pin should be pulled low.
JTAG_TCK	I	AD8	JTAG Clock - maximum frequency is 25 MHz, typically run at 10 MHz. In normal operation this pin should be pulled either high or low.
JTAG_TMS	I U	AA10	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	AF7	JTAG test data input. Synchronous to JTAG_TCK.
JTAG_TDO	O	AC9	JTAG test data output. Synchronous to JTAG_TCK.

Table 11 - JTAG Interface Package Ball Definition

3.6 Miscellaneous Inputs

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

Signal	I/O	Package Balls	Description
PULL_LO	I	AF8 AD9	Reserved inputs, must be pulled low.
PULL_HI	I	AF16	Reserved input, must be pulled high.

Table 12 - Miscellaneous Inputs Package Ball Definitions

3.7 Power and Ground Connections

Signal	Package Balls				Description
VDD_IO	J9 J13 J17 L9 N9 R9 U9 V11 V15	J10 J14 J18 L18 N18 R18 U18 V12 V16	J11 J15 K9 M9 P9 T9 V9 V13 V17	J12 J16 K18 M18 P18 T18 V10 V14 V18	3.3 V VDD Power Supply for IO Ring
GND	A1 F6 L11 L15 M13 N1 N13 N22 P12 P16 R13 T5 T14 AA6 AF13	A13 F21 L12 L16 M14 N5 N14 N26 P13 P24 R14 T11 T15 AA21 AF26	A26 K5 L13 M11 M15 N11 N15 P6 P14 R11 R15 T12 T16 AB10	E22 K22 L14 M12 M16 N12 N16 P11 P15 R12 R16 T13 T24 AF1	0 V Ground Supply
VDD_CORE	F7 F20 K6 N6 Y6 AA13	F12 H6 K21 T21 Y21 AA14	F15 H21 M21 V6 AA9 AA18		1.8 V VDD Power Supply for Core Region
A1VDD	T6				1.8 V PLL Power Supply

Table 13 - Power and Ground Package Ball Definition

4.0 Miscellaneous

4.1 JTAG Interface and Board Level Test Features.

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

4.2 External Component Requirements

- Direct connection to PowerQUICC™ II (MPC8260) host processor and associated memory, but can support other processors with appropriate glue logic.
- Ethernet PHY for each MAC port
- Optional ZBT-SRAM for extended packet memory buffer depth

4.3 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
- 3.3 V I/O Supply rail with 5 V tolerance
- 1.8 V Core Supply rail

5.0 Memory Map and Register definitions

All memory map and register definitions are included in the ZL50130 Programmers Model document.

6.0 Test Modes Operation

6.1 Overview

The ZL50130 supports the following modes of operation.

6.1.1 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 it's own dedicated .bsdl file which fully describes it's boundary scan architecture.

6.1.2 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.

6.2 Test Mode Control

The System Test Mode is selected using the dedicated device input bus test_mode[2:0] as follows in Table 6.3.

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

Table 14 - Test Mode Control

6.3 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.

6.4 System Tri-state Mode

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

7.0 DC Characteristics

7.1 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}	-	±10	mA
Continuous current at digital outputs	I_O	-	±15	mA
Package power dissipation	PD	-	4	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.

7.2 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

7.3 DC Characteristics

Characteristics	Symbol	Min.	Typ.	Max.	Units	Test Condition
Input Leakage	I_{LEIP}			±1	µA	No pull up/down V_{DD} = 3.6 V
Output (High impedance) Leakage	I_{LEOP}			1	µA	No pull up/down V_{DD} = 3.6 V
Input Capacitance	C_{IP}		2		pF	
Output Capacitance	C_{OP}		4		pF	
Pullup Current	I_{PU}		-33		µA	Input at 0 V
Pulldown Current	I_{PD}		33		µA	Input at V_{DD}
Core 1.8 V supply current	I_{DD_CORE}		700	850	mA	
PLL 1.8 V supply current	I_{DD_PLL}			1.30	mA	
I/O 3.3 V supply current	I_{DD_IO}		20	30	mA	

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.

7.4 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	

Table 15 - Input Levels

7.5 Output Levels

Characteristics	Symbol	Min.	Typ.	Max.	Units	Test Condition
Output Low Voltage	V_{OL}			0.4	V	
Output High Voltage	V_{OH}	2.4			V	
Output Low Current	I_{OL}		1.6		mA	
Output High Current	I_{OH}		1.2		mA	

Table 16 - Output Levels

Further details of the DC characteristics will be added in a future specification updated.

8.0 AC Characteristics

8.1 Packet Interface Timing

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

8.1.1 MII Transmit Timing

Parameter	Symbol	100 Mbit/s			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 17 - MII Transmit Timing - 100 Mbit/s

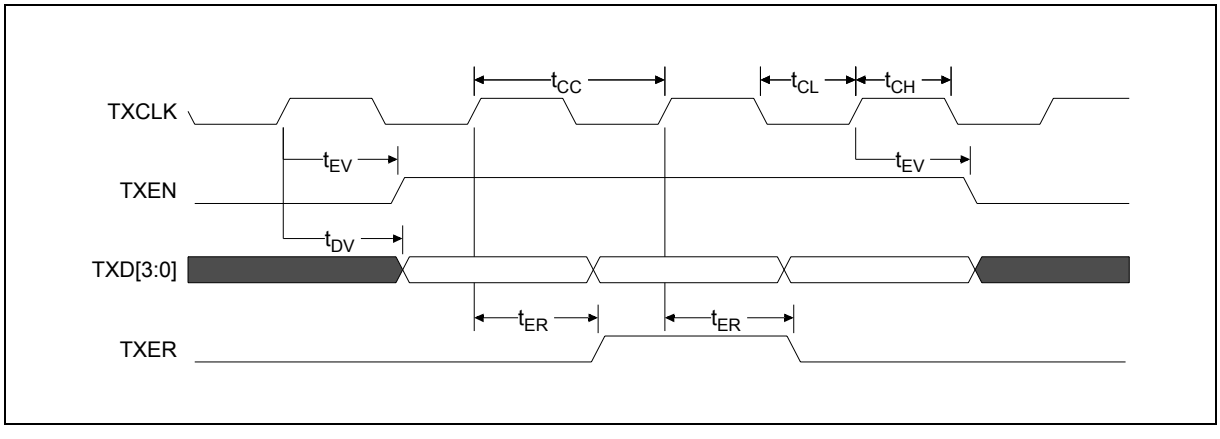


Figure 6 - MII Transmit Timing Diagram

8.1.2 MII Receive Timing

Parameter	Symbol	100 Mbit/s			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 (RXCL edge)	t_{ERS}	10	-	-	ns	
RXER input hold time (RXCLK rising edge)	t_{ERH}	5	-	-	ns	

Table 18 - MII Receive Timing - 100 Mbit/s

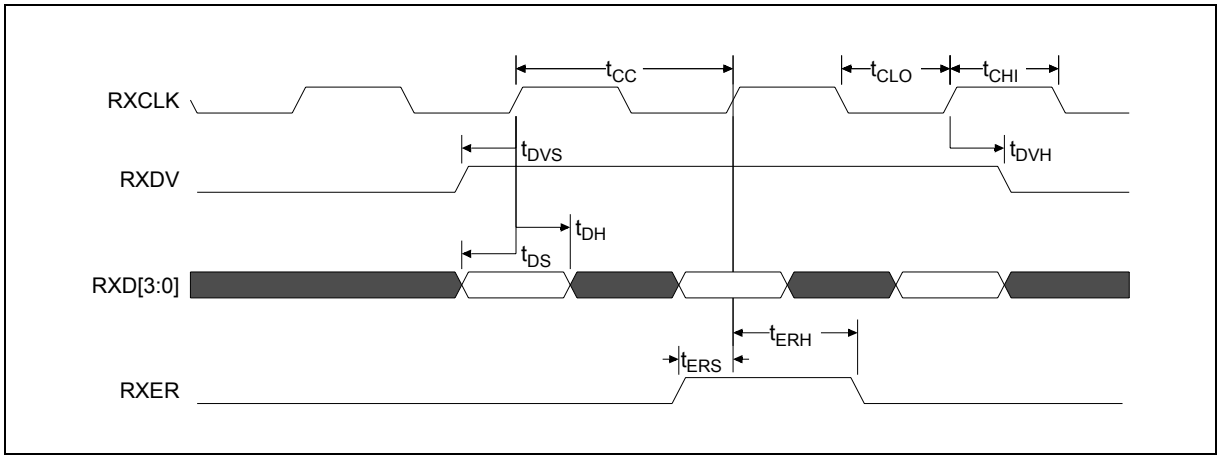


Figure 7 - MII Receive Timing Diagram

8.1.3 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	tMR	-	-	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}	10	-	-	ns	Note 1
M_MDIO Output Delay (M_MDC rising edge)	t_{MD}	1	-	300	ns	Note 2

Table 19 - 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

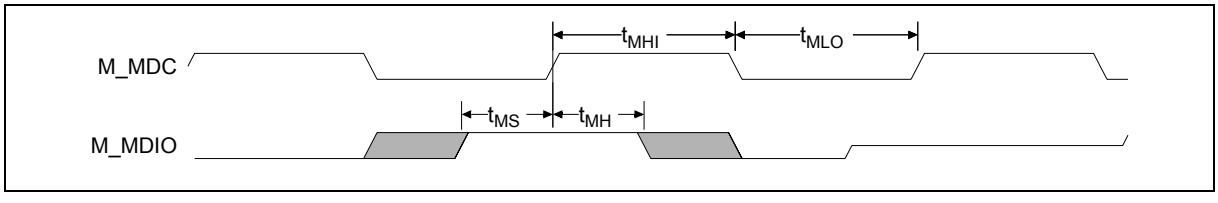


Figure 8 - Management Interface Timing for Ethernet Port - Read

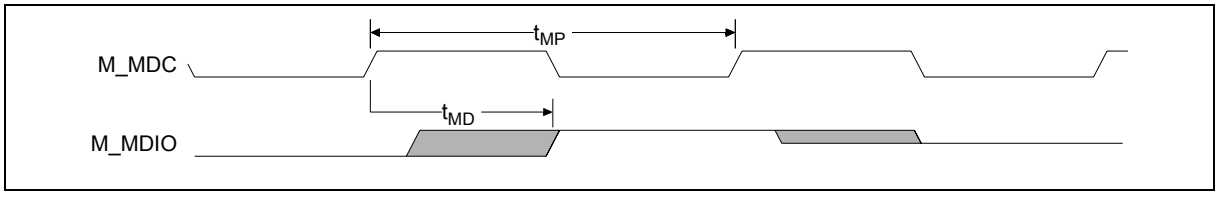


Figure 9 - Management Interface Timing for Ethernet Port - Write

8.2 External Memory Interface Timing

The timings for the External Memory Interface are based on the requirements of a ZBT-SRAM device, with the system clock speed at 100 MHz.

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
RAM_DATA[63:0] Output Valid Delay	t_{RDV}	-	-	4	ns	Load C_L = 30 pF
RAM_RW/RAM_ADDR[19:0] Delay	t_{RAV}	-	-	4	ns	Load C_L = 30 pF Note 1
RAM_BW[7:0]# Delay	t_{RBW}	-	-	4	ns	Load C_L = 30 pF
RAM_DATA[63:0] Setup Time	t_{RDS}	2	-	-	ns	
RAM_DATA[63:0] Hold Time	t_{RDH}	0.5	-	-	ns	
RAM_PARITY[7:0] Output Valid Delay	t_{RPV}	-	-	4	ns	Load C_L = 30 pF
RAM_PARITY[7:0] Setup Time	t_{RPS}	2	-	-	ns	
RAM_PARITY[7:0] Hold Time	t_{RPS}	0.5	-	-	ns	

Table 20 - External Memory Timing

Note 1: Must be capable of driving TWO separate RAM loads simultaneously

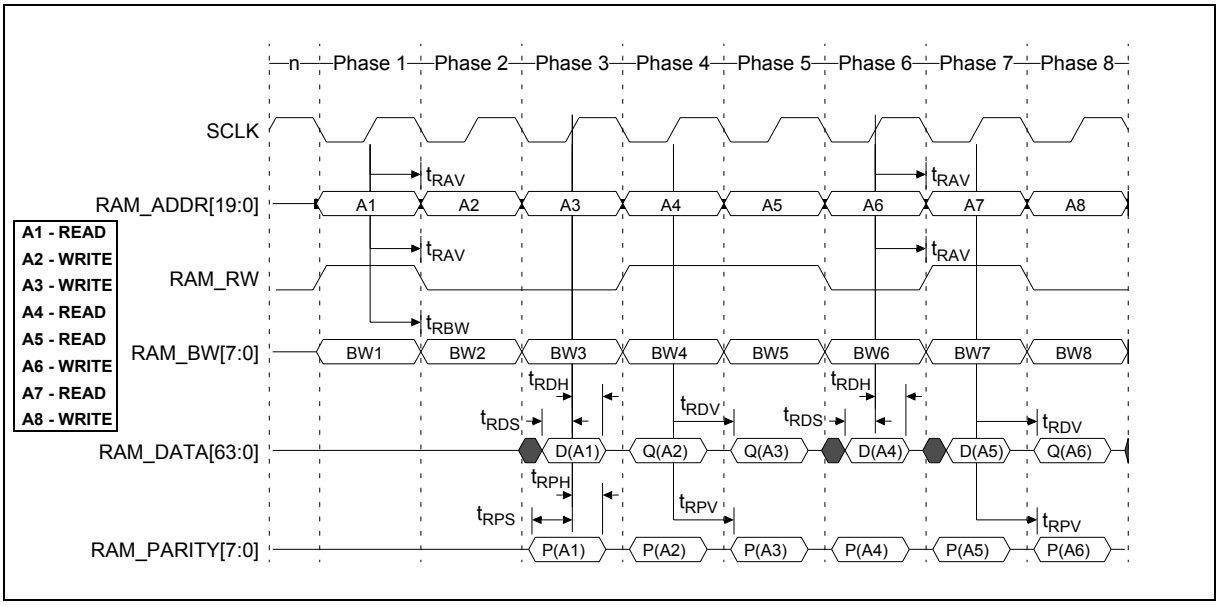


Figure 10 - External RAM Read and Write timing

8.3 CPU Interface Timing

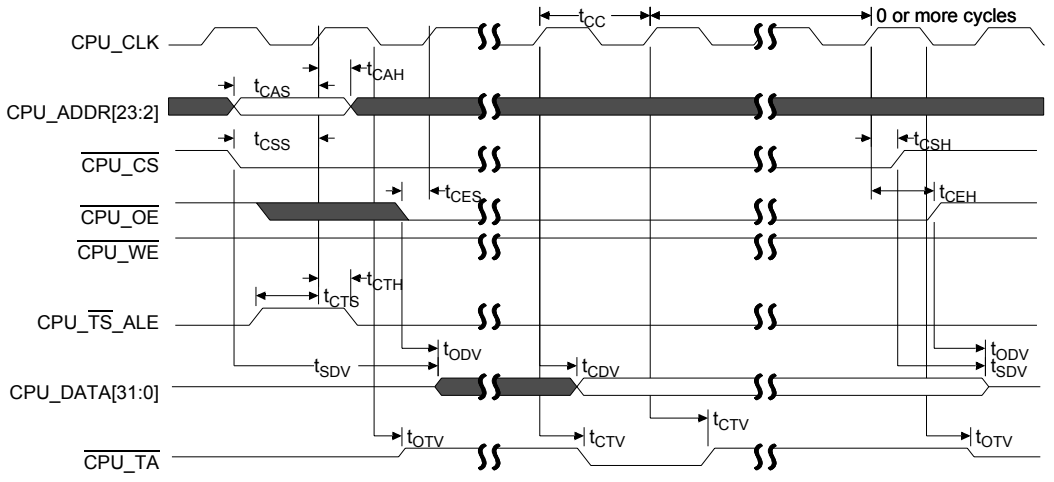
Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
CPU_CLK Period	t_{CC}		15.152		ns	
CPU_CLK High Time	t_{CCH}	6			ns	
CPU_CLK Low Time	t_{CCL}	6			ns	
CPU_CLK Rise Time	t_{CCR}			4	ns	
CPU_CLK Fall Time	t_{CCF}			4	ns	
CPU_ADDR[23:2] Setup Time	t_{CAS}	4			ns	
CPU_ADDR[23:2] Hold Time	t_{CAH}	2			ns	
CPU_DATA[31:0] 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	
$\overline{\text{CPU\_SDACK1}}$/$\overline{\text{CPU\_SDACK2}}$ Setup Time	t_{CKS}	2			ns	
$\overline{\text{CPU\_SDACK1}}$/$\overline{\text{CPU\_SDACK2}}$ Hold Time	t_{CKH}	2			ns	Note 1
$\overline{\text{CPU\_TA}}$ Output Valid Delay	t_{CTV}	2		11.3	ns	Note 1,2
$\overline{\text{CPU\_DREQ0}}$/$\overline{\text{CPU\_DREQ1}}$ Output Valid Delay	t_{CWV}	2		6	ns	Note 1
$\overline{\text{CPU\_IREQ0}}$/$\overline{\text{CPU\_IREQ1}}$ 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}			9.5	ns	

Table 21 - CPU Timing Specification

Note 1: Load = 50 pF maximum

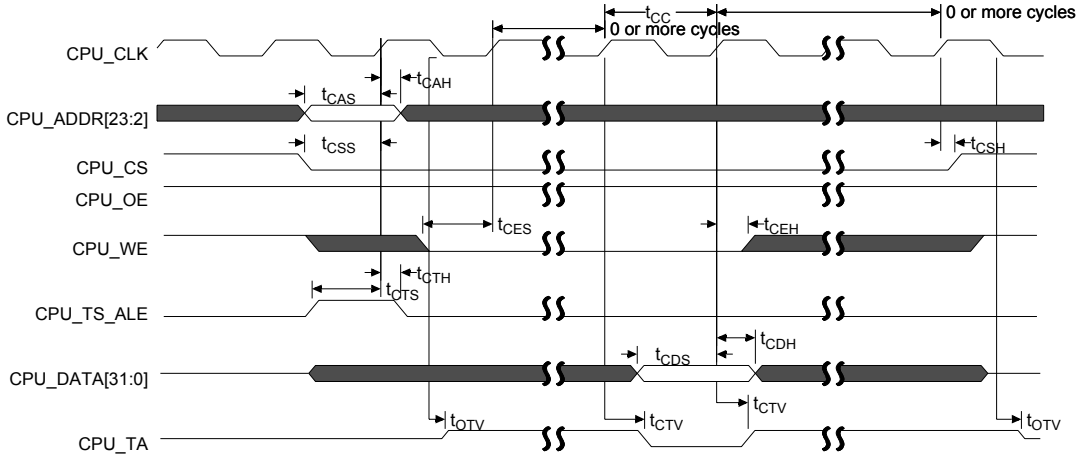
Note 2: The maximum value of t_{CTV} may cause setup violations if directly connected to the MPC8260. See Section 10.2 for details of how to accommodate this during board design

The actual point where read/write data is transferred occurs at the positive clock edge following the assertion of CPU_TA, not at the positive clock edge during the assertion of CPU_TA.



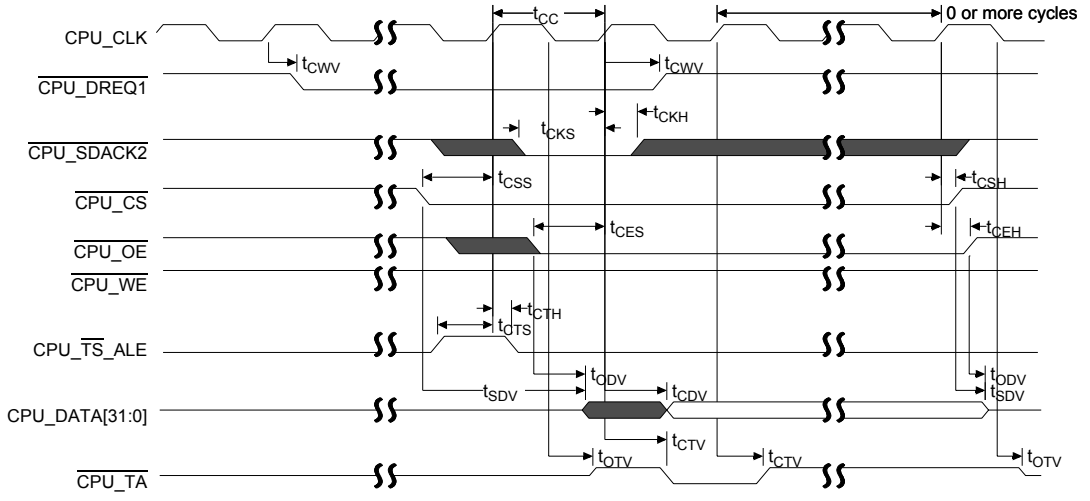
NOTE: CPU_DATA is valid when CPU_TA is asserted. CPU_DATA will remain valid while both CPU_CS and CPU_OE are asserted. CPU_TA will continue to be driven until CPU_CS is deasserted. CPU_CS and CPU_OE must BOTH be asserted to enable the CPU_DATA output.

Figure 11 - CPU Read - MPC8260



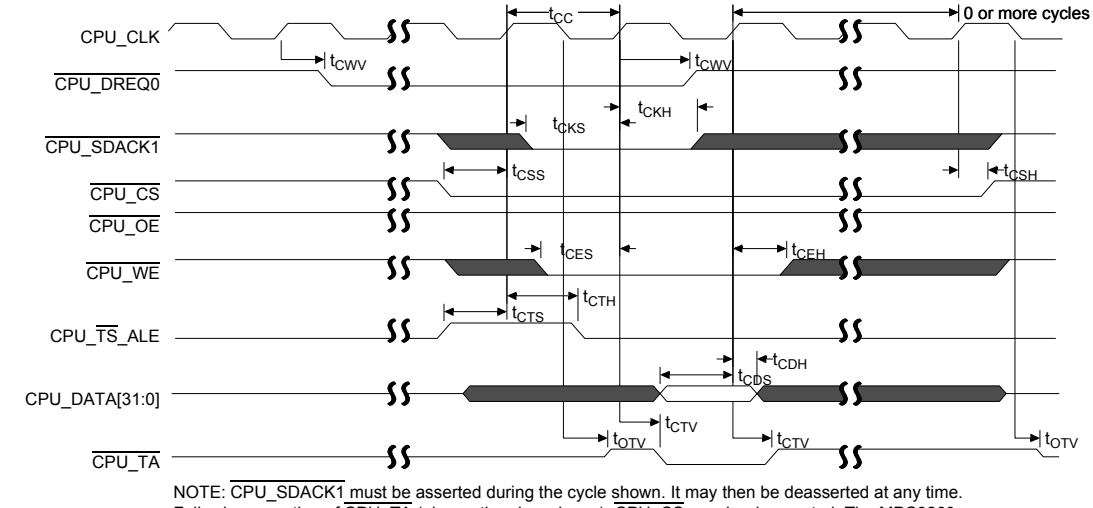
NOTE: Following assertion of CPU_TA, CPU_CS may be deasserted. The MPC8260 will continue to assert CPU_CS until CPU_TA has been synchronized internally. CPU_TA will continue to be driven until CPU_CS is finally deasserted. During continued assertion of CPU_CS, CPU_WE and CPU_DATA may be removed.

Figure 12 - CPU Write - MPC8260



NOTE: CPU_SDACK2 must be asserted during the cycle shown. It may then be deasserted at any time. CPU_DATA is valid when CPU_TA is asserted (always timed as shown). CPU_DATA will remain valid while CPU_CS and CPU_OE are asserted. CPU_TA will continue to be driven until CPU_CS is deasserted. CPU_CS and CPU_OE must BOTH be asserted to enable the CPU_DATA output.

Figure 13 - CPU DMA Read - MPC8260



NOTE: CPU_SDACK1 must be asserted during the cycle shown. It may then be deasserted at any time. Following assertion of CPU_TA (always timed as shown), CPU_CS may be deasserted. The MPC8260 will continue to assert CPU_CS until CPU_TA has been synchronized internally. CPU_TA will continue to be driven until CPU_CS is finally deasserted. During continued assertion of CPU_CS, CPU_WE and CPU_DATA may be removed.

Figure 14 - CPU DMA Write - MPC8260

8.4 System Function Port

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
SYSTEM_CLK Frequency	CLK_{FR}	-	100	-	MHz	Note 1 and Note 2
SYSTEM_CLK accuracy (synchronous master mode)	CLK_{ACS}	-	-	±30	ppm	Note 3
SYSTEM_CLK accuracy (synchronous slave mode and asynchronous mode)	CLK_{ACA}	-	-	±200	ppm	Note 4

Table 22 - 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 0.1ppm/ºC, a 10ºC change in temperature while the DPLL is in will result in a frequency accuracy offset of 1ppm. 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.

8.5 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	
$\overline{\text{JTAG\_TRST}}$ setup time	t_{RSTSU}	10	-	-	ns	With respect to JTAG_CLK falling edge. Note 1
$\overline{\text{JTAG\_TRST}}$ 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 23 - 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

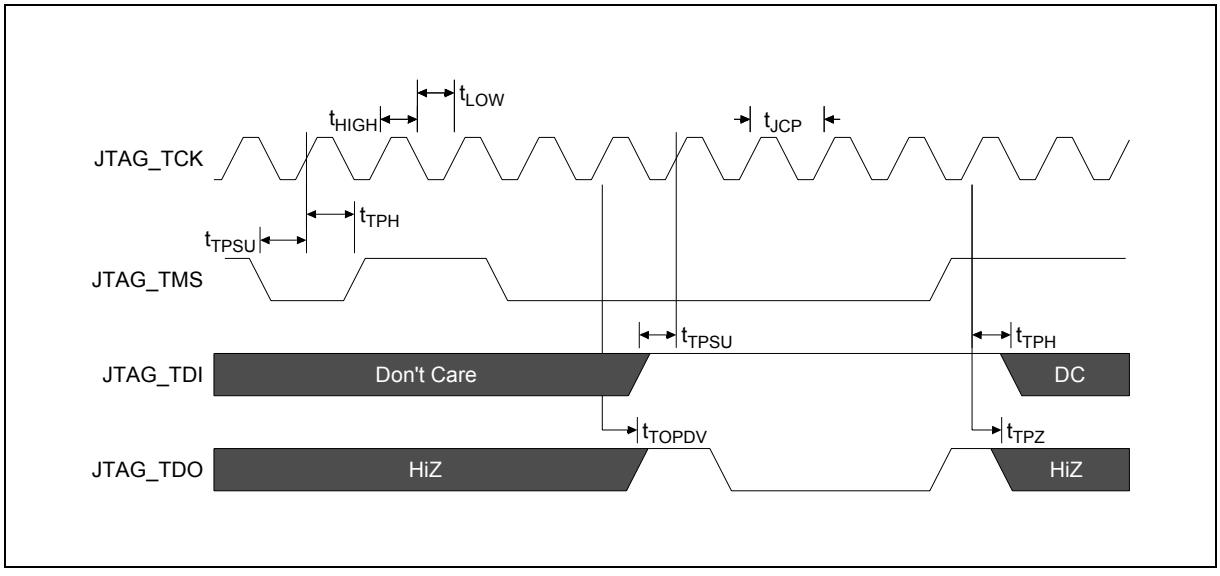


Figure 15 - JTAG Signal Timing

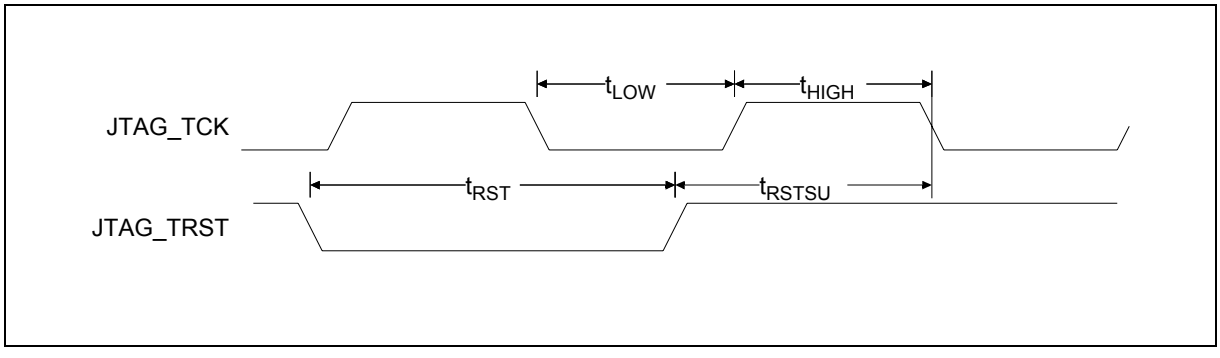


Figure 16 - JTAG Clock and Reset Timing

9.0 Power Up sequence

To power up the ZL50130 the following procedure must be used:

- The Core supply must never exceed the I/O supply by more than 0.5 V_{DC}.
- 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 17.

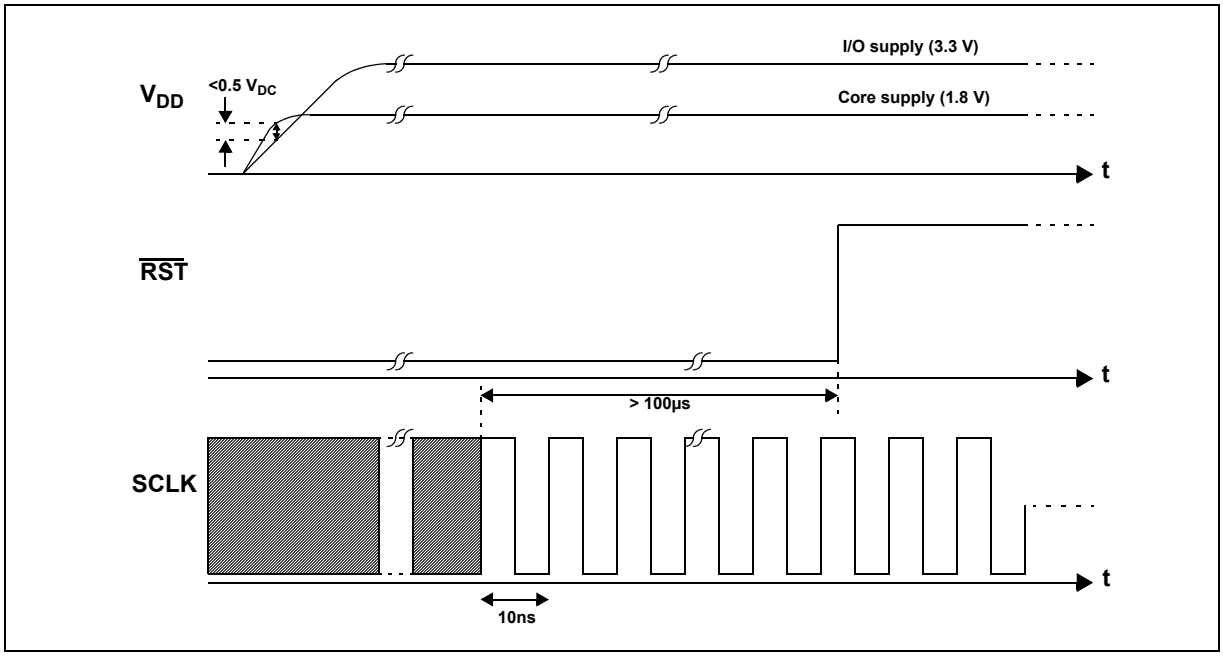


Figure 17 - Powering Up the ZL50130

10.0 Design and Layout Guidelines

This guide will provide information and guidance for PCB layouts when using the ZL50130. Specific areas of guidance are:

- High Speed Clock and Data, Outputs and Inputs
- CPU_TA Output

10.1 High Speed Clock & Data Interfaces

On the ZL50130 series of devices there are three 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:

- External Memory Interface
- MAC Interfaces
- CPU Interface

In general the output drivers used in the ZL50130 are capable of driving modest capacitive loads with a reasonably fast edge speed (<2.5 ns). Therefore these outputs are not designed to drive multiple loads, connectors, backplanes or cables. 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.

the equivalent length of rise time = rise time (ps) / delay (ps/mm)

For example:

Typical FR4 board delay = 6.8 ps/mm
Typical rise/fall time for a ZL50130 output = 2.5 ns

critical track length = (1/6) x (2500/6.8) = 61 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.

10.1.1 External Memory Interface - special considerations during layout

The timing of address, data and control are all related to the system clock which is also used by the external SSRAM to clock these signals. Therefore the propagation delay of the clock to the ZL50130 and the SSRAM must be matched to within 250 ps, worst case conditions. Trace lengths of theses signals must also be minimized (<100 mm) and matched to ensure correct operation under all conditions.

10.1.2 MAC Interface - special considerations during layout

The MII interface passes data to and from the ZL50130 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.

10.1.3 Summary

Particular effort should be made to minimize crosstalk from ZL50130 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.

10.2 CPU TA Output

The CPU_TA output signal from the ZL50130 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 recommend that the circuit shown in Figure 18 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

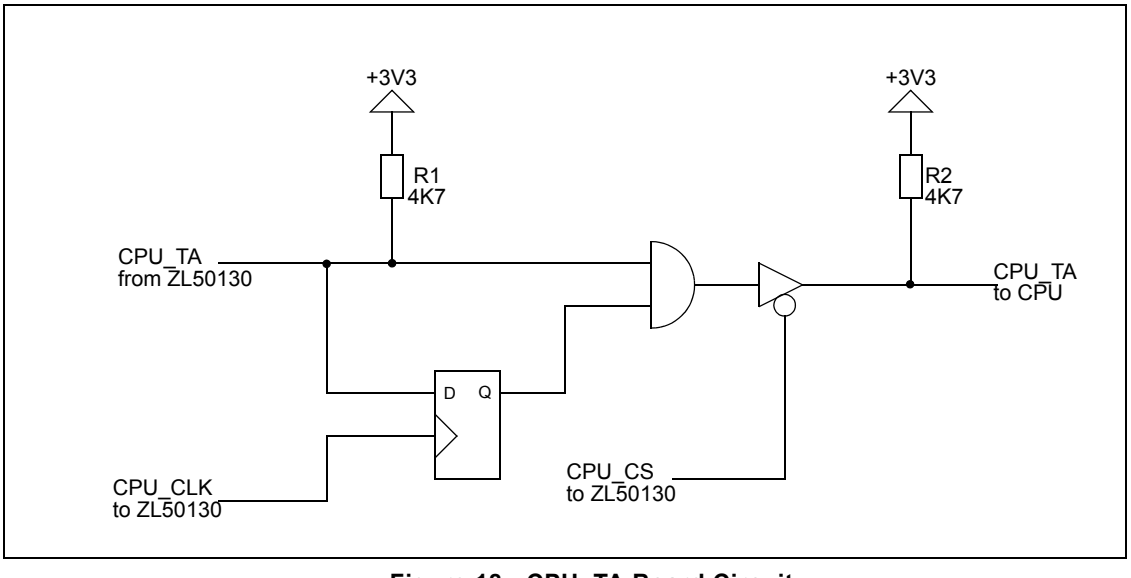


Figure 18 - CPU_TA Board Circuit

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 ZL50130 and that the clock to the 74LCX74 is derived from the same clock source as that input to the ZL50130.

11.0 Physical Specification

The ZL50130 will be packaged in a PBGA device.

Features:

- Body Size: 35 mm x 35 mm (typ)
- Ball Count: 552
- Ball Pitch: 1.27 mm (typ)
- Ball Matrix: 26 x 26
- Ball Diameter: 0.75 mm (typ)
- Total Package Thickness: 2.33 mm (typ)

ZL50130 Package view from TOP side. Note that ball A1 is non-chamfered corner.

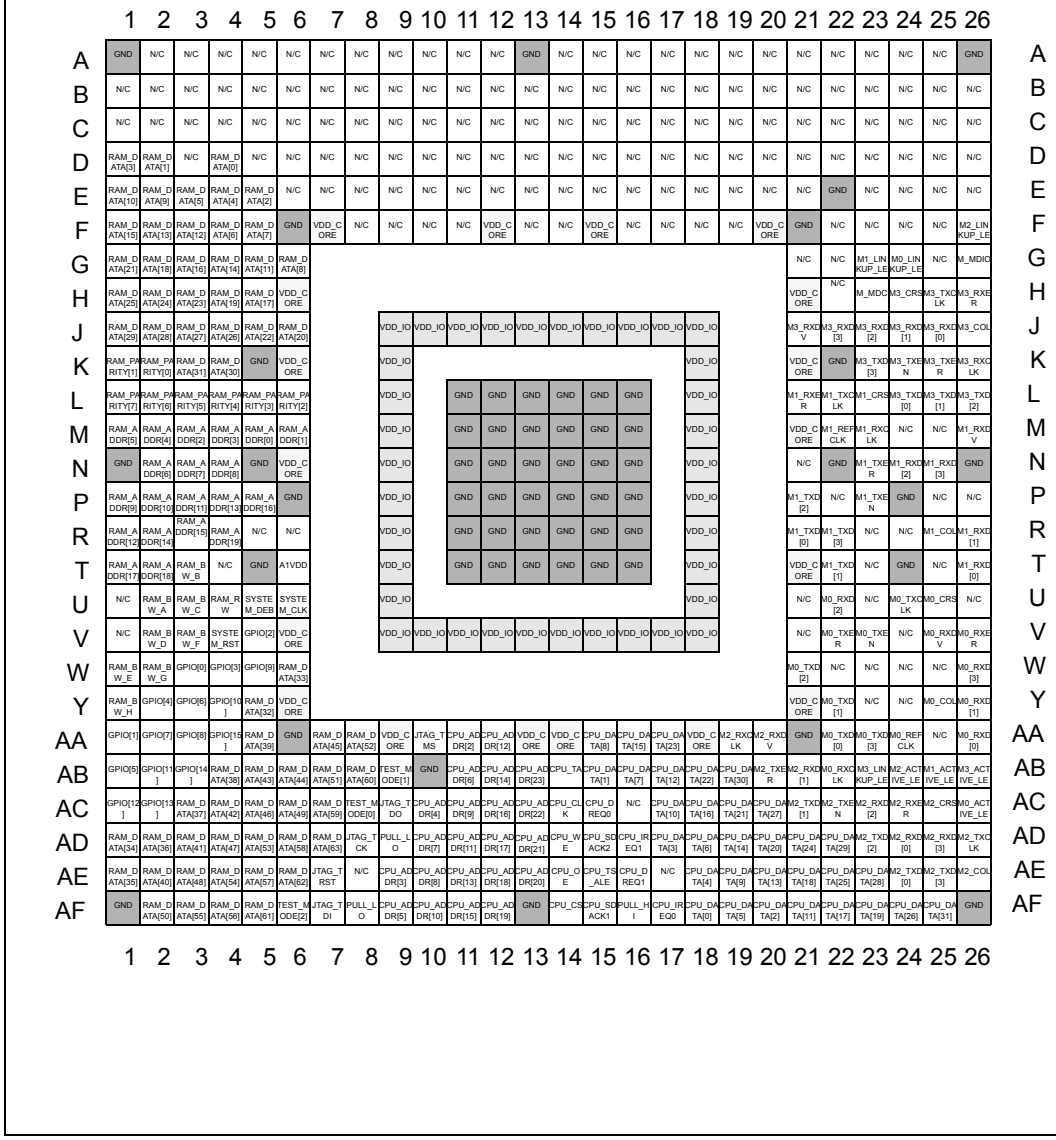


Figure 19 - ZL50130 Package View and Ball Positions

Ball Signal Assignment

Ball Number	Signal Name
A1	GND
A2	N/C
A3	N/C
A4	N/C
A5	N/C
A6	N/C
A7	N/C
A8	N/C
A9	N/C
A10	N/C
A11	N/C
A12	N/C
A13	GND
A14	N/C
A15	N/C
A16	N/C
A17	N/C
A18	N/C
A19	N/C
A20	N/C
A21	N/C
A22	N/C
A23	N/C
A24	N/C
A25	N/C
A26	GND
B1	N/C
B2	N/C
B3	N/C
B4	N/C
B5	N/C
B6	N/C
B7	N/C
B8	N/C
B9	N/C
B10	N/C
B11	N/C
B12	N/C
B13	N/C
B14	N/C
B15	N/C
B16	N/C
B17	N/C
B18	N/C
B19	N/C
B20	N/C
B21	N/C
B22	N/C
B23	N/C
B24	N/C
B25	N/C
B26	N/C
C1	N/C
C2	N/C
C3	N/C
C4	N/C
C5	N/C
C6	N/C
C7	N/C
C8	N/C
C9	N/C
C10	N/C
C11	N/C
C12	N/C
C13	N/C
C14	N/C
C15	N/C
C16	N/C
C17	N/C
C18	N/C
C19	N/C
C20	N/C
C21	N/C
C22	N/C
C23	N/C
C24	N/C
C25	N/C
C26	N/C
D1	RAM_DATA[3]
D2	RAM_DATA[1]
D3	N/C
D4	RAM_DATA[0]
D5	N/C
D6	N/C
D7	N/C
D8	N/C
D9	N/C
D10	N/C
D11	N/C
D12	N/C
D13	N/C
D14	N/C
D15	N/C
D16	N/C
D17	N/C
D18	N/C
D19	N/C
D20	N/C
D21	N/C
D22	N/C
D23	N/C

Ball Number	Signal Name
D24	N/C
D25	N/C
D26	N/C
E1	RAM_DATA[10]
E2	RAM_DATA[9]
E3	RAM_DATA[5]
E4	RAM_DATA[4]
E5	RAM_DATA[2]
E6	N/C
E7	N/C
E8	N/C
E9	N/C
E10	N/C
E11	N/C
E12	N/C
E13	N/C
E14	N/C
E15	N/C
E16	N/C
E17	N/C
E18	N/C
E19	N/C
E20	N/C
E21	N/C
E22	GND
E23	N/C
E24	N/C
E25	N/C
E26	N/C
F1	RAM_DATA[15]
F2	RAM_DATA[13]
F3	RAM_DATA[12]
F4	RAM_DATA[6]
F5	RAM_DATA[7]

Ball Number	Signal Name
F6	GND
F7	VDD_CORE
F8	N/C
F9	N/C
F10	N/C
F11	N/C
F12	VDD_CORE
F13	N/C
F14	N/C
F15	VDD_CORE
F16	N/C
F17	N/C
F18	N/C
F19	N/C
F20	VDD_CORE
F21	GND
F22	N/C
F23	N/C
F24	N/C
F25	N/C
F26	M2_LINKUP_LED
G1	RAM_DATA[21]
G2	RAM_DATA[18]
G3	RAM_DATA[16]
G4	RAM_DATA[14]
G5	RAM_DATA[11]
G6	RAM_DATA[8]
G21	N/C
G22	N/C
G23	M1_LINKUP_LED
G24	M0_LINKUP_LED
G25	N/C
G26	M_MDIO
H1	RAM_DATA[25]

Ball Number	Signal Name
H2	RAM_DATA[24]
H3	RAM_DATA[23]
H4	RAM_DATA[19]
H5	RAM_DATA[17]
H6	VDD_CORE
H21	VDD_CORE
H22	N/C
H23	M_MDC
H24	M3_CRS
H25	M3_TXCLK
H26	M3_RXER
J1	RAM_DATA[29]
J2	RAM_DATA[28]
J3	RAM_DATA[27]
J4	RAM_DATA[26]
J5	RAM_DATA[22]
J6	RAM_DATA[20]
J9	VDD_IO
J10	VDD_IO
J11	VDD_IO
J12	VDD_IO
J13	VDD_IO
J14	VDD_IO
J15	VDD_IO
J16	VDD_IO
J17	VDD_IO
J18	VDD_IO
J21	M3_RXDV
J22	M3_RXD[3]
J23	M3_RXD[2]
J24	M3_RXD[1]
J25	M3_RXD[0]
J26	M3_COL
K1	RAM_PARITY[1]

Ball Number	Signal Name
K2	RAM_PARITY[0]
K3	RAM_DATA[31]
K4	RAM_DATA[30]
K5	GND
K6	VDD_CORE
K9	VDD_IO
K18	VDD_IO
K21	VDD_CORE
K22	GND
K23	M3_TXD[3]
K24	M3_TXEN
K25	M3_TXER
K26	M3_RXCLK
L1	RAM_PARITY[7]
L2	RAM_PARITY[6]
L3	RAM_PARITY[5]
L4	RAM_PARITY[4]
L5	RAM_PARITY[3]
L6	RAM_PARITY[2]
L9	VDD_IO
L11	GND
L12	GND
L13	GND
L14	GND
L15	GND
L16	GND
L18	VDD_IO
L21	M1_RXER
L22	M1_TXCLK
L23	M1_CRS
L24	M3_TXD[0]
L25	M3_TXD[1]
L26	M3_TXD[2]
M1	RAM_ADDR[5]

Ball Number	Signal Name
M2	RAM_ADDR[4]
M3	RAM_ADDR[2]
M4	RAM_ADDR[3]
M5	RAM_ADDR[0]
M6	RAM_ADDR[1]
M9	VDD_IO
M11	GND
M12	GND
M13	GND
M14	GND
M15	GND
M16	GND
M18	VDD_IO
M21	VDD_CORE
M22	M1_REFCLK
M23	M1_RXCLK
M24	N/C
M25	N/C
M26	M1_RXDV
N1	GND
N2	RAM_ADDR[6]
N3	RAM_ADDR[7]
N4	RAM_ADDR[8]
N5	GND
N6	VDD_CORE
N9	VDD_IO
N11	GND
N12	GND
N13	GND
N14	GND
N15	GND
N16	GND
N18	VDD_IO
N21	N/C

Ball Number	Signal Name
N22	GND
N23	M1_TXER
N24	M1_RXD[2]
N25	M1_RXD[3]
N26	GND
P1	RAM_ADDR[9]
P2	RAM_ADDR[10]
P3	RAM_ADDR[11]
P4	RAM_ADDR[13]
P5	RAM_ADDR[16]
P6	GND
P9	VDD_IO
P11	GND
P12	GND
P13	GND
P14	GND
P15	GND
P16	GND
P18	VDD_IO
P21	M1_TXD[2]
P22	N/C
P23	M1_TXEN
P24	GND
P25	N/C
P26	N/C
R1	RAM_ADDR[12]
R2	RAM_ADDR[14]
R3	RAM_ADDR[15]
R4	RAM_ADDR[19]
R5	N/C
R6	N/C
R9	VDD_IO
R11	GND
R12	GND

Ball Number	Signal Name
R13	GND
R14	GND
R15	GND
R16	GND
R18	VDD_IO
R21	M1_TXD[0]
R22	M1_TXD[3]
R23	N/C
R24	N/C
R25	M1_COL
R26	M1_RXD[1]
T1	RAM_ADDR[17]
T2	RAM_ADDR[18]
T3	RAM_BW_B
T4	N/C
T5	GND
T6	A1VDD
T9	VDD_IO
T11	GND
T12	GND
T13	GND
T14	GND
T15	GND
T16	GND
T18	VDD_IO
T21	VDD_CORE
T22	M1_TXD[1]
T23	N/C
T24	GND
T25	N/C
T26	M1_RXD[0]
U1	N/C
U2	RAM_BW_A
U3	RAM_BW_C

Ball Number	Signal Name
U4	RAM_RW
U5	SYSTEM_DEBUG
U6	SYSTEM_CLK
U9	VDD_IO
U18	VDD_IO
U21	N/C
U22	M0_RXD[2]
U23	N/C
U24	M0_TXCLK
U25	M0_CRS
U26	N/C
V1	N/C
V2	RAM_BW_D
V3	RAM_BW_F
V4	SYSTEM_RST
V5	GPIO[2]
V6	VDD_CORE
V9	VDD_IO
V10	VDD_IO
V11	VDD_IO
V12	VDD_IO
V13	VDD_IO
V14	VDD_IO
V15	VDD_IO
V16	VDD_IO
V17	VDD_IO
V18	VDD_IO
V21	N/C
V22	M0_TXER
V23	M0_TXEN
V24	N/C
V25	M0_RXDV
V26	M0_RXER
W1	RAM_BW_E

Ball Number	Signal Name
W2	RAM_BW_G
W3	GPIO[0]
W4	GPIO[3]
W5	GPIO[9]
W6	RAM_DATA[33]
W21	M0_TXD[2]
W22	N/C
W23	N/C
W24	N/C
W25	N/C
W26	M0_RXD[3]
Y1	RAM_BW_H
Y2	GPIO[4]
Y3	GPIO[6]
Y4	GPIO[10]
Y5	RAM_DATA[32]
Y6	VDD_CORE
Y21	VDD_CORE
Y22	M0_TXD[1]
Y23	N/C
Y24	N/C
Y25	M0_COL
Y26	M0_RXD[1]
AA1	GPIO[1]
AA2	GPIO[7]
AA3	GPIO[8]
AA4	GPIO[15]
AA5	RAM_DATA[39]
AA6	GND
AA7	RAM_DATA[45]
AA8	RAM_DATA[52]
AA9	VDD_CORE
AA10	JTAG_TMS
AA11	CPU_ADDR[2]

Ball Number	Signal Name
AA12	CPU_ADDR[12]
AA13	VDD_CORE
AA14	VDD_CORE
AA15	CPU_DATA[8]
AA16	CPU_DATA[15]
AA17	CPU_DATA[23]
AA18	VDD_CORE
AA19†	M2_RXCLK
AA20†	M2_RXDV
AA21	GND
AA22	M0_TXD[0]
AA23	M0_TXD[3]
AA24	M0_REFCLK
AA25	N/C
AA26	M0_RXD[0]
AB1	GPIO[5]
AB2	GPIO[11]
AB3	GPIO[14]
AB4	RAM_DATA[38]
AB5	RAM_DATA[43]
AB6	RAM_DATA[44]
AB7	RAM_DATA[51]
AB8	RAM_DATA[60]
AB9	TEST_MODE[1]
AB10	GND
AB11	CPU_ADDR[6]
AB12	CPU_ADDR[14]
AB13	CPU_ADDR[23]
AB14	CPU_TA
AB15	CPU_DATA[1]
AB16	CPU_DATA[7]
AB17	CPU_DATA[12]
AB18	CPU_DATA[22]
AB19	CPU_DATA[30]

Ball Number	Signal Name
AB20	M2_TXER
AB21	M2_RXD[1]
AB22	M0_RXCLK
AB23	M3_LINKUP_LED
AB24	M2_ACTIVE_LED
AB25	M1_ACTIVE_LED
AB26	M3_ACTIVE_LED
AC1	GPIO[12]
AC2	GPIO[13]
AC3	RAM_DATA[37]
AC4	RAM_DATA[42]
AC5	RAM_DATA[46]
AC6	RAM_DATA[49]
AC7	RAM_DATA[59]
AC8	TEST_MODE[0]
AC9	JTAG_TDO
AC10	CPU_ADDR[4]
AC11	CPU_ADDR[9]
AC12	CPU_ADDR[16]
AC13	CPU_ADDR[22]
AC14	CPU_CLK
AC15	CPU_DREQ0
AC16	N/C
AC17	CPU_DATA[10]
AC18	CPU_DATA[16]
AC19	CPU_DATA[21]
AC20	CPU_DATA[27]
AC21	M2_TXD[1]
AC22	M2_TXEN
AC23	M2_RXD[2]
AC24	M2_RXER
AC25	M2_CRS
AC26	M0_ACTIVE_LED
AD1	RAM_DATA[34]

Ball Number	Signal Name
AD2	RAM_DATA[36]
AD3	RAM_DATA[41]
AD4	RAM_DATA[47]
AD5	RAM_DATA[53]
AD6	RAM_DATA[58]
AD7	RAM_DATA[63]
AD8	JTAG_TCK
AD9	PULL_LO
AD10	CPU_ADDR[7]
AD11	CPU_ADDR[11]
AD12	CPU_ADDR[17]
AD13	CPU_ADDR[21]
AD14	CPU_WE
AD15	CPU_SDACK2
AD16	CPU_IREQ1
AD17	CPU_DATA[3]
AD18	CPU_DATA[6]
AD19	CPU_DATA[14]
AD20	CPU_DATA[20]
AD21	CPU_DATA[24]
AD22	CPU_DATA[29]
AD23	M2_TXD[2]
AD24	M2_RXD[0]
AD25	M2_RXD[3]
AD26	M2_TXCLK
AE1	RAM_DATA[35]
AE2	RAM_DATA[40]
AE3	RAM_DATA[48]
AE4	RAM_DATA[54]
AE5	RAM_DATA[57]
AE6	RAM_DATA[62]
AE7	JTAG_TRST
AE8	N/C
AE9	CPU_ADDR[3]

Ball Number	Signal Name
AE10	CPU_ADDR[8]
AE11	CPU_ADDR[13]
AE12	CPU_ADDR[18]
AE13	CPU_ADDR[20]
AE14	CPU_OE
AE15	CPU_TS_ALE
AE16	CPU_DREQ1
AE17	N/C
AE18	CPU_DATA[4]
AE19	CPU_DATA[9]
AE20	CPU_DATA[13]
AE21	CPU_DATA[18]
AE22	CPU_DATA[25]
AE23	CPU_DATA[28]
AE24	M2_TXD[0]
AE25	M2_TXD[3]
AE26	M2_COL
AF1	GND
AF2	RAM_DATA[50]
AF3	RAM_DATA[55]
AF4	RAM_DATA[56]
AF5	RAM_DATA[61]
AF6	TEST_MODE[2]
AF7	JTAG_TDI
AF8	PULL_LO
AF9	CPU_ADDR[5]
AF10	CPU_ADDR[10]
AF11	CPU_ADDR[15]
AF12	CPU_ADDR[19]
AF13	GND
AF14	CPU_CS
AF15	CPU_SDACK1
AF16	PULL_HI
AF17	CPU_IREQ0

Ball Number	Signal Name
AF18	CPU_DATA[0]
AF19	CPU_DATA[5]
AF20	CPU_DATA[2]
AF21	CPU_DATA[11]
AF22	CPU_DATA[17]
AF23	CPU_DATA[19]
AF24	CPU_DATA[26]
AF25	CPU_DATA[31]
AF26	GND

N/C - No Connect Pins, these unused pins must be left open circuit.

12.0 Reference Documents

12.1 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
- 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
- IETF's PWE3 draft-ietf-l2tpext-l2tp-base-02
- *Optional Packet Memory Device* - Micron MT55L128L32P1 8 Mb ZBT-SRAM

12.2 Zarlink ZL50130 Product Related Documentation

- ZL50130 Programmers Model
- ZL50130 API Users Guide
- ZL50130 Product Preview

13.0 Related Products

- MVTX260x 24 Port 10/100 Mbit/s Ethernet Switch
- MVTX280x 4/8 Port Gigabit Ethernet Switch
- ZL50418 Ethernet Switches

14.0 Glossary

CONTEXT A programmed connection representing a unique packet stream.

CPU Central Processing Unit

DMA Direct Memory Access

IETF Internet Engineering Task Force

IP Internet Protocol (version 4, RFC 791, version 6, RFC 2460)

JTAG Joint Test Algorithms Group (generally used to refer to a standard way of providing a board-level test facility)

L2TP Layer 2 Tunneling Protocol (RFC 2661)

LAN Local Area Network

MAC Media Access Control

MII Media Independent Interface

MIB Management Information Base

MPLS Multi Protocol Label Switching

PLL Phase Locked Loop

PSN Packet Switched Network

PWE3 Pseudo-Wire End-to-End Emulation (a working group of the IETF)

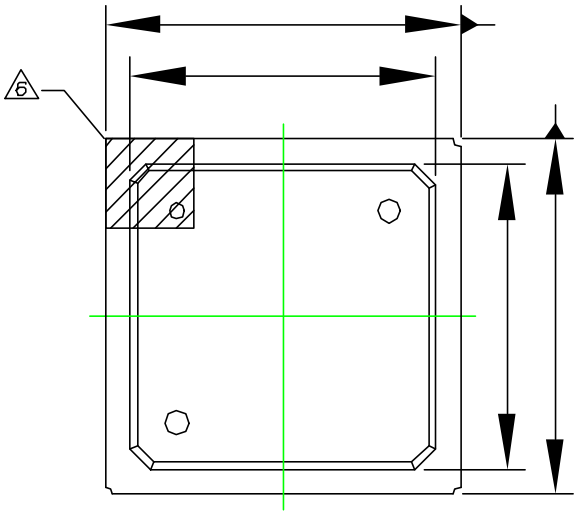
SSRAM Synchronous Static Random Access Memory

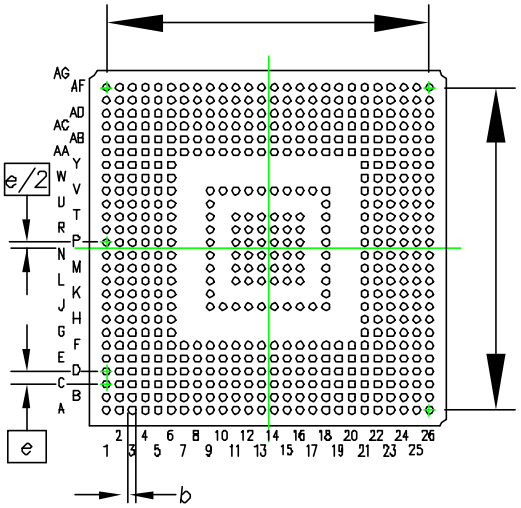
UDP User Datagram Protocol (RFC 768)

VLAN Virtual Local Area Network

WFQ Weighted Fair Queuing

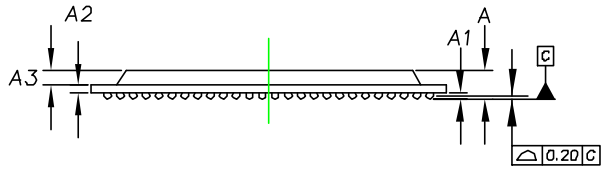
ZBT Zero Bus Turnaround, a type of synchronous SRAM





SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	2.13	2.33	2.53	.084	.092	.100
A1	0.50	0.60	0.70	.020	.024	.028
A2	0.51	0.56	0.61	.020	.022	.024
A3	1.12	1.17	1.22	.044	.046	.048
b	0.60	0.75	0.90	.024	.030	.035
D	34.80	35.00	35.20	1.370	1.378	1.386
D1	31.75 BSC			1.250 BSC		
D2	29.90	30.00	30.10	1.177	1.181	1.185
E	34.80	35.00	35.20	1.370	1.378	1.386
E1	31.75 BSC			1.250 BSC		
E2	29.90	30.00	30.10	1.177	1.181	1.185
e	1.27 BSC			.050 BSC		

Confirms to JEDEC MS-034 BAR-2 iss. A



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

ISSUE	1			
ACN	213837			
DATE	12Dec02			
APPRD.				

ZARLINK SEMICONDUCTOR

Previous package codes

BP

Package Code GA

Package Outline for 552 PBGA (35x35)mm, 1.27mm pitch

GPD00809

For more information about all Zarlink products
visit our Web Site at

www.zarlink.com

TECHNICAL DOCUMENTATION - NOT FOR RESALE