ZL50409

Managed 9-Port 10/100M Ethernet Switch

Data Sheet

July 2003

Features

- 8 10/100 Mbps auto-negotiating ports with RMII, MII & GPSI interface options
- 1 10/100 Mbps auto-negotiating MII port (port 9) that can be used as a WAN uplink or as a 9th port
- Supports both managed and unmanaged options
  - Supports 8/16-bit parallel or serial+MII interface in managed mode
  - Serial interface in lightly managed or unmanaged mode
- Internal 2Mbit (256KB) buffer memory
- Up to 4K MAC addresses
- Provides port based and ID tagged VLAN support (IEEE 802.1Q), up to 4K VLANs
- Supports IP Multicast with IGMP snooping
- Supports spanning tree with CPU, on per port basis
- 8 port trunking groups with up to 8 ports per group
- Failover Backplane Features
  - Link Heart Beat
- Rate Control (both ingress and egress)
  - Bandwidth rationing, Bandwidth on demand, SLA (Service Level Agreement)
  - Smooth out traffic to uplink ports

Ordering Information

ZL50409GDC	208 Pin LBGA

-40°C to +85°C

  - Ingress Rate Control
    - Back pressure
    - Flow Control
    - WRED (Weighted Random Early Discard)
  - Egress Rate Control
    - per queue shaper (Port 9)
    - WRED
  - Down to 16kbps Rate Control granularity
- Packet Filtering and Port Security
  - Static address filtering for source and/or destination MAC
  - Static MAC address not subject to aging
  - Secure mode freezes MAC address learning (each port may independently use this mode)
- Full Duplex Ethernet IEEE 802.3x Flow Control
- Backpressure flow control for Half Duplex ports

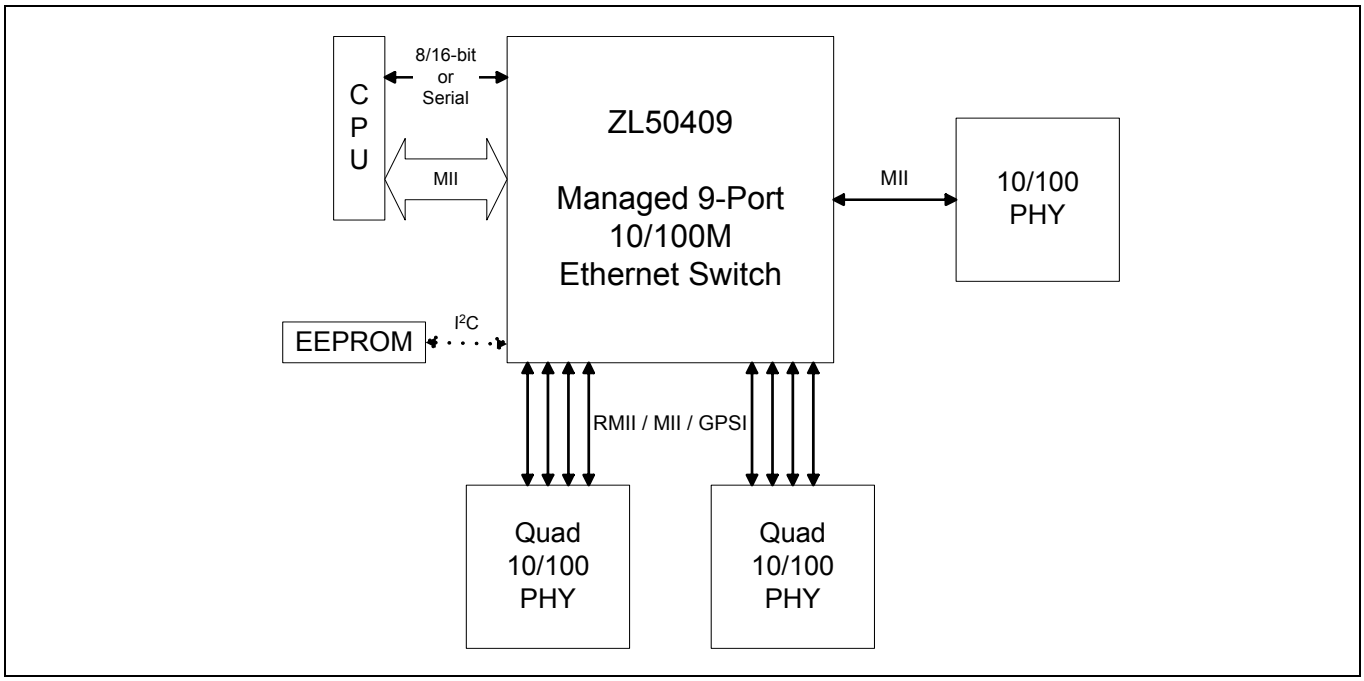


Figure 1 - System Block Diagram

- Supports Ethernet multicasting and broadcasting and flooding control
- Supports per-system option to enable flow control for best effort frames even on QoS enabled ports
- QoS Support
  - Supports IEEE 802.1p/Q Quality of Service with 2 transmission priority queues (4 for MII port), with strict priority and WFQ service disciplines
  - Provides 2 levels of dropping precedence with WRED mechanism
  - User controls the WRED thresholds.
  - Buffer management: per class and per port buffer reservations
  - Port-based priority: VLAN priority in a tagged frame can be overwritten by the priority of Port VLAN ID
- Classification based on:
  - Port based priority
  - VLAN Priority field in VLAN tagged frame
  - DS/TOS field in IP packet
  - UDP/TCP logical ports: 8 hard-wired and 8 programmable ports, including one programmable range
- The precedence of the above classifications is programmable
- MIB Statistics counters for all ports
- Hardware auto-negotiation through serial management interface (MDIO) for Ethernet ports
- I^2C EEPROM for configuration in unmanaged mode
- Built-in reset logic triggered by system malfunction
- Built-In Self Test for internal SRAM
- IEEE-1149.1 (JTAG) test port

Description

The ZL50409 is a low density, low cost, high performance, non-blocking Ethernet switch chip. A single chip provides 9 ports at 10/100 Mbps and a CPU interface for managed and unmanaged switch applications.

The chip supports up to 4K MAC addresses and up to 4K port-based Virtual LANs (VLANs).

With strict priority and/or WFQ transmission scheduling and WRED dropping schemes, the ZL50409 provides powerful QoS functions for various multimedia and mission-critical applications. The chip provides 2 transmission priorities (4 priorities for MII port) and 2 levels of dropping precedence. Each packet is assigned a transmission priority and dropping precedence based on the VLAN priority field in a VLAN tagged frame, or the DS/TOS field, or the UDP/TCP logical port fields in IP packets. The ZL50409 recognizes a total of 16 UDP/TCP logical ports, 8 hard-wired and 8 programmable (including one programmable range).

The ZL50409 supports 8 groups of port trunking/load sharing. Each group can contain up to 8 ports. Port trunking/load sharing can be used to group ports between interlinked switches to increase the effective network bandwidth.

In half-duplex mode, all ports support backpressure flow control, to minimize the risk of losing data during long activity bursts. In full-duplex mode, IEEE 802.3x flow control is provided. The ZL50409 also supports a per-system option to enable flow control for best effort frames, even on QoS-enabled ports.

Statistical information for SNMP and the Remote Monitoring Management Information Base (RMON MIB) are collected independently for all ports. Access to these statistical counters/registers is provided via the CPU interface. SNMP Management frames can be received and transmitted via the CPU interface, creating a complete network management solution.

The ZL50409 is fabricated using 0.18 micron technology. The ZL50409 is packaged in a 208-pin Ball Grid Array package.

Table of Contents

1.0 BGA and Ball Signal Descriptions

1.1 BGA Views (Top-View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A	SCLK	P_CS#	P_RD#	P_WE#	P_DATA1	P_DATA3	P_DATA5	P_DATA7	P_DATA9	P_DATA11	P_DATA13	P_DATA15	REF_CLK	RSVD	M9_MTXCK	M9_TXEN	A
B	P_INT#	P_A0	P_A1	P_A2	P_DATA0	P_DATA2	P_DATA4	P_DATA6	P_DATA8	P_DATA10	P_DATA12	P_DATA14	TCK	TMS	RSVD	M9_RXCK	B
C	RESETOUT#	TSTOUT1	TSTOUT3	TSTOUT5	TSTOUT6	TSTOUT7	TSTOUT9	TSTOUT11	TSTOUT12	TSTOUT14	TSTOUT15	TRST#	TDI	RSVD	M9_CRS	M9_COL	C
D	RESIN#	TSTOUT0	TSTOUT2	TSTOUT4	3.3V	SCAN_EN	TSTOUT8	TSTOUT10	1.8V	TSTOUT13	TDO	3.3V	RSVD	RSVD	M9_RXDV	RSVD	D
E	M2_COL	M0_COL	M1_COL	3.3V									3.3V	RSVD	RSVD	RSVD	E
F	M_MDC	M_MDIO	M0_RXD2	M0_RXD3									M9_RXD2	M9_RXD3	RSVD	RSVD	F
G	M0_RXD0	M0_RXD1	M0_RXCK	M0_TXD3			GND	GND	GND	GND			M9_RXD0	M9_RXD1	M9_TXD2	M9_TXD3	G
H	M0_CRS	M0_TXEN	M0_TXD2	1.8V			GND	GND	GND	GND			1.8V	M7_COL	M9_TXD0	M9_TXD1	H
J	M0_TXD0	M0_TXD1	M0_TXCK	M1_RXD3			GND	GND	GND	GND			M7_TXD3	M7_TXCK	M7_TXD1	M7_TXD0	J
K	M1_RXD0	M1_RXD1	M1_RXD2	M1_RXCK			GND	GND	GND	GND			M7_TXD2	M7_RXD2	M7_TXEN	M7_CRS	K
L	M1_CRS	M1_TXEN	M1_TXD2	M1_TXD3									M7_RXD3	M7_RXCK	M7_RXD1	M7_RXD0	L
M	M1_TXD0	M1_TXD1	M1_TXCK	3.3V									3.3V	M6_COL	M5_COL	M4_COL	M
N	M2_RXD3	M2_TXCK	M2_TXD3	M3_RXD3	3.3V	M3_TXD3	1.8V	M4_RXD3	M4_TXCK	M4_TXD3	M5_RXD3	M5_TXCK	M5_TXD3	M6_RXD3	M6_TXCK	M6_TXD3	N
P	M2_RXD2	M2_RXCK	M2_TXD2	M3_RXD2	M3_RXCK	M3_TXD2	M3_COL	M4_RXD2	M4_RXCK	M4_TXD2	M5_RXD2	M5_RXCK	M5_TXD2	M6_RXD2	M6_RXCK	M6_TXD2	P
R	M2_RXD1	M2_TXEN	M2_TXD1	M3_RXD1	M3_TXEN	M3_TXD1	M_CLK	M4_RXD1	M4_TXEN	M4_TXD1	M5_RXD1	M5_TXEN	M5_TXD1	M6_RXD1	M6_TXEN	M6_TXD1	R
T	M2_RXD0	M2_CRS	M2_TXD0	M3_RXD0	M3_CRS	M3_TXD0	M3_TXCK	M4_RXD0	M4_CRS	M4_TXD0	M5_RXD0	M5_CRS	M5_TXD0	M6_RXD0	M6_CRS	M6_TXD0	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

1.2 Power and Ground Distribution

G7-10, H7-10, J7-10, K7-10	GND	V_{SS}	Ground
D5, D12, E4, E13, M4, M13, N5	3.3V	V_{CC}	I/O Power
D9, H4, H13, N7	1.8V	V_{DD}	Core Power

1.3 Ball Signal Descriptions

All pins are CMOS type; all Input Pins are 5 Volt tolerance; and all Output Pins are 3.3 CMOS drive.

Notes

# =	Active low signal
Input =	Input signal
In-ST =	Input signal with Schmitt-Trigger
Output =	Output signal (Tri-State driver)
Out-OD=	Output signal with Open-Drain driver
I/O-TS =	Input & Output signal with Tri-State driver
I/O-OD =	Input & Output signal with Open-Drain driver

Ball Signal Description Table

Ball No(s)	Symbol	I/O	Description
CPU BUS Interface in Managed Mode			
A12, B12, A11, B11, A10, B10, A9, B9, A8, B8, A7, B7, A6, B6, A5, B5	P_DATA[15:0]	I/O-TS with pull up	Processor Bus Data Bit [15:0]. P_DATA[7:0] is used in 8-bit mode.
B4, B3, B2	P_A[2:0]	I/O with pull up	Processor Bus Address Bit [2:0]
A4	P_WE#	Input with pull up	CPU Bus-Write Enable
A3	P_RD#	I/O with pull up	CPU Bus-Read Enable
A2	P_CS#	Input with pull up	Chip Select
B1	P_INT#	Output	CPU Interrupt
Fast Ethernet Access Ports [7:0] MII			
L13, K14, L15, L16, N14, P14, R14, T14, N11, P11, R11, T11, N8, P8, R8, T8, N4, P4, R4, T4, N1, P1, R1, T1, J4, K3, K2, K1, F4, F3, G2, G1	M[7:0]_RXD[3:0]	Input with pull up	Ports [7:0] – Receive Data Bit [3:0]
K16, T15, T12, T9, T5, T2, L1, H1	M[7:0]_CRS_DV	Input with pull up	Ports [7:0] – Carrier Sense and Receive Data Valid
K15, R15, R12, R9, R5, R2, L2, H2	M[7:0]_TXEN	I/O-TS slew	Ports [7:0] – Transmit Enable These pins also serve as bootstrap pins.
J13, K13, J15, J16, N16, P16, R16, T16, N13, P13, R13, T13, N10, P10, R10, T10, N6, P6, R6, T6, N3, P3, R3, T3, L4, L3, M2, M1, G4, H3, J2, J1	M[7:0]_TXD[3:0]	Output, slew	Ports [7:0] – Transmit Data Bit [3:0]

Ball Signal Description Table (continued)

Ball No(s)	Symbol	I/O	Description
H14, M14, M15, M16, P7, E1, E3, E2	M[7:0]_COL	Input with pull down,	Ports[7:0] – Collision
J14, N15, N12, N9, T7, N2, M3, J3	M[7:0]_TXCLK	I/O with pull up	Ports[7:0] – Transmit Clock
L14, P15, P12, P9, P5, P2, K4, G3	M[7:0]_RXCLK	I/O with pull up	Ports[7:0] – Receive Clock
MII Ethernet Access Port			
G16, G15, H16, H15	M9_TXD[3:0]	Output	Transmit Data Bit [3:0]
D15	M9_RXDV	Input w/ pull up	Receive Data Valid
C15	M9_CRS	Input w/ pull down	Carrier Sense
C16	M9_COL	Input w/ pull down	Collision Detected
B16	M9_RXCLK	I/O w/ pull up	Receive Clock
F14, F13, G14, G13	M9_RXD[3:0]	Input w/ pull up	Receive Data Bit [3:0]
A16	M9_TXEN	Output w/ pull up	Transmit Data Enable
A15	M9_ MTXCLK	I/O w/ pull up	Transmit Clock
A13	REF_CLK	Input w/ pull up	MMAC Reference Clock
Test Interface			
D2, C2, D3, C3, D4, C4, C5, C6, D7, C7, D8, C8, C9, D10, C10, C11	TSTOUT[15:0]	I/O-TS	[15:4] Reserved [3] EEPROM checksum is good [2] Initialization Completed [1] Memory Self Test in progress [0] Initialization started These pins also serve as bootstrap pins.
Test Facility			
C13	TDI	Input w/pull up	JTAG Test
C12	TRST#	Input w/pull up	JTAG Test
B13	TCK	Input w/pull up	JTAG Test
B14	TMS	Input w/pull up	JTAG Test
D11	TDO	Output	JTAG Test
D6	SCAN_EN	Input	Scan Enable. Manufacturing test option. Should be externally pulled-down for proper operation
System Clock, Power, and Ground Pins			
A1	SCLK	Input	System Clock. Based on system requirement, SCLK needs to operate at difference frequency. SCLK requires 40/60% duty cycle clock

Ball Signal Description Table (continued)

Ball No(s)	Symbol	I/O	Description
D9, H4, H13, N7	V_{DD}	Power	+1.8 Volt DC Supply
D5, D12, E4, E13, M4, M13, N5	V_{CC}	Power	+3.3 Volt DC Supply
G7-10, H7-10, J7-10, K7-10	V_{SS}	Power Ground	Ground
MISC			
F15, F16, E15, E16, E14, D13, D14, C14, A14, B15, D16	RSVD	NC	Reserved Pins. Leave unconnected.
D1	RESIN#	Input	Reset Input
C1	RESETOUT#	Output	Reset PHY
F1	M_MDC	Output	MII Management Data Clock –
F2	M_MDIO	I/O-TS with pull up	MII Management Data I/O –
R7	M_CLK	Input	RMAC Reference Clock
Bootstrap Pins (1= pull up 0= pull down) Note: External pull-up/down resistors are required on all bootstrap pins for proper operation. Recommend 10K for pull-ups and 1K for pull-downs.			
D2	TSTOUT[0]	I/O-TS	Enable Debounce of STROBE signal Pullup – Enabled Pulldown - Disabled
C3, D3, C2	TSTOUT[3:1]	I/O-TS	Management interface operation mode: 000 – 16-bit parallel interface 001 – 8-bit parallel interface 010 – Serial with MII as Ethernet frame transfer interface. 011 – Serial only. CPU can transmit/receive frames with the serial interface. 111 – Unmanaged Serial. No CPU packet can be transmit or received with the serial interface. EEPROM can be used to configure the device at bootup. A one (1) indicates pullup. A zero (0) indicates pulldown.TSTOUT[1] is the Least Significant Bit (LSB).
C5, C4, D4	TSTOUT[6:4]	I/O-TS	Device ID. Default address of the device for serial interface. Up to 8 device can be sharing the serial management bus with different device ID. A one (1) indicates pullup. A zero (0) indicates pulldown. TSTOUT[4] is the Least Significant Bit (LSB).

Ball Signal Description Table (continued)

Ball No(s)	Symbol	I/O	Description
C6	TSTOUT7	I/O-TS	EEPROM not installed. Pullup: Not installed Pulldown: Installed
D7	TSTOUT8	I/O-TS Must be externally pulled-up	Manufacturing Option. Must be pulled-up.
C7	TSTOUT9	I/O-TS	Module Detect Pullup: Enable. In this mode, the device will detect the existence of a PHY (for hot swap purpose). Pulldown: Disable
D8	TSTOUT10	I/O-TS Must be externally pulled-down	Reserved. Must be pulled-down.
C8	TSTOUT11	I/O-TS	Power Saving Pullup: Enable Mac power saving mode Pulldown: Disable Mac power saving mode
C9	TSTOUT12	I/O-TS	Timeout Reset Enable Pullup: Enable Pulldown: Disable
D10	TSTOUT[15:13]	I/O-TS Must be externally pulled-up	Manufacturing Options. Must be pulled-up.
K15, R15, R12, R9, R5, R2, L2, H2	M[7:0]_TXEN	I/O-TS slew	User Defined Bootstrap: Usually used in conjuction with Module Detect to determine what interface to use for the inserted module. Can be read from BOOTSTRAP2 register

1.4 Signal Mapping in different operation mode

The ZL50409 Fast Ethernet ports (0-7) support 3 interface options: RMII, MII & GPSI. The table below summarizes the interface signals required for each interface and how they relate back to the Pin Symbol name shown in Table , “Ball Signal Description Table” on page 10.

Notes:
I – Input
O – Output
U – Pullup
D - Pulldown

Fast Ethernet Ports Pin Symbol	No Module	RMII Mode (ECR4Pn[4:3]='11')	MII Mode (ECR4Pn[4:3]='01')	GPSI Mode (ECR4Pn[4:3]='00')
M[7:0]_RXD0	(U)	M[7:0]_RXD0 (IU)	M[7:0]_RXD0 (IU)	M[7:0]_RXD (IU)
M[7:0]_RXD1	(U)	M[7:0]_RXD1 (IU)	M[7:0]_RXD1 (IU)	NC
M[7:0]_RXD2	(U)	NC	M[7:0]_RXD2 (IU)	NC
M[7:0]_RXD3	(U)	NC	M[7:0]_RXD3 (IU)	NC
M[7:0]_TXEN	(O)	M[7:0]_TXEN (O)	M[7:0]_TXEN (O)	M[7:0]_TXEN (O)
M[7:0]_CRS_DV	(U)	M[7:0]_CRS_DV (IU)	M[7:0]_DV (IU)	M[7:0]_CRS (IU)
M[7:0]_TXD0	(O)	M[7:0]_TXD0 (O)	M[7:0]_TXD0 (O)	M[7:0]_TXD (O)
M[7:0]_TXD1	(O)	M[7:0]_TXD1 (O)	M[7:0]_TXD1 (O)	NC
M[7:0]_TXD2	(O)	NC	M[7:0]_TXD2 (O)	NC
M[7:0]_TXD3	(O)	NC	M[7:0]_TXD3 (O)	NC
M[7:0]_COL	(D)	NC	M[7:0]_COL (ID)	M[7:0]_COL (ID)
M[7:0]_TXCLK	(U)	NC	M[7:0]_TXCLK (IOU)	M[7:0]_TXCLK (IOU)
M[7:0]_RXCLK	(U)	NC	M[7:0]_RXCLK (IOU)	M[7:0]_RXCLK (IOU)

Table 1 - Signal Mapping In Different Operation Mode

The ZL50409 CPU access support 5 interface options: 8 or 16-bit parallel, serial+MII (port 8), serial only, and unmanaged serial (with optional EEPROM). The table below summarizes the interface signals required for each interface, and how they relate back to the Pin Symbol name shown in Table , “Ball Signal Description Table” on page 10.

Management Interface Pin Symbol	16-bit CPU (TSTOUT[3:1]=’000’)	8-bit CPU (TSTOUT[3:1]=’001’)	Serial with MII (TSTOUT[3:1]=’010’)	Serial Only (TSTOUT[3:1]=’011’ or ‘111’)
P_A[0]	P_A[0] (IU)	P_A[0] (IU)	NC	SDA (IOU) (111 only)
P_A[1]	P_A[1] (IU)	P_A[1] (IU)	NC	SCL (OU) (111 only)
P_A[2]	P_A[2] (IU)	P_A[2] (IU)	NC	NC
P_WE#	P_WE# (IU)	P_WE# (IU)	STROBE (IU)	STROBE (IU)
P_RD#	P_RD# (IU)	P_RD# (IU)	DATAOUT (OU)	DATAOUT (OU)
P_CS#	P_CS# (IU)	P_CS# (IU)	DATAIN (IU)	DATAIN (IU)
P_INT#	P_INT# (O)	P_INT# (O)	P_INT# (O)	P_INT# (O)
P_DATA0	P_DATA0 (IOU)	P_DATA0 (IOU)	CPU_MII_TXD0 (OU)	NC
P_DATA1	P_DATA1 (IOU)	P_DATA1 (IOU)	CPU_MII_TXD1 (OU)	NC
P_DATA2	P_DATA2 (IOU)	P_DATA2 (IOU)	CPU_MII_TXD2 (OU)	NC
P_DATA3	P_DATA3 (IOU)	P_DATA3 (IOU)	CPU_MII_TXD3 (OU)	NC
P_DATA4	P_DATA4 (IOU)	P_DATA4 (IOU)	CPU_MII_TXCLK (OU)	NC
P_DATA5	P_DATA5 (IOU)	P_DATA5 (IOU)	CPU_MII_TXEN (OU)	NC
P_DATA6	P_DATA6 (IOU)	P_DATA6 (IOU)	NC	NC
P_DATA7	P_DATA7 (IOU)	P_DATA7 (IOU)	NC	NC
P_DATA8	P_DATA8 (IOU)	NC	CPU_MII_RXD0 (IU)	NC
P_DATA9	P_DATA9 (IOU)	NC	CPU_MII_RXD1 (IU)	NC
P_DATA10	P_DATA10 (IOU)	NC	CPU_MII_RXD2 (IU)	Nc
P_DATA11	P_DATA11 (IOU)	NC	CPU_MII_RXD3 (IU)	NC
P_DATA12	P_DATA12 (IOU)	NC	CPU_MII_RXCLK (OU)	NC
P_DATA13	P_DATA13 (IOU)	NC	CPU_MII_RXDV (IU)	NC
P_DATA14	P_DATA14 (IOU)	NC	NC	NC
P_DATA15	P_DATA15 (IOU)	NC	NC	NC

Table 2 - Signal Mapping In Different Operation Mode

1.5 Bootstrap Options

TSTOUT[15:0] and M[7:0]_TXEN pins serve as bootstrap pins during device power-up or reset. Please refer to "Typical Reset & Bootstrap Timing Diagram" on page 110 for more information on when the bootstrap pins are sampled. The bootstrap pins require external pull-up/down resistors for proper operation.

The table below summarizes the bootstrap options.

Feature	Description
CPU Interface	The ZL50409 allows the selection of 5 different management interfaces: 8/16-bit parallel, serial with MII, serial only and unmanaged serial with I2C EEPROM. TSTOUT[3:1] is used to select the interface options mentioned above. If the serial interface is selected, addition bootstrap options are required: • TSTOUT[0] enables or disables the DEBOUNCE feature (refer to "Synchronous Serial Interface" on page 24) • TSTOUT[6:4] selects the device ID Also, an optional I^2C EEPROM can be used to configure the device at power-up or reset. TSTOUT[7] selects the EEPROM option.
Ethernet Interface	The ZL50409 supports module hotswap on all it's ports. This is enabled via TSTOUT[9]. When enabled, bootstrap pins M[7:0]_TXEN (ports 0-7) are used to specify the module type to support multiple ethernet interfaces during module hotswap. Another feature is the ZL50409 MAC power savings mode. When enabled via TSTOUT[11], each port's MAC will detect inactivity on the port and go into a power savings state. Once activitiy is detected once again on the port, the MAC will come out of this state.
Misc. Features	One other feature selected via bootstrap is Timeout Reset Enable (TSTOUT[12]). This enables a monitoring block with the device which will detect if any hardware state machine is in a non-idle state for more than 5 seconds. Refer to section 2.12 for more details on this feature.

Table 3 - Bootstrap Features

2.0 Block Functionality

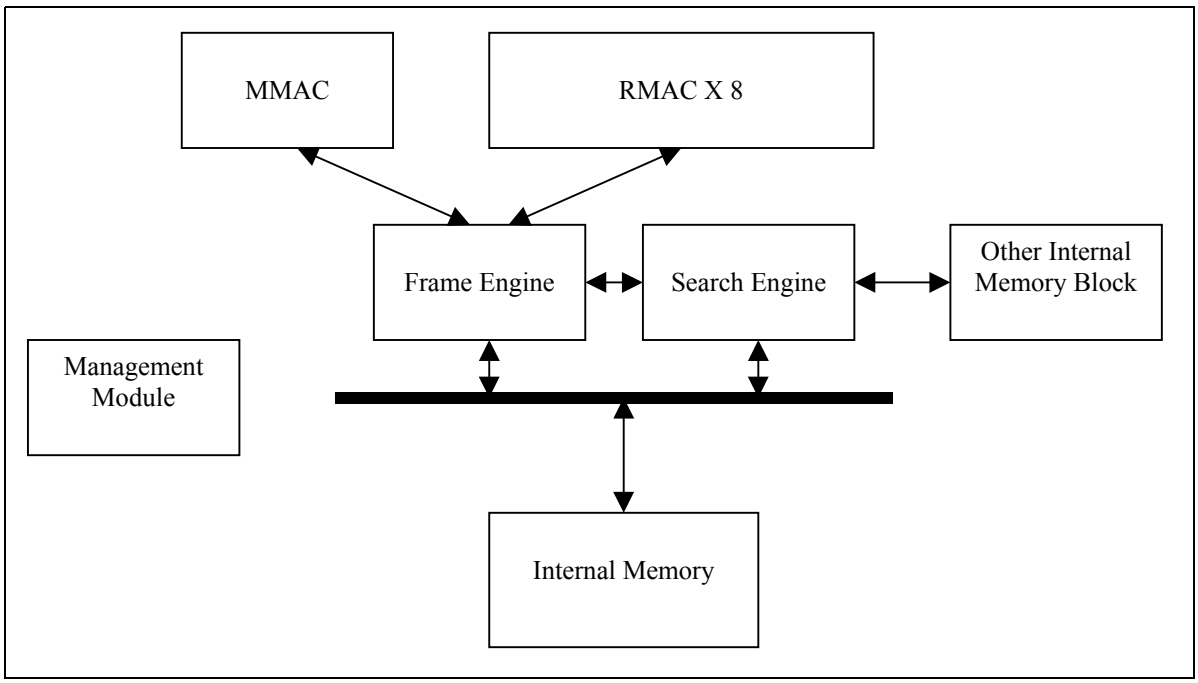


Figure 2 - Functional Block Diagram

2.1 Internal Memory

Two Megabit of internal memory is provided for ethernet Frame Data Buffering (FDB) and for storing of Mac Control Table database (MCT). The MCT is used for storing MAC addresses and their physical port number. The FDB is used for storing the received frame data contents. The contents are stored in this memory until it is ready to be transmitted to the egress port. A memory arbiter is used to arbitrary the memory access requests from various sources. Build in self test is used to detect any error in the memory array when the device is powered up. Build in self test can also be requested by the writing the GCR register.

2.2 MII MAC Module (MMAC)

The MII Media Access Control (MMAC) module provides the necessary buffers and control interface between the Frame Engine (FE) and the external physical device (PHY).

The ZL50409 MMAC implements the MII interface and meets the IEEE 802.3Z specification. It is able to operate in 10M/100M either Half or Full Duplex mode with a back pressure/flow control mechanism. Furthermore, it will automatically retransmit upon collision for up to 16 total transmissions.

This port is denoted as port 9. The PHY address for the PHY device connected to the MMAC port has to be 10h.

2.3 RMII MAC Module (RMAC)

The RMII Media Access Control (RMAC) module provides the necessary buffers and control interface between the Frame Engine (FE) and the external physical device (PHY). It has three interfaces, MII, RMII or GPSI (only for 10M). The RMAC of the ZL50409 device meets the IEEE 802.3 specification. It is able to operate in either Half or Full Duplex mode with a back pressure/flow control mechanism. In addition, it will automatically retransmit upon collision for up to 16 total transmissions.

These eight ports are denoted as ports 0 to 7. The PHY addresses for the PHY devices connected to the 8 RMAC ports has to be from 08h (port 0) to 0Fh (port 7).

2.4 Management Module

The CPU can send a control frame to access or configure the internal network management database. The Management Module decodes the control frame and executes the functions requested by the CPU.

This Module is only active in managed mode. In unmanaged mode, no control frame is accepted by the device.

2.5 Frame Engine

The main function of the frame engine is to forward a frame to its proper destination port or ports. When a frame arrives, the frame engine parses the frame header (64 bytes) and formulates a switching request, sent to the search engine, to resolve the destination port. The arriving frame is moved to the internal memory. After receiving a switch response from the search engine, the frame engine performs transmission scheduling based on the frame's priority. The frame engine forwards the frame to the MAC module when the frame is ready to be sent.

2.6 Search Engine

The Search Engine resolves the frame's destination port or ports according to the destination MAC address (L2) or IP multicast address (IP multicast packet) by searching the database. It also performs MAC learning, priority assignment, and trunking functions.

2.7 Other Internal Memory blocks

Several internal tables are required and are described as follows:

- Network Management (NM) Database - The NM database contains the information in the statistics counters and MIB.
- MAC address Control (MCLT) Link Table - The MCT Link Table stores the linked list of MCT entries that have collisions in the external MAC Table.

2.8 Management and Configuration

One extra port is dedicated to the CPU via the CPU interface module. Two modes this port can operate: managed or unmanaged mode. The different between these modes is tx/rx Ethernet frame and receiving interrupt due to the lack of constant attention or processing power from the CPU. The CPU interface utilizes a 16/8-bit bus in managed mode. It also supports a serial and an I^2C interface, which provides an easy and lower cost way to configure the system for reduced management.

Supported CPU interface modes are

Operation Mode	ISA Interface	Serial	MII	I²C
16-bit CPU	16-bit	NA	NA	NA
8-bit CPU	8-bit	NA	NA	NA
Serial with MII interface	NA	Yes	Yes	No
Serial	NA	Yes	No	No
Unmanaged Serial	NA	Yes	No	Yes

Table 1 - Supported CPU interface modes

1. 16-bit CPU interface similar to the Industry Standard Architecture (ISA) specification.
2. 8-bit CPU interface similar to ISA.
3. Serial with MII. A synchronous serial interface (SSI) bus is used for accessing the configuration register and control frame. MII is used for sending and receiving CPU packets.

4. Serial only. Configuration registers access, Control frame and CPU transmit/receive packets are sent through a synchronous serial interface (SSI) bus.
5. Unmanaged. The ZL50409 can be configured by EEPROM using an I²C interface at bootup, or via a synchronous serial interface (SSI) otherwise. All configuration registers and internal control blocks are accessible by the interface. However, the CPU cannot receive or transmit frames nor will it receive any interrupt information.

The ZL50409 CPU interface provides for easy and effective management of the switching system.

Figure 3 on page 19 provides an overview of the 8/16-bit interface. Figure 4 on page 20 provides an overview of the SSI interface. Figure 5 on page 21 provides an overview of the SSI+MII interface.

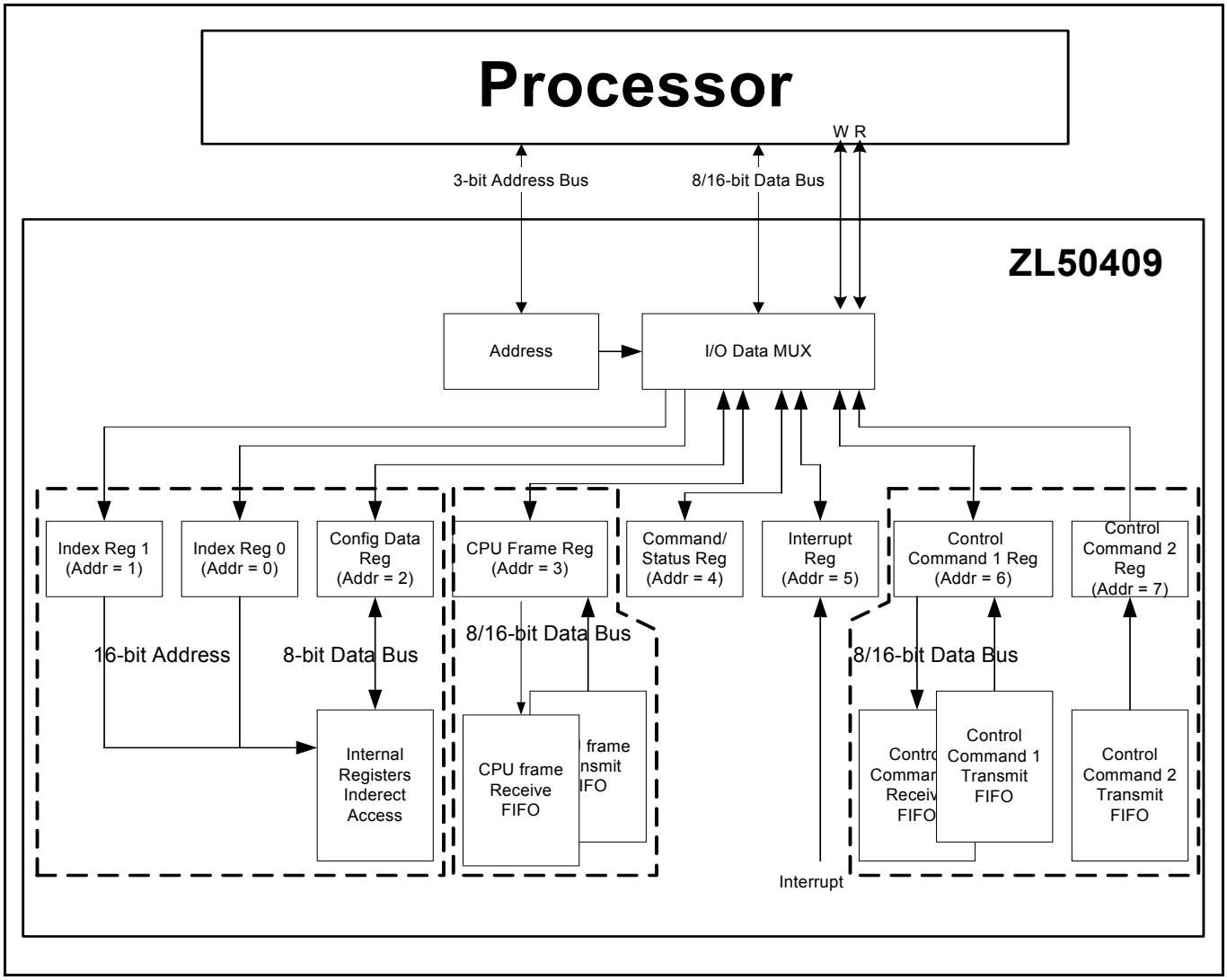


Figure 3 - Overview of the ZL50409 8/16-bit Interface

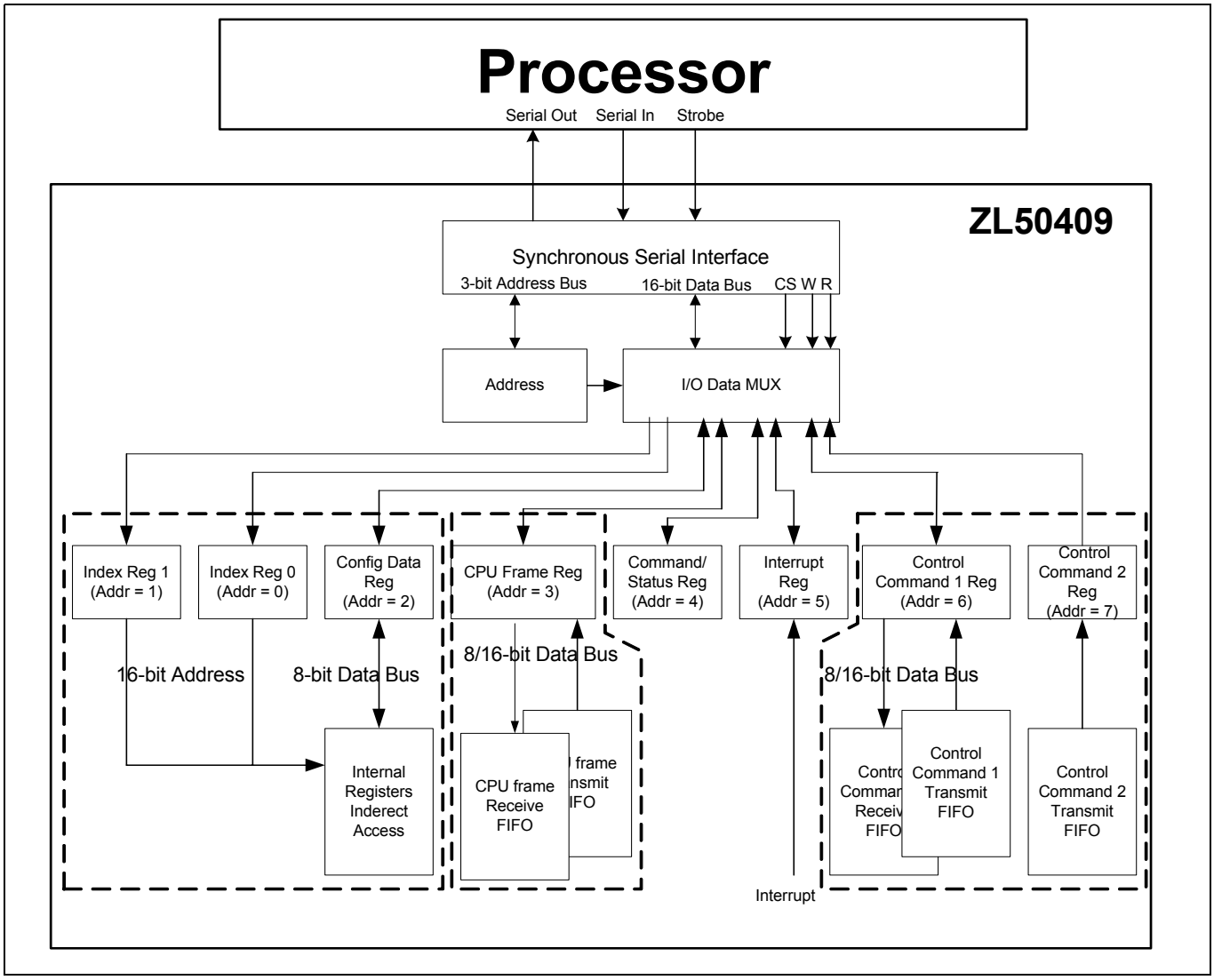


Figure 4 - Overview of the ZL50409 SSI Interface

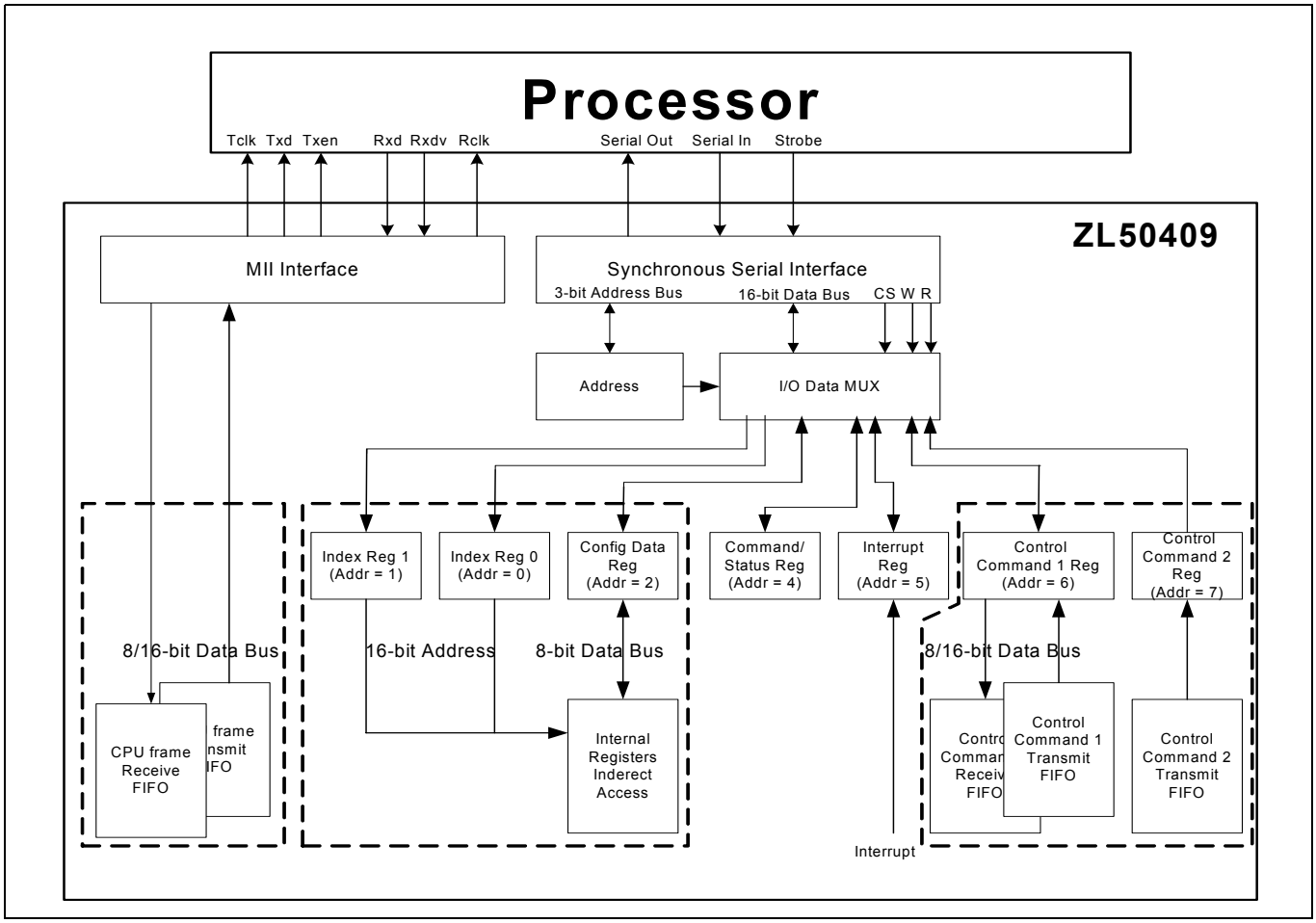


Figure 5 - Overview of the ZL50409 SSI+MII Interface

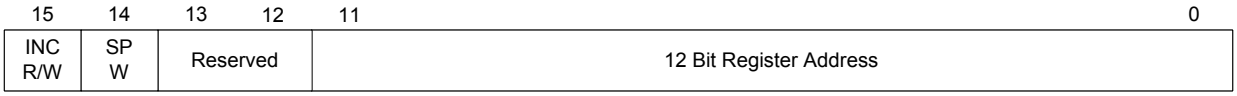
2.9 Register Configuration, Frame Transmission, and Frame Reception

2.9.1 Register Configuration

The ZL50409 has many programmable parameters, covering such functions as QoS weights, VLAN control, and port mirroring setup. In managed mode, the CPU interface provides an easy way of configuring these parameters. The parameters are contained in 8-bit configuration registers. The ZL50409 allows indirect access to these registers, as follows:

- In serial mode, address, command and data are shifted in serially. To access the configuration register, only one “index” registers (addresses 000b) needs to be written with the configuration register address. The desired data can be written into “configure data” register (address 010b). For example, if “XX” is required to be written to register “YY”, a write of “YY” is required to write to address “000b” (Index register). Then, a write of “XX” is required to write to address “010b” (Conig Data Register). This completes the register write and register “YY” will contain the value of “XX”.
- If operating in 8-bit interface mode, two “index” registers (addresses 000b and 001b) need to be written, to indicate the desired 8-bit register address. In 16-bit mode, only one register (address 000b) needs to be written for the desired 16-bit register address.
- To indirectly configure the register addressed by the index register(s), a “configure data” register (address 010b) must be written with the desired 8-bit data.
- Similarly, to read the value in the register addressed by the index register(s), the “configure data” register can now simply be read.

- ZL50409 supports incremental read/write. If CPU requires to read or write to the configuration register incrementally, CPU only has to write to index register once with the MSB of configuration register address set and then CPU can continuously reading or writing to “configure data” register (010b).
- ZL50409 supports special register-write in serial and 16-bit mode. This allows CPU to write to two consecutive configuration registers in a single write operation. By writing to bit[14] of configuration register address, CPU can write 16-bit data to address 010b. Lower 8 bit of data is for the address specified in index register and upper 8 bit of data is for the address + 1. In 8-bit mode, this special feature will be ignored.



In summary, access to the many internal registers is carried out simply by directly accessing only two registers – one register to indicate the index of the desired parameter, and one register to read or write a value. Of course, because there is only one bus master, there can never be any conflict between reading and writing the configuration registers.

2.9.2 Rx/Tx of Standard Ethernet Frames

In serial mode with MII, the MII interface is used for CPU to transmit and receive Ethernet frames. In 8/16-bit or serial only mode, the Ethernet frame is transmitted and received through the CPU interface.

To transmit a frame from the CPU in 8/16-bit or serial only mode:

- The CPU writes a “data frame” register (address 011) with the data it wants to transmit (minimum 64 bytes). After writing all the data, it then writes the frame size, destination port number, and frame status.
- The ZL50409 forwards the Ethernet frame to the desired destination port, no longer distinguishing the fact that the frame originated from the CPU.

To receive a frame into the CPU in 8/16-bit or serial only mode:

- The CPU receives an interrupt when an Ethernet frame is available to be received.
- Frame information arrives first in the data frame register. This includes source port number, frame size, and VLAN tag.
- The actual data follows the frame information. The CPU uses the frame size information to read the frame out.

To transmit a frame from the CPU with MII interface:

- ZL50409 acts as a PHY to provide receive clock (RXCLK) to CPU so the CPU will depend on this receive clock to send packets to ZL50409
- ZL50409 has the ability to halt the receive clock if the receive FIFO of ZL50409 is overflow. Transmitting from CPU to ZL50409 will resume once the receive FIFO of ZL50409 is no longer overflow
- Follow the standard Ethernet transmission format. CPU assert receive data valid (RXDV) before transmitting data to ZL50409 and de-assert RXDV after transmitting the last data

To receive a frame into the CPU with MII interface:

- ZL50409 acts as a PHY to provide transmit clock (TXCLK) to CPU so the CPU will depend on the transmit clock to receive packets from ZL50409
- ZL50409 has the ability to halt the transmit clock if the transmit FIFO of ZL50409 is under-run. CPU will resume receiving packets from ZL50409 once the transmit FIFO of ZL50409 is no longer under-run
- Follow the standard Ethernet transmission format. CPU will see transmit enable (TXEN) be asserted by ZL50409 and CPU can start receiving data. CPU will stop receiving data once TXEN is de-asserted by ZL50409.

In summary, in 8/16-bit or serial only mode, receiving and transmitting frames to and from the CPU is a simple process that uses one direct access register only. In serial mode with MII interface, the CPU will be allowed to transmit and receive frames using standard 802.3 Ethernet transmission format.

2.9.3 Control Frames

In addition to standard Ethernet frames described in the preceding section, the CPU is also called upon to handle special “Control frames,” generated by the ZL50409 and sent to the CPU. These proprietary frames are related to such tasks as statistics collection, MAC address learning, and aging, etc… All Control frames are up to 40 bytes long. Transmitting and receiving these frames is similar to transmitting and receiving Ethernet frames, except that the register accessed is the “Control frame data” register (address 111).

Specifically, there are eleven types of control frames generated by the CPU and sent to the ZL50409:

- Memory read request
- Memory write request
- Learn Unicast MAC address
- Delete Unicast MAC address
- Search Unicast MAC address
- Learn IP Multicast address
- Delete IP Multicast address
- Search IP Multicast address
- Learn Multicast MAC address
- Delete Multicast MAC address
- Search Multicast MAC address

Note: Memory read and write requests by the CPU may include all internal memories which include statistic counters, Mac address control link table and the 2Mbit (256KB) memory block.

In addition, there are nine types of Control frames generated by the ZL50409 and sent to the CPU:

- Interrupt CPU when statistics counter rolls over
- Response to memory read request from CPU
- Learn Unicast MAC address
- Delete Unicast MAC address
- Delete Multicast MAC address
- Delete IP Multicast address
- Response to search Unicast MAC address request from CPU
- Response to search IP Multicast address request from CPU
- Response to search Multicast Mac address request from CPU

The format of the Control Frame is described in the processor interface application note.

2.10 I²C Interface

The I²C interface serves the function of configuring the ZL50409 at boot time. The master is the ZL50409, and the slave is the EEPROM memory.

The I²C interface uses two bus lines, a serial data line (SDA) and a serial clock line (SCL). The SCL line carries the control signals that facilitate the transfer of information from EEPROM to the switch. Data transfer is 8-bit serial and bidirectional, at 50 Kbps. Data transfer is performed between master and slave IC using a request / acknowledgment style of protocol. The master IC generates the timing signals and terminates data transfer. Figure 6 on page 23 depicts the data transfer format. The slave address is the memory address of the EEPROM. Refer to “ZL50409 Register Description” on page 49 for I²C address for each register.

START	SLAVE ADDRESS	R/W	ACK	DATA 1 (8bits)	ACK	DATA 2	ACK	DATA M	ACK	STOP

Figure 6 - Data Transfer Format for I²C Interface

2.10.1 Start Condition

Generated by the master (in our case, the ZL50409). The bus is considered to be busy after the Start condition is generated. The Start condition occurs if while the SCL line is High, there is a High-to-Low transition of the SDA line.

Other than in the Start condition (and Stop condition), the data on the SDA line must be stable during the High period of SCL. The High or Low state of SDA can only change when SCL is Low. In addition, when the I²C bus is free, both lines are High.

2.10.2 Address

The first byte after the Start condition determines which slave the master will select. The slave in our case is the EEPROM. The first seven bits of the first data byte make up the slave address.

2.10.3 Data Direction

The eighth bit in the first byte after the Start condition determines the direction (R/W) of the message. A master transmitter sets this bit to W; a master receiver sets this bit to R.

2.10.4 Acknowledgment

Like all clock pulses, the acknowledgment-related clock pulse is generated by the master. However, the transmitter releases the SDA line (High) during the acknowledgment clock pulse. Furthermore, the receiver must pull down the SDA line during the acknowledge pulse so that it remains stable Low during the High period of this clock pulse. An acknowledgment pulse follows every byte transfer.

If a slave receiver does not acknowledge after any byte, then the master generates a Stop condition and aborts the transfer.

If a master receiver does not acknowledge after any byte, then the slave transmitter must release the SDA line to let the master generate the Stop condition.

2.10.5 Data

After the first byte containing the address, all bytes that follow are data bytes. Each byte must be followed by an acknowledge bit. Data is transferred MSB first.

2.10.6 Stop Condition

Generated by the master. The bus is considered to be free after the Stop condition is generated. The Stop condition occurs if while the SCL line is High, there is a Low-to-High transition of the SDA line.

2.11 Synchronous Serial Interface

The synchronous serial interface (SSI) serves the function of configuring the ZL50409 not at boot time but via a PC. The PC serves as master and the ZL50409 serves as slave. The protocol for the synchronous serial interface is nearly identical to the I²C protocol. The main difference is that there is no acknowledgment bit after each byte of data transferred. Debounce logic on the clock signal (STROBE) can be turned off to speedup command time.

3 ID bits are used to allow up to eight ZL50409 devices to share the same synchronous serial interface. The ID of each device can be setup by bootstrap.

To reduce the number of signals required, the register address, command and data are shifted in serially through the DATAIN pin. STROBE- pin is used as the shift clock. DATAOUT pin is used as data return path.

Each command consists of four parts.

- START pulse
- Register Address
- Read or Write command
- Data to be written or read back

Write operation can be aborted in the middle by sending an ABORT pulse to the ZL50409. Read operation can only be aborted before issuing the read command to the ZL50409.

A START command is detected when DATAIN is sampled high when STROBE- rise and DATAIN is sampled low when STROBE- fall.

An ABORT command is detected when DATAIN is sampled low when STROBE- rise and DATAIN is sampled high when STROBE- fall.

2.11.1 Write Command

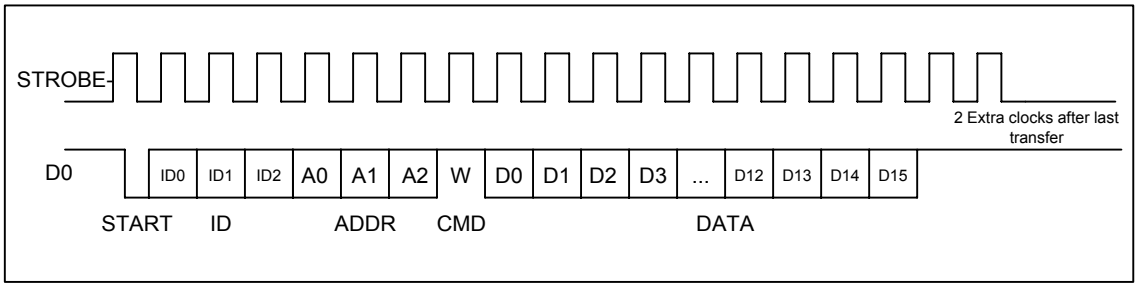


Figure 7 - Serial Interface Write Command Functional Timing

2.11.2 Read Command

All registers in ZL50409 can be modified through this synchronous serial interface.

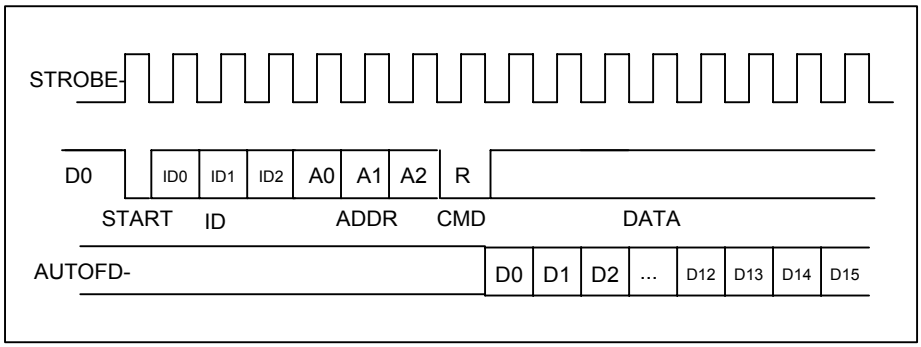


Figure 8 - Serial Interface Read Command Functional Timing

2.12 Timeout Reset Monitor

The ZL50409 supports a state machine monitoring block which can trigger a reset or interrupt if any state machine is determined to be stuck in a non-idle state for more than 5 seconds. This feature is enabled via a bootstrap pin (TSTOUT12). It also requires some register configuration via the CPU interface.

See Programming Timeout Reset application note for more information.

2.13 JTAG

An IEEE1149.1 compliant test interface is provided for boundary scan.

3.0 ZL50409 Data Forwarding Protocol

3.1 Unicast Data Frame Forwarding

When a frame arrives, it is assigned a handle in memory by the Frame Control Buffer Manager (FCB Manager). An FCB handle will always be available, because of advance buffer reservations.

The memory (SRAM) interface is a 64-bit bus, connected to internal memory block. The Receive DMA (RxDMA) is responsible for multiplexing the data and the address. On a port's "turn," the RxDMA will move 8 bytes (or up to the end-of-frame) from the port's associated RxFIFO into memory (Frame Data Buffer, or FDB).

Once an entire frame has been moved to the FDB, and a good end-of-frame (EOF) has been received, the Rx interface makes a switch request. The RxDMA arbitrates among multiple switch requests.

The switch request consists of the first 64 bytes of a frame, containing among other things, the source and destination MAC addresses of the frame. The search engine places a switch response in the switch response queue of the frame engine when done. Among other information, the search engine will have resolved the destination port of the frame and will have determined that the frame is unicast.

After processing the switch response, the Transmission Queue Manager (TxQ manager) of the frame engine is responsible for notifying the destination port that it has a frame to forward. But first, the TxQ manager has to decide whether or not to drop the frame, based on global FDB reservations and usage, as well as TxQ occupancy at the destination. If the frame is not dropped, then the TxQ manager links the frame's FCB to the correct per-port-per-class TxQ. The switch response will come with 8 classified results. The TxQ manager will map this result into the per-port-per-class queue. Unicast TxQ's are linked lists of transmission jobs, represented by their associated frames' FCB's. There is one linked list for each transmission class for each port. There are 2 transmission classes for each of the 8 RMAC ports, and 4 classes for the MMAC port – a total of 24 unicast queues.

The TxQ manager is responsible for scheduling transmission among the queues representing different classes for a port. When the port control module determines that there is room in the MAC Transmission FIFO (TxFIFO) for another frame, it requests the handle of a new frame from the TxQ manager. The TxQ manager chooses among the head-of-line (HOL) frames from the per-class queues for that port, using a Zarlink Semiconductor scheduling algorithm.

The Transmission DMA (TxDMA) is responsible for multiplexing the data and the address. On a port's turn, the TxDMA will move 8 bytes (or up to the EOF) from memory into the port's associated TxFIFO. After reading the EOF, the port control requests a FCB release for that frame. The TxDMA arbitrates among multiple buffer release requests.

The frame is transmitted from the TxFIFO to the line.

3.2 Multicast Data Frame Forwarding

After receiving the switch response, the TxQ manager has to make the dropping decision. A global decision to drop can be made, based on global FDB utilization and reservations. If so, then the FCB is released and the frame is dropped. In addition, a selective decision to drop can be made, based on the TxQ occupancy at some subset of the multicast packet's destinations. If so, then the frame is dropped at some destinations but not others, and the FCB is not released.

If the frame is not dropped at a particular destination port, then the TxQ manager formats an entry in the multicast queue for that port and class. Multicast queues are physical queues (unlike the linked lists for unicast frames). There are 2 multicast queues for each of the 8 RMAC ports. There are 4 multicast queues for the MMAC port. The mapping from the classified result to the priority queue is the same as the unicast traffic. By default, for the RMAC ports to map the 8 transmit priorities into 2 multicast queues, the 2 LSB are discarded. For the MMAC port, to map the 8 transmit priorities into 4 multicast queues, the LSB are discarded. The priority mapping can be modified

through memory configuration command. The multicast queue that is in FIFO format shares the space in the 2M bits internal memory block. The size and starting address can also be programmed through memory configuration command.

During scheduling, the TxQ manager treats the unicast queue and the multicast queue of the same class as one logical queue. The older head of line of the two queues is forwarded first. The port control requests a FCB release only after the EOF for the multicast frame has been read by all ports to which the frame is destined.

3.3 Frame Forwarding To and From CPU

Frame forwarding from the CPU port to a regular transmission port is nearly the same as forwarding between transmission ports. The only difference is that the physical destination port must be indicated in addition to the destination MAC address.

Frame forwarding to the CPU port is nearly the same as forwarding to a regular transmission port. The only difference is in frame scheduling. Instead of using the patent-pending Zarlink Semiconductor scheduling algorithms, scheduling for the CPU port is simply based on strict priority. That is, a frame in a high priority queue will always be transmitted before a frame in a lower priority queue. There are four output queues to the CPU and one receive queue.

4.0 Search Engine

4.1 Search Engine Overview

The ZL50409 search engine is optimized for high throughput searching, with enhanced features to support:

- Up to 4K of Unicast MAC addresses/Multicast MAC addresses and IP Multicast MAC addresses
- Up to 4K VLANs
- 8 groups of port trunking
- Traffic classification into 2 (or 4 for MMAC) transmission priorities, and 2 drop precedence levels
- Packet filtering based on Mac address, Protocol or Logical Port number
- Security
- IP Multicast
- Individual Flooding, Broadcast, Multicast Storm Control
- MAC address learning and aging

4.2 Basic Flow

Shortly after a frame enters the ZL50409 and is written to the Frame Data Buffer (FDB), the frame engine generates a Switch Request, which is sent to the search engine. The switch request consists of the first 64 bytes of the frame, which contain all the necessary information for the search engine to perform its task. When the search engine is done, it writes to the Switch Response Queue, and the frame engine uses the information provided in that queue for scheduling and forwarding.

In performing its task, the search engine extracts and compresses the useful information from the 64-byte switch request. Among the information extracted are the source and destination MAC addresses, the packet's VLAN ID, and whether the frame is unicast or multicast or broadcast. Requests are sent to the SRAM to locate the associated entries in the MCT table.

When all the information has been collected from the SRAM, the search engine has to compare the MAC address on the current entry with the MAC address for which it is searching. If it is not a match, the process is repeated on the internal MCT Table. All MCT entries other than the first of each linked list are maintained internal to the chip. If the desired MAC address is still not found, then the result is either learning (source MAC address unknown) or flooding (destination MAC address unknown).

In addition, VLAN information is used to select the correct set of destination ports for the frame (for multicast), or to verify that the frame's destination port is associated with the VLAN (for unicast).

If the destination MAC address belongs to a port trunk, then the trunk number is retrieved instead of the port number. But on which port of the trunk will the frame be transmitted? This is easily computed using a hash of the source and destination MAC addresses.

When all the information is compiled, the switch response is generated, as stated earlier. The search engine also interacts with the CPU with regard to learning and aging.

4.3 Search, Learning, and Aging

4.3.1 MAC Search

The search block performs source MAC address and destination MAC address (or destination IP address for IP multicast) searching. As we indicated earlier, if a match is not found, then the next entry in the linked list must be examined, and so on until a match is found or the end of the list is reached.

In tag based VLAN mode, if the frame is unicast, and the destination port is not a member of the correct VLAN, then the frame is forwarded to all the members in the VLAN domain; otherwise, the frame is forwarded. If the frame is multicast or broadcast, the frame is forwarded to all the members in the VLAN. Moreover, if port trunking is enabled, this block selects the destination port (among those in the trunk group).

In port based VLAN mode, a bitmap is used to determine whether the frame should be forwarded to the outgoing port. The main difference in this mode is that the bitmap is not dynamic. Ports cannot enter and exit groups because of real-time learning made by a CPU.

The MAC search block is also responsible for updating the source MAC address timestamp used for aging.

4.3.2 Learning

The learning module learns new MAC addresses and performs port change operations on the MCT database. The goal of learning is to update this database as the networking environment changes over time.

When CPU reporting is enabled, learning and port change will be performed when the CPU request queue has room, and a “Learn MAC Address” message is sent to the CPU. When fast learning mode is enabled, learning and port change will be performed when and a latter “Learn MAC Address” message is sent to the CPU when CPU queue has room.

4.3.3 Aging

Aging time is controlled by register 400h and 401h.

The aging module scans and ages MCT entries based on a programmable “age out” time interval. As we indicated earlier, the search module updates the source MAC address timestamps for each frame it processes. When an entry is ready to be aged, the entry is removed from the table, and a “Delete MAC Address” message is sent to inform the CPU.

Supported MAC entry types are: dynamic, static, source filter, destination filter, IP multicast, source and destination filter, secure and multicast MAC address. Only dynamic entries can be aged; all others are static. The MAC entry type is stored in the “status” field of the MCT data structure.

4.4 MAC Address Filtering

The ZL50409's implementation of intelligent traffic switching provides filters for source and destination MAC addresses. This feature filters unnecessary traffic, thereby providing intelligent control over traffic flows and broadcast traffic.

Broadcast, unknown unicast and unknown multicast MAC address can also be filter on per VLAN basis.

MAC address filtering allows the ZL50409 to block an incoming packet to an interface when it sees a specified MAC address in either the source address or destination address of the incoming packet. For example, if your network is

congested because of high utilization from a MAC address, you can filter all traffic transmitted from that address and restore network flow, while you troubleshoot the problem.

4.5 Protocol Filtering

Packet filtering can be performed based on protocol type field in the packets. Up to eight protocols can be programmed to filter or allow packet to pass through the switch.

4.6 Logical Port Filtering

Similar to protocol filtering, if the packet's logical ports match the programmable registers, the packet can be filtered or passed through the switch. Up to eight programmable ports and one ranges can be assigned.

4.7 Quality of Service

Quality of Service (QoS) refers to the ability of a network to provide better service to selected network traffic over various technologies. Primary goals of QoS include dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic), and improved loss characteristics.

Traditional Ethernet networks have had no prioritization of traffic. Without a protocol to prioritize or differentiate traffic, a service level known as "best effort" attempts to get all the packets to their intended destinations with minimum delay; however, there are no guarantees. In a congested network or when a low-performance switch/router is overloaded, "best effort" becomes unsuitable for delay-sensitive traffic and mission-critical data transmission.

The advent of QoS for packet-based systems accommodates the integration of delay-sensitive video and multimedia traffic onto any existing Ethernet network. It also alleviates the congestion issues that have previously plagued such "best effort" networking systems. QoS provides Ethernet networks with the breakthrough technology to prioritize traffic and ensure that a certain transmission will have a guaranteed minimum amount of bandwidth.

Extensive core QoS mechanisms are built into the ZL50409 architecture to ensure policy enforcement and buffering of the ingress port, as well as weighted fair-queue (WFQ) scheduling at the egress port.

In the ZL50409, QoS-based policies sort traffic into a small number of classes and mark the packets accordingly. The QoS identifier provides specific treatment to traffic in different classes, so that different quality of service is provided to each class. Frame and packet scheduling and discarding policies are determined by the class to which the frames and packets belong. For example, the overall service given to frames and packets in the premium class will be better than that given to the standard class; the premium class is expected to experience lower loss rate or delay.

The ZL50409 supports the following QoS techniques:

- In a port-based setup, any station connected to the same physical port of the switch will have the same transmit priority.
- In a tag-based setup, a 3-bit field in the VLAN tag provides the priority of the packet. This priority can be mapped to different queues in the switch to provide QoS.
- In a TOS/DS-based set up, TOS stands for "Type of Service" that may include "minimize delay," "maximize throughput," or "maximize reliability." Network nodes may select routing paths or forwarding behaviours that are suitably engineered to satisfy the service request.
- In a logical port-based set up, a logical port provides the application information of the packet. Certain applications are more sensitive to delays than others; using logical ports to classify packets can help speed up delay sensitive applications, such as VoIP.

4.8 Priority Classification Rule

Figure 9 on page 30 shows the ZL50409 priority classification rule.

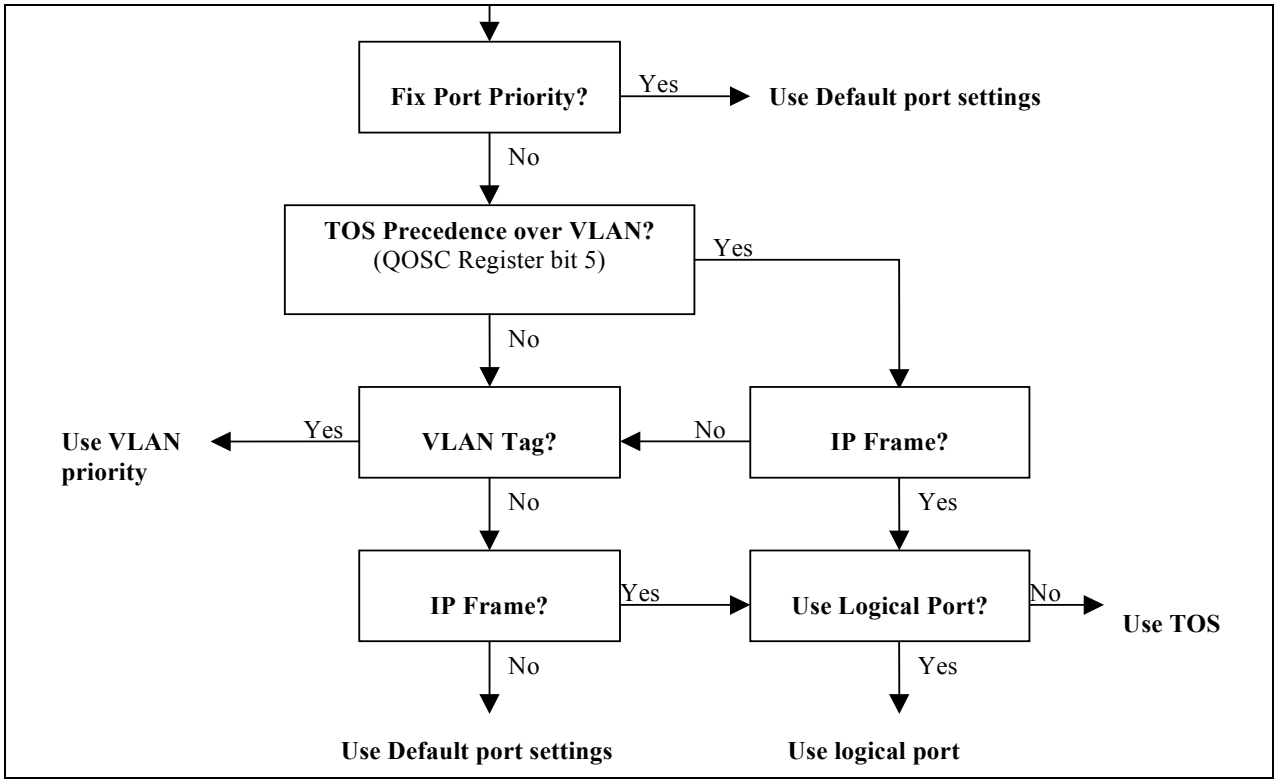


Figure 9 - Priority Classification Rule

4.9 Port and Tag Based VLAN

The ZL50409 supports two models for determining and controlling how a packet gets assigned to a VLAN: port priority and tag -based VLAN.

4.9.1 Port-Based VLAN

An administrator can use the PVMAP Registers to configure the ZL50409 for port-based VLAN (see “Registration Definition” on page 42). For example, ports 1-3 might be assigned to the Marketing VLAN, ports 4-6 to the Engineering VLAN, and ports 7-9 to the Administrative VLAN. The ZL50409 determines the VLAN membership of each packet by noting the port on which it arrives. From there, the ZL50409 determines which outgoing port(s) is/are eligible to transmit each packet, or whether the packet should be discarded.

	Destination Port Numbers Bit Map				
Port Registers	9	…	2	1	0
Register for Port #0 PVMAP00_0[7:0] to PVMAP00_1[1:0]	0		1	1	0
Register for Port #1 PVMAP01_0[7:0] to PVMAP01_1[1:0]	0		1	0	1
Register for Port #2 PVMAP02_0[7:0] to PVMAP02_1[1:0]	0		0	0	0
…					
Register for Port #9 PVMAP09_0[7:0] to PVMAP09_1[1:0]	0		0	0	0

Table 2 - Port-Based VLAN Mapping

For example, in the above table a 1 denotes that an outgoing port is eligible to receive a packet from an incoming port. A 0 (zero) denotes that an outgoing port is not eligible to receive a packet from an incoming port.

In this example:

- Data packets received at port #0 are eligible to be sent to outgoing ports 1 and 2.
- Data packets received at port #1 are eligible to be sent to outgoing ports 0 and 2.
- Data packets received at port #2 are **NOT** eligible to be sent to ports 0 and 1.

4.9.2 Tag-Based VLAN

The ZL50409 supports the IEEE 802.1q specification for “tagging” frames. The specification defines a way to coordinate VLANs across multiple switches. In the specification, an additional 4-octet header (or “tag”) is inserted in a frame after the source MAC address and before the frame type. 12 bits of the tag are used to define the VLAN ID. Packets are then switched through the network with each ZL50409 simply swapping the incoming tag for an appropriate forwarding tag rather than processing each packet's contents to determine the path. This approach minimizes the processing needed once the packet enters the tag-switched network. In addition, coordinating VLAN IDs across multiple switches enables VLANs to extend to multiple switches.

Up to 4K VLANs are supported in the ZL50409. When tag-based VLAN is enabled, each MAC address is learned with it associated VLAN.

5.0 Frame Engine

5.1 Data Forwarding Summary

When a frame enters the device at the RxMAC, the RxDMA will move the data from the MAC RxFIFO to the FDB. Data is moved in 8-byte granules in conjunction with the scheme for the SRAM interface.

A switch request is sent to the Search Engine. The Search Engine processes the switch request.

A switch response is sent back to the Frame Engine and indicates whether the frame is unicast or multicast, and its destination port or ports. On receiving the response, the Frame Engine will check all the QoS related information and decide if this frame can be forwarded.

A Transmission Scheduling Request is sent in the form of a signal notifying the TxQ manager. Upon receiving a Transmission Scheduling Request, the device will format an entry in the appropriate Transmission Scheduling Queue (TxSch Q) or Queues. There are 2 TxSch Q for each RMAC port (and 4 per MMAC port), one for each

priority. Creation of a queue entry either involves linking a new job to the appropriate linked list if unicast, or adding an entry to a physical queue if multicast.

When the port is ready to accept the next frame, the TxQ manager will get the head-of-line (HOL) entry of one of the TxSch Qs, according to the transmission scheduling algorithm (so as to ensure per-class quality of service). (The unicast linked list and the multicast queue for the same port-class pair are treated as one logical queue. The older HOL between the two queues goes first.

The TxDMA will pull frame data from the memory and forward it granule-by-granule to the MAC TxFIFO of the destination port.

5.2 Frame Engine Details

This section briefly describes the functions of each of the modules of the ZL50409 frame engine.

5.2.1 FCB Manager

The FCB manager allocates FCB handles to incoming frames, and releases FCB handles upon frame departure. The FCB manager is also responsible for enforcing buffer reservations and limits that will be used for QoS control and source port flow control. The default values can be determined by referring to Chapter 7. The frame buffer is managed in a 128bytes block unit. During initialization, this block will link all the available blocks in a free buffer list. When each port is ready to receive, this module hands the buffer handle to each requesting port. The FCB manager will also link the released buffer back into the free buffer list.

5.2.2 Rx Interface

The Rx interface is mainly responsible for communicating with the RxMAC. It keeps track of the start and end of frame and frame status (good or bad). Upon receiving an end of frame that is good, the Rx interface makes a switch request.

5.2.3 RxDMA

The RxDMA arbitrates among switch requests from each Rx interface. It also buffers the first 64 bytes of each frame for use by the search engine when the switch request has been made.

5.2.4 TxQ Manager

First, the TxQ manager checks the per-class queue status and global reserved resource situation, and using this information, makes the frame dropping decision after receiving a switch response. The dropping decision includes the head-of-link blocking avoidance if the source port is not flow control enabled. If the decision is not to drop, the TxQ manager links the unicast frame's FCB to the correct per-port-per-class TxQ and updates the FCB information. If multicast, the TxQ manager writes to the multicast queue for that port and class and also update the FCB information including the duplicate count for this multicast frame. The TxQ manager can also trigger source port flow control for the incoming frame's source if that port is flow control enabled. Second, the TxQ manager handles transmission scheduling; it schedules transmission among the queues representing different classes for a port. Once a frame has been scheduled, the TxQ manager reads the FCB information and writes to the correct port control module. The detail of the QoS decision guideline is described in chapter 5.

5.2.5 Port Control

The port control module calculates the SRAM read address for the frame currently being transmitted. It also writes start of frame information and an end of frame flag to the MAC TxFIFO. When transmission is done, the port control module requests that the buffer be released.

5.2.6 TxDMA

The TxDMA multiplexes data and address from port control, and arbitrates among buffer release requests from the port control modules.

6.0 Quality of Service and Flow Control

6.1 Model

Quality of service is an all-encompassing term for which different people have different interpretations. In general, the approach to quality of service described here assumes that we do not know the offered traffic pattern. We also assume that the incoming traffic is not policed or shaped. Furthermore, we assume that the network manager knows his applications, such as voice, file transfer, or web browsing, and their relative importance. The manager can then subdivide the applications into classes and set up a service contract with each. The contract may consist of bandwidth or latency assurances per class. Sometimes it may even reflect an estimate of the traffic mix offered to the switch. As an added bonus, although we do not assume anything about the arrival pattern, if the incoming traffic is policed or shaped, we may be able to provide additional assurances about our switch's performance.

Table 3 shows examples of QoS applications with three transmission priorities, but best effort (P0) traffic may form a fourth class with no bandwidth or latency assurances. MMAC port actually has four total transmission priorities.

Goals	TotalAssured Bandwidth (user defined)	Low Drop Probability (low-drop)	High Drop Probability (high-drop)
Highest transmission priority, P3	50 Mbps	Apps: phone calls, circuit emulation. Latency: < 1 ms. Drop: No drop if P3 not oversubscribed.	Apps: training video. Latency: < 1 ms. Drop: No drop if P3 not oversubscribed; first P3 to drop otherwise.
Middle transmission priority, P2	37.5 Mbps	Apps: interactive apps, Web business. Latency: < 4-5 ms. Drop: No drop if P2 not oversubscribed.	Apps: non-critical interactive apps. Latency: < 4-5 ms. Drop: No drop if P2 not oversubscribed; firstP2 to drop otherwise.
Low transmission priority, P1	12.5 Mbps	Apps: emails, file backups. Latency: < 16 ms desired, but not critical. Drop: No drop if P1 not oversubscribed.	Apps: casual web browsing. Latency: < 16 ms desired, but not critical. Drop: No drop if P1 not oversubscribed; first to drop otherwise.
Total	100 Mbps		

Table 3 - Two-dimensional World Traffic

A class is capable of offering traffic that exceeds the contracted bandwidth. A well-behaved class offers traffic at a rate no greater than the agreed-upon rate. By contrast, a misbehaving class offers traffic that exceeds the agreed-upon rate. A misbehaving class is formed from an aggregation of misbehaving microflows. To achieve high link utilization, a misbehaving class is allowed to use any idle bandwidth. However, such leniency must not degrade the quality of service (QoS) received by well-behaved classes.

As Table 3 illustrates, the six traffic types may each have their own distinct properties and applications. As shown, classes may receive bandwidth assurances or latency bounds. In the table, P3, the highest transmission class, requires that all frames be transmitted within 1 ms, and receives 50% of the 100 Mbps of bandwidth at that port.

Best-effort (P0) traffic forms a fourth class that only receives bandwidth when none of the other classes have any traffic to offer. It is also possible to add a fourth class that has strict priority over the other three; if this class has even one frame to transmit, then it goes first. In the ZL50409, each RMAC port will support two total classes, and the MMAC port will support four classes. We will discuss the various modes of scheduling these classes in the next section.

In addition, each transmission class has two subclasses, high-drop and low-drop. Well-behaved users should rarely lose packets. But poorly behaved users–users who send frames at too high a rate – will encounter frame loss, and the first to be discarded will be high-drop. Of course, if this is insufficient to resolve the congestion, eventually some low-drop frames are dropped, and then all frames in the worst case.

Table 3 shows that different types of applications may be placed in different boxes in the traffic table. For example, casual web browsing fits into the category of high-loss, high-latency-tolerant traffic, whereas VoIP fits into the category of low-loss, low-latency traffic.

6.2 Two QoS Configurations

There are two basic pieces to QoS scheduling in the MMAC port of ZL50409: strict priority (SP) or weighted fair queuing (WFQ). The only configuration for a RMAC port is strict priority between the two queues.

6.2.1 Strict Priority

When strict priority is part of the scheduling algorithm, if a queue has any frame to transmit, it goes first. For RMAC ports, this is an easy way to provide the different service. For all recognizable traffic, the bandwidth is guaranteed to 100% of the line rate. This scheme works as long as the overall high priority bandwidth is not over the line rate and the latency on all the low priority traffic is don't care. The strict priority queue in the MMAC port is similar to RMAC ports other than having 4 queues instead of 2 queues. The priority queue P0 can be scheduled only if the priority queue P1 is empty, so as to priority queues P2 and P3. The lowest priority queue is treated as best effort queue.

Because we do not provide any assurances for best effort traffic, we do not enforce latency by dropping best effort traffic. Furthermore, because we assume that strict priority traffic is carefully controlled before entering the ZL50409, we do not enforce a fair bandwidth partition by dropping strict priority traffic. To summarize, dropping to enforce bandwidth or delay does not apply to strict priority or best effort queues. We only drop frames from best effort and strict priority queues when queue size is too long or global / class buffer resources become scarce.

6.2.2 Weighted Fair Queuing

In some environments – for example, in an environment in which delay assurances are not required, but precise bandwidth partitioning on small time scales is essential, WFQ may be preferable to a strict assurance scheduling discipline. The ZL50409 provides this kind of scheduling algorithm on MMAC port only. The user sets four WFQ "weights" such that all weights are whole numbers and sum to 64. This provides per-class bandwidth partitioning with granular within 2%.

In WFQ mode, though we do not assure frame latency, the ZL50409 still retains a set of dropping rules that helps to prevent congestion and trigger higher level protocol end-to-end flow control.

6.3 WRED Drop Threshold Management Support

To avoid congestion, the Weighted Random Early Detection (WRED) logic drops packets according to specified parameters. The following table summarizes the behavior of the WRED logic.

	Px > WRED_L1	Px > WRED_L2	BM Reject
High Drop	X%	100%	100%
Low Drop	Y%	Z%	100%

Table 4 - WRED Logic Behaviour

Px is the total byte count, in the priority queue x, can be the strict priority queue of RMAC ports and higher 3 priority queues for MMAC port. The WRED logic has two drop levels, depending on the value of Px. Each drop level has defined high-drop and low-drop percentages, which indicate the minimum and maximum percentages of the data that can be discarded. The X, Y Z percent can be programmed by the register RDRC0, RDRC1. All packets will be dropped only if the system runs out of the specific buffer resource, per class buffer or per source port buffer. The WRED thresholds of each queue can be programmed by the QOS control registers (refer to the register group 8). See Programming QoS Registers application note for more information.

6.4 Shaper

Although traffic shaping is not a primary function of the ZL50409, the chip does implement a shaper for every queue in the MMAC port. Our goal in shaping is to control the average rate of traffic exiting the ZL50409. If shaper is enabled, strict priority will be applied to that queue. The priority between two shaped queue is the same as in strict priority scheduling.

Traffic rate is set using a programmable whole number, no greater than 64. For example, if the setting is 32, then the traffic rate transmit out of the shaped queue is 32/64 * 100 Mbps = 50 Mbps. See Programming QoS Register application note for more information.

Also, when shaping is enabled, it is possible for a queue to explode in length if fed by a greedy source. The reason is that a shaper is by definition not work-conserving; that is, it may hold back from sending a packet even if the line is idle. Though we do have global resource management, we do nothing other than per port WRED to prevent this situation locally. We assume the traffic is policed at a prior stage to the ZL50409 or WRED dropping is fine and shall restrain this situation.

6.5 Rate Control

The ZL50409 provides a rate control function on its RMAC ports. This rate control function applies to both the incoming and outgoing traffic aggregate on each RMAC port. It provides a way of reducing the average rate below full wire speed. Note that the rate control function does not shape or manipulate any particular traffic class. Furthermore, though the average rate of the port can be controlled with this function, the peak rate will still be full line rate.

Two principal parameters are used to control the average rate for a RMAC port. A port's rate is controlled by allowing, on average, M bytes to be transmitted every N microseconds. Both of these values are programmable. The user can program the number of bytes in 8-byte increments, and the time may be set in units of 10us or 1ms.

The value of M/N will, of course, equal the average data rate of the traffic aggregate on the given RMAC port. Although there are many (M,N) pairs that will provide the same average data rate performance, the smaller the time interval N, the "smoother" the output pattern will appear.

In addition to controlling the average data rate on a RMAC port, the rate control function also manages the maximum burst size at wire speed. The maximum burst size can be considered the memory of the rate control mechanism; if the line has been idle for a long time, to what extent can the port "make up for lost time" by transmitting a large burst? This value is also programmable, measured in 8-byte increments.

Example: Suppose that the user wants to restrict Fast Ethernet port P's average departure rate to 32 Mbps – 32% of line rate – when the average is taken over a period of 10 ms. In an interval of 10 ms, exactly 40000 bytes can be transmitted at an average rate of 32 Mbps.

So how do we set the parameters? The rate control parameters are contained in an internal RAM block accessible through the CPU port (See Programming QoS Registers application note and Processor interface application note). The data format is shown below.

63:40	39:32	31:16	15:0
0	Time interval	Maximum burst size	Number of bytes

As we indicated earlier, the number of bytes is measured in 8-byte increments, so the 16-bit field “Number of bytes” should be set to 40000/8, or 5000. In addition, the time interval has to be set to 10 in units of 1 ms. Though we want the average data rate on port P to be 32 Mbps when measured over an interval of 10 ms, we can also adjust the maximum number of bytes that can be transmitted at full line rate in any single burst. Suppose we wish this limit to be 12 kilobytes. The number of bytes is measured in 8-byte increments, so the 16-bit field “Maximum burst size” is set to 12000/8, or 1500.

The action on the incoming traffic and outgoing traffic when credit is not available are different. For the outgoing traffic, the queued frames will be held in the queue until the credit become available. The consequence of this holding is the exploding queue size that may cause dropping on the receiving side. The capability of ZL50409 on this perspective is quite limited due to the small frame buffer on chip. The actions on the incoming traffic depending on the flow control state of that port. If the ingress port flow control is turned on, the XOFF flow control will be triggered when the credit is running lower than half of the maximum burst size. The XON will be triggered when the available credit is increased to above the threshold. If the port flow control is disabled, the received traffic will subject to WRED depending on the credit availability. If the none of the credit is available, all received frame will be dropped. If only a quarter of maximum burst credits are available, the frame that been marked as high drop will be drop 100%, the low drop frame will be dropped at ra%, If half of the maximum burst credits are available, high drop frame will be dropped at rb%. The ra% and rb% can be programmed by RDRC2 register.

6.6 Buffer Management

Because the number of FDB slots is a scarce resource, and because we want to ensure that one misbehaving source port or class cannot harm the performance of a well-behaved source port or class, we introduce the concept of buffer management into the ZL50409. Our buffer management scheme is designed to divide the total buffer space into numerous reserved regions and one shared pool, as shown in Figure 10 on page 37.

As shown in the figure, the FDB pool is divided into several parts. A reserved region for temporary frames stores frames prior to receiving a switch response. Such a temporary region is necessary, because when the frame first enters the ZL50409, its destination port and class are as yet unknown, and so the decision to drop or not needs to be temporarily postponed. This ensures that every frame can be received first before subjecting them to the frame drop discipline after classifying.

Six reserved sections, one for each of the first six priority classes, ensure a programmable number of FDB slots per class. The lowest two classes do not receive any buffer reservation. Furthermore, a frame is stored in the region of the FDB corresponding to its class. As we have indicated, the eight classes use only four transmission scheduling queues for RMAC ports, but as far as buffer usage is concerned, there are still eight distinguishable classes.

Another segment of the FDB reserves space for each of the 10 ports — 9 ports for Ethernet and one CPU port (port number 8). Two parameters can be set, one for the source port reservation for RMAC ports and CPU port, and one for the source port reservation for the MMAC port. These 10 reserved regions make sure that no well-behaved source port can be blocked by another misbehaving source port.

In addition, there is a shared pool, which can store any type of frame. The frame engine allocates the frames first in the six priority sections. When the priority section is full or the packet has priority 1 or 0, the frame is allocated in the shared pool. Once the shared pool is full the frames are allocated in the section reserved for the source port.

The following registers define the size of each section of the Frame data Buffer:

- PR100- Port Reservation for RMAC Ports
- PRM- Port Reservation for MMAC Port
- SFCB- Share FCB Size
- C1RS- Class 1 Reserve Size
- C2RS- Class 2 Reserve Size
- C3RS- Class 3 Reserve Size

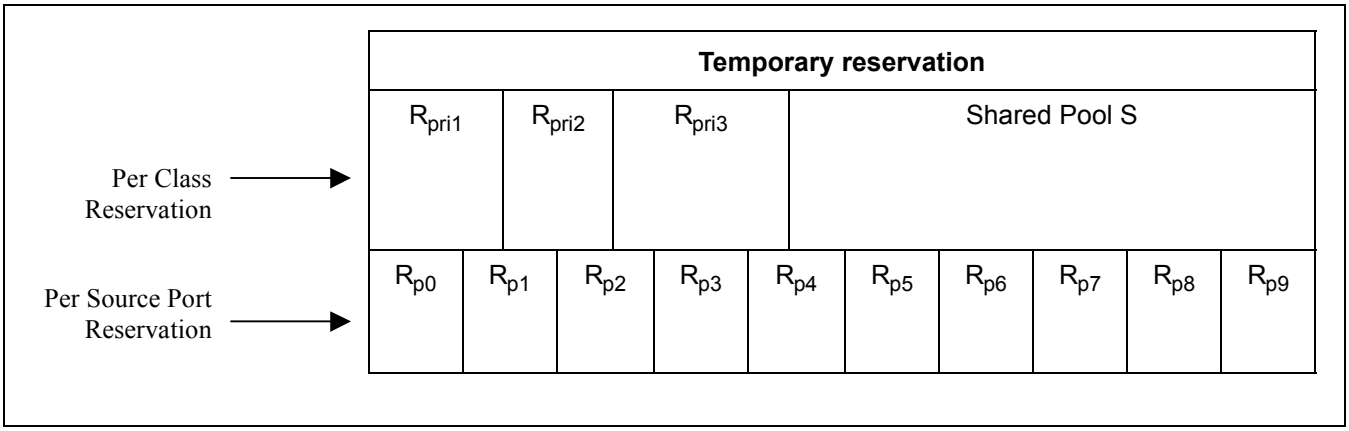


Figure 10 - Buffer Partition Scheme Used to Implement Buffer Management in the ZL50409

6.6.1 Dropping When Buffers Are Scarce

As already discussed, the WRED mechanism may drop frames on output queue status. In addition to these reasons for dropping, we also drop frames when global buffer space becomes scarce. The function of buffer management is to make sure that such dropping causes as little blocking as possible. If a received frame is dispatched to the best effort queue, the buffer management will check on the overall buffer situation plus the output queue status to decide the frame drop condition. If the source port has not enough buffer for it, the frame will be dropped. If the output queue reach the UCC (unicast congest control) and the shared buffer has run out, the frame will be dropped by b%. If the output queue reach the UCC and the source port reservation is lower than the buffer low threshold, the frame will be dropped. All the dropping functions are disabled if the source port is flow control capable.

6.7 ZL50409 Flow Control Basics

Because frame loss is unacceptable for some applications, the ZL50409 provides a flow control option. When flow control is enabled, scarcity of source port buffer space may trigger a flow control signal; this signal tells a source port sending a packet to this switch, to temporarily hold off.

While flow control offers the clear benefit of no packet loss, it also introduces a problem for quality of service. When a source port receives an Ethernet flow control signal, all microflows originating at that port, well-behaved or not, are halted. A single packet destined for a congested output can block other packets destined for un-congested outputs. The resulting head-of-line blocking phenomenon means that quality of service cannot be assured with high confidence when flow control is enabled.

On the other hand, the ZL50409 will still prioritize the received frame disregarding the outgoing port flow control capability. If a frame is classified as high priority, it is still subjected to the WRED, which means the no-loss on the high priority queue is not guaranteed. To resolve this situation, the user may set the output port WRED threshold so high that may never be reached, or program the priority mapping table in the queue manager to map all the traffic to best effort queue on the flow control capable port. The first method has side impact on the global resource management since the port may hold too much per class resource that is scarce in the system. The second method, by nature, lost the benefit of prioritization.

6.7.1 Unicast Flow Control

For unicast frames, flow control is triggered by source port resource availability. Recall that the ZL50409's buffer management scheme allocates a reserved number of FDB slots for each source port. If a programmed number of a source port's reserved FDB slots have been used, then flow control Xoff is triggered.

Xon is triggered when a port is currently being flow controlled, and all of that port's reserved FDB slots have been released.

Note that the ZL50409's per-source-port FDB reservations assure that a source port that sends a single frame to a congested destination will not be flow controlled.

6.7.2 Multicast Flow Control

Flow control for multicast frames is triggered by a global buffer counter. When the system exceeds a programmable threshold of multicast packets, Xoff is triggered. Xon is triggered when the system returns below this threshold.

Note: If per-port flow control is on, QoS performance will be affected.

6.8 Mapping to IETF Diffserv Classes

The mapping between priority classes discussed in this chapter and elsewhere is shown below.

ZL50409	P3	P2	P1	P0
IETF	NM+EF	AF0	AF1	BE0

Table 5 - Mapping between ZL50409 and IETF Diffserv Classes for MMAC Port

As the table illustrates, the classes of Table 5 are merged in pairs—one class corresponding to NM+EF, two AF classes, and a single BE class.

For RMAC ports, the classes of Table 6 are merged in pairs—one class corresponding to NM+EF+AF1, AF0+ BE class.

ZL50409	P1	P1	P0	P0
IETF	NM+EF	AF0	AF1	BE0

Table 6 - Mapping between ZL50409 and IETF Diffserv Classes for RMAC Ports

Features of the ZL50409 that correspond to the requirements of their associated IETF classes are summarized in the table below.

Network management (NM) and Expedited forwarding (EF)	Global buffer reservation for NM and EF Shaper for EF traffic on MMAC port Option of strict priority scheduling No dropping if admission controlled
Assured forwarding (AF)	Four AF classes for MMAC port Programmable bandwidth partition, with option of WFQ service Option of delay-bounded service keeps delay under fixed levels even if not admission-controlled Random early discard, with programmable levels Global buffer reservation for each AF class
Best effort (BE)	Two BE classes for MMAC port Service only when other queues are idle means that QoS not adversely affected Random early discard, with programmable levels Traffic from flow control enabled ports automatically classified as BE

Table 7 - ZL50409 Features Enabling IETF Diffserv Standards

6.9 Failover Backplane Feature

The ZL50409 implements a hardware assisted link failure detection mechanism utilizing a Link Heart Beat (LHB) packet. The LHB packet format is defined as a 64-byte MAC control frame with a user defined opcode. The packet format is illustrated below:

01-80-c2-00-00-01	xx-xx-xx-xx-xx-xx	88-08	yy-yy	00-00-...	CRC

Where “xx-xx-...” is the source port MAC address and “yy-yy” is the special opcode defined by register setup (LHBReg0,1). The opcode “00-01” is reserved for the flow control packet.

The LHB is done between two compatible MACs providing this function. A timer parameter will be set for both the receiver and transmitter (LHBTimer).

On the transmission side, the MAC will monitor the transmission activities. If there is no activity for more than the set period, a LHB packet will be sent to its link partner. Therefore, there should always be at least one packet transmitted from the MAC for every period specified.

On the receiving side, the MAC will also monitor the activity. If there is no good packet received for more than 2X the set period, an alarm will be raised to the CPU. The LHB packet is only used by the ZL50409 to reset the timeout counter, it is ignored otherwise (i.e. not passed on within the system).

See the Link Heart Beat Application Note for more information.

7.0 Port Trunking

7.1 Features and Restrictions

A port group (i.e. trunk) can include up to 8 physical ports. There are eight trunk groups total.

Load distribution among the ports in a trunk for unicast is performed using hashing based on source MAC address and destination MAC address. Three other options include source MAC address only, destination MAC address only, and source port (in bidirectional ring mode only). Load distribution for multicast is performed similarly.

If a VLAN includes any of the ports in a trunk group, all the ports in that trunk group should be in the same VLAN member map.

The ZL50409 also provides a safe fail-over mode for port trunking automatically. If one of the ports in the trunking group goes down, the ZL50409 can redistribute the traffic over to the remaining ports in the trunk with software assistance.

7.2 Unicast Packet Forwarding

The search engine finds the destination MCT entry, and if the status field says that the destination port found belongs to a trunk, then the trunk group number is retrieved.

The source port of the packet is checked against the destination trunk group. If the source port belongs to the destination trunk group, the packet is discarded.

A hash key, based on some combination of the source and destination MAC addresses for the current packet, selects the appropriate forwarding port, as specified in the Trunk_Hash registers. Each trunk has eight trunk_hash registers which selects one of the potential eight outgoing ports. The hash key provides a pseudo flow identifier which force the same flow to the same destination flow. As a result, the packet will always arrive in order.

7.3 Multicast Packet Forwarding

For multicast packet forwarding, the device must determine the proper set of ports from which to transmit the packet based on the VLAN and hash key.

Three functions are required in order to distribute multicast packets to the appropriate destination ports in a port trunking environment.

- Determining the VLAN group it is forwarding port per group.
- The source port/group must be excluded from the forwarding.
- Select one port per trunk group to forward the packet to. This selection is based on hash key described in previous section.

For example, port 0,1 and 2 belong to trunk group 0 and port 3 and 4 belong to trunk group 1. A single VLAN is established in this system with port 0,1,2,3,4,5 and 6 as the members in the VLAN. When a multicast packet is sent in from port 3, the ZL50409 select port 0,1,2,3,4,5 and 6 as potential destination based on the VLAN. Then port 3 and 4 are removed because they belong to the source port group (trunk group 1). Two ports from trunk group 0 will be removed based on the hash key. In this example, we assume port 0 and 1 are removed. As a result, port 2,5 and 6 are the only outgoing ports for this multicast packet.

8.0 Port Mirroring

8.1 Mirroring Features

Packets can be mirrored (duplicated) for network monitor purpose and/or network debug purpose. Three types of mirroring is available in ZL50409.

1. Source or Destination Mac address based
2. Flow based
3. Port based

In source or destination mac address based mirroring, the “M” bit of the mirroring MAC address in the MCT is set. Also, the user need to specify the mirroring MAC address is source or destination of the packet. If source is selected, any packet received with the mirroring MAC address as source MAC address will be copied to the mirrored port. In the same way, if destination is selected, any packet received with mirroring MAC address as destination MAC address will be copied to the mirrored port.

In flow based mirroring, a flow is established based on the source and destination mac address pair. When enabled, a packet with source and destination address match the pre-programmed source and destination mac address pair will be copied to the mirrored port. In reverse direction (source and destination match pre programmed destination and source), the flow can also be enabled and the frame will be copied to the mirrored port.

In port based mirroring, traffic from any RMAC port can be mirrored to any RMAC port. The traffic from the source port can be either ingress or egress traffic. Up to two ports can be setup as mirrored ports. As a result, the traffic (both ingress and egress) of a specific port can be monitored by setting up both mirrored ports. Once a port is setup as mirrored port, it cannot be used for regular traffic.

The mirrored port can be any port in the ZL50409.

8.2 Using port mirroring for loop back

To perform remote loop back test, port mirroring can be used to bounce back the packet to the source port to check the data path.

The CPU needs to setup the remote device through the command channel to enable port mirroring in the remote device. A CPU packet is send to the port in test in Device A. The packet will be forwarded to the test port, external cable, the destination port in Device B, and loop back to itself, back to the cable and go back to Device A and the CPU. This way, the whole channel can be tested.

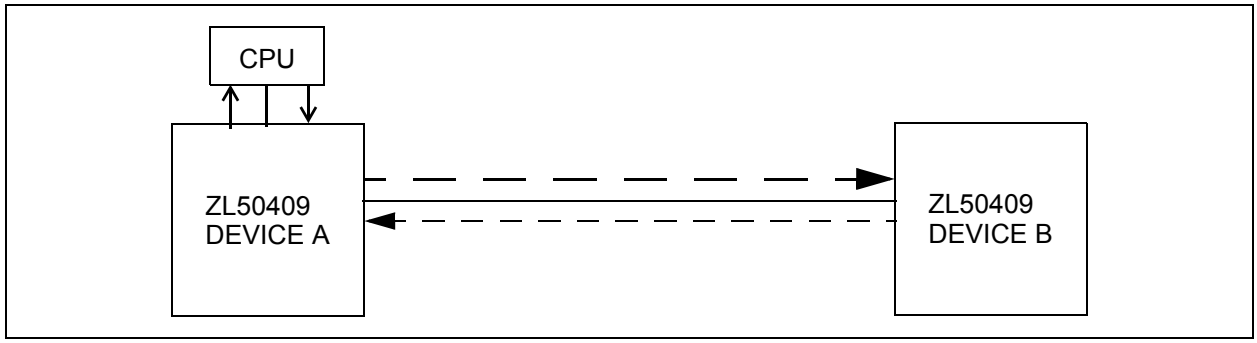


Figure 11 - Remote Loopback Test

9.0 GPSI (7WS) Interface

9.1 GPSI connection

The RMAC ethernet port can function in GPSI (7WS) mode. In this mode, the TXD[0], RXD[0] serve as TX data, RX data and respectively. The link and duplex of the port can be controlled by programming the ECR register. Only port-based VLAN is supported with GPSI interface.

10.0 Clock Speed Requirements

10.1 System Clock (SCLK) speed requirement

SCLK is the primary clock for the ZL50409 device. The speed requirement is based on the system configuration. Below is a table for a few configuration.

Configuration	SCLK speed
9 port 10/100	50Mhz
8 port 10/100	45Mhz

Table 8 - SCLK Speed Requirements

10.2 RMAC Reference Clock (M_CLK) speed requirement

M_CLK is a 50MHz clock used for the RMAC ports (ports 0-7).

If none of the RMAC ports are configured in RMII mode, a different clock frequency can be applied to M_CLK, as long as it's less than 50MHz. In this case, register USD must be set to provide an internal 1usec timing.

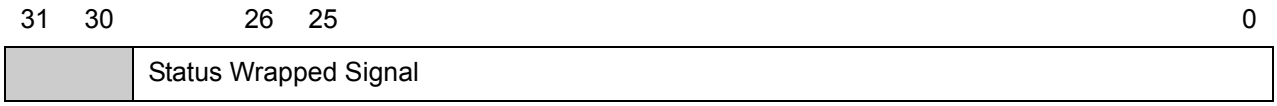
10.3 MMAC Reference Clock (REF_CLK) speed requirement

REF_CLK must be connected to SCLK for the MMAC port (port 9).

11.0 Hardware Statistics Counter

11.1 Hardware Statistics Counters List

ZL50409 hardware provides a full set of statistics counters for each Ethernet port. The CPU accesses these counters through the CPU interface. All hardware counters are rollover counters. When a counter rolls over, the CPU is interrupted, so that long-term statistics may be kept. The MAC detects all statistics, except for the delay exceed discard counter (detected by buffer manager) and the filtering counter (detected by queue manager). The following is the wrapped signal sent to the CPU through the command block.



B[0]	0-d	Bytes Sent (D)
B[1]	1-L	Unicast Frame Sent
B[2]	1-U	Frame Send Fail
B[3]	2-l	Flow Control Frames Sent
B[4]	2-u	Non-Unicast Frames Sent
B[5]	3-d	Bytes Received (Good and Bad) (D)
B[6]	4-d	Frames Received (Good and Bad) (D)
B[7]	5-d	Total Bytes Received (D)
B[8]	6-L	Total Frames Received
B9]	6-U	Flow Control Frames Received
B[10]	7-l	Multicast Frames Received
B[11]	7-u	Broadcast Frames Received
B[12]	8-L	Frames with Length of 64 Bytes
B[13]	8-U	Jabber Frames
B[14]	9-L	Frames with Length Between 65-127 Bytes
B[15]	9-U	Oversize Frames
B[16]	A-l	Frames with Length Between 128-255 Bytes
B[17]	A-u	Frames with Length Between 256-511 Bytes
B[18]	B-l	Frames with Length Between 512-1023 Bytes
B[19]	B-u	Frames with Length Between 1024-1528 Bytes

B[20]	C-l	Fragments
B[21]	C-U1	Alignment Error
B[22]	C-U	Undersize Frames
B[23]	D-l	CRC
B[24]	D-u	Short Event
B[25]	E-l	Collision
B[26]	E-u	Drop
B[27]	F-l	Filtering Counter
B[28]	F-U1	Delay Exceed Discard Counter
B[29]	F-U	Late Collision

Notation: X-Y

X: Address in the contain memory

Y: Size and bits for the counter

d: D Word counter

L: 24 bits counter bit[23:0]

U: 8 bits counter bit[31:24]

U1: 8 bits counter bit[23:16]

l: 16 bits counter bit[15:0]

u: 16 bits counter bit[31:16]

11.2 IEEE 802.3 HUB Management (RFC 1516)

11.2.1 Event Counters

11.2.1.1 READABLEOCTET

Counts number of bytes (i.e. octets) contained in good valid frames received.

Frame size: $\geq$ 64 bytes, $\leq$ 1522 bytes if VLAN Tagged;
1518 bytes if not VLAN Tagged

No FCS (i.e. checksum) error

No collisions

11.2.1.2 READABLEFRAME

Counts number of good valid frames received.

Frame size:	≥ 64 bytes,	≤ 1522 bytes if VLAN Tagged;
		1518 bytes if not VLAN Tagged
No FCS error		
No collisions		

11.2.1.3 FCSERRORS

Counts number of valid frames received with bad FCS.

Frame size:	≥ 64 bytes,	≤ 1522 bytes if VLAN Tagged;
		1518 bytes if not VLAN Tagged
No framing error		
No collisions		

11.2.1.4 ALIGNMENTERRORS

Counts number of valid frames received with bad alignment (not byte-aligned).

Frame size:	≥ 64 bytes,	≤ 1522 bytes if VLAN Tagged;
		1518 bytes if not VLAN Tagged
No framing error		
No collisions		

11.2.1.5 FrameTooLongs

Counts number of frames received with size exceeding the maximum allowable frame size.

Frame size:	≥ 64 bytes,	> 1522 bytes if VLAN Tagged;
		1518 bytes if not VLAN Tagged
FCS error:	don’t care	
Framing error:	don’t care	
No collisions		

11.2.1.6 ShortEvents

Counts number of frames received with size less than the length of a short event.

Frame size:	≥ 64 bytes,	< 10 bytes
FCS error:	don't care	
Framing error:	don't care	
No collisions		

11.2.1.7 Runts

Counts number of frames received with size under 64 bytes, but greater than the length of a short event.

Frame size:	≥ 10 bytes,	< 64 bytes
FCS error:	don't care	
Framing error:	don't care	
No collisions		

11.2.1.8 COLLISIONS

Counts number of collision events.

Frame size:	any size

11.2.1.9 LATEEVENTS

Counts number of collision events that occurred late (after LateEventThreshold = 64 bytes).

Frame size:	any size
Events are also counted by collision counter	

11.2.1.10 VeryLongEvents

Counts number of frames received with size larger than Jabber Lockup Protection Timer (TW3).

Frame size:	> Jabber

11.2.1.11 DATARATEMISATCHES

For repeaters or HUB application only.

11.2.1.12 AUTOPARTITIONS

For repeaters or HUB application only.

11.2.1.13 TotalErrors

Sum of the following errors:

FCS errors

Alignment errors

Frame too long

Short events

Late events

Very long events

11.3 IEEE – 802.1 Bridge Management (RFC 1286)

11.3.1 Event Counters

11.3.1.1 INFRAMES

Counts number of frames received by this port or segment.

Note: A frame received by this port is only counted by this counter if and only if it is for a protocol being processed by the local bridge function.

11.3.1.2 OUTFRAMES

Counts number of frames transmitted by this port.

Note: A frame transmitted by this port is only counted by this counter if and only if it is for a protocol being processed by the local bridge function.

11.3.1.3 INDISCARDS

Counts number of valid frames received which were discarded (i.e., filtered) by the forwarding process.

11.3.1.4 DELAYEXCEEDEDDISCARDS

Counts number of frames discarded due to excessive transmit delay through the bridge.

11.3.1.5 MTUEXCEEDEDDISCARDS

Counts number of frames discarded due to excessive size.

11.4 RMON – Ethernet Statistic Group (RFC 1757)

11.4.1 Event Counters

11.4.1.1 DROP EVENTS

Counts number of times a packet is dropped, because of lack of available resources. DOES NOT include all packet dropping -- for example, random early drop for quality of service support.

11.4.1.2 OCTETS

Counts the total number of octets (i.e. bytes) in any frames received.

11.4.1.3 BROADCASTPKTS

Counts the number of good frames received and forwarded with broadcast address.

Does not include non-broadcast multicast frames.

11.4.1.4 MULTICASTPKTS

Counts the number of good frames received and forwarded with multicast address.

Does not include broadcast frames.

11.4.1.5 CRCAlignErrors

Frame size:	≥ 64 bytes,	< 1522 bytes if VLAN tag (1518 if no VLAN)
No collisions:		

Counts number of frames received with FCS or alignment errors

11.4.1.6 UNDERSIZEPKTS

Counts number of frames received with size less than 64 bytes.

Frame size:	< 64 bytes,
No FCS error	
No framing error	
No collisions	

11.4.1.7 OVERSIZEPKTS

Counts number of frames received with size exceeding the maximum allowable frame size.

Frame size:	1522 bytes if VLAN tag (1518 bytes if no VLAN)
FCS error	don’t care
Framing error	don’t care
No collisions	

11.4.1.8 FRAGMENTS

Counts number of frames received with size less than 64 bytes and with bad FCS.

Frame size:	< 64 bytes
Framing error	don’t care
No collisions	

11.4.1.9 Jabbers

Counts number of frames received with size exceeding maximum frame size and with bad FCS.

Frame size:	> 1522 bytes if VLAN tag (1518 bytes if no VLAN)
Framing error	don’t care
No collisions	

11.4.1.10 Collisions

Counts number of collision events detected.

Only a best estimate since collisions can only be detected while in transmit mode, but not while in receive mode.

Frame size:	any size

11.4.1.11 Packet Count for Different Size Groups

Six different size groups – one counter for each:

Pkts64Octets	for any packet with size = 64 bytes
Pkts65to127Octets	for any packet with size from 65 bytes to 127 bytes
Pkts128to255Octets	or any packet with size from 128 bytes to 255 bytes
Pkts256to511Octets	for any packet with size from 256 bytes to 511 bytes
Pkts512to1023Octets	for any packet with size from 512 bytes to 1023 bytes
Pkts1024to1518Octets	for any packet with size from 1024 bytes to 1518 bytes

Counts both good and bad packets.

11.5 Miscellaneous Counters

In addition to the statistics groups defined in previous sections, the ZL50409 has other statistics counters for its own purposes. We have two counters for flow control – one counting the number of flow control frames received, and another counting the number of flow control frames sent. We also have two counters, one for unicast frames sent, and one for non-unicast frames sent. A broadcast or multicast frame qualifies as non-unicast. Furthermore, we have a counter called “frame send fail.” This keeps track of FIFO under-runs, late collisions, and collisions that have occurred 16 times.

12.0 Register Definition

12.1 ZL50409 Register Description

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
0. ETHERNET Port Control Registers Substitute [N] with Port number (0..9)						
ECR1P”N”	Port Control Register 1 for Port N	000 + 2N	R/W	000-009	0C0	
ECR2P”N”	Port Control Register 2 for Port N	001 + 2N	R/W	00A-013	000	
ECR3P”N”	Port Control Register 3 for Port N	080 + 2N	R/W	014-01D	000	
ECR4P”N”	Port Control Register 4 for Port N	081 + 2N	R/W	01E-027	018	
BUF_LIMIT	Frame Buffer Limit	036	R/W	NA	040	
FCC	Flow Control Grant Period	037	R/W	NA	003	
1. VLAN Control Registers Substitute [N] with Port number (0..9)						
AVTCL	VLAN Type Code Register Low	100	R/W	028	000	
AVTCH	VLAN Type Code Register High	101	R/W	029	081	
PVMAP”N”_0	Port “N” Configuration Register 0	102 + 4N	R/W	02A-033	0FF	
PVMAP”N”_1	Port “N” Configuration Register 1	103 + 4N	R/W	034-03D	0FF	
PVMAP”N”_3	Port “N” Configuration Register 3	105 + 4N	R/W	03E-047	000	
PVMODE	VLAN Operating Mode	170	R/W	048	000	
2. TRUNK Control Registers						
TRUNK”N”	Trunk Group N (0..7)	200 + N	R/W	NA	000	
TRUNK”N”_ HASH10	Trunk Group N (0..7) Hash 10 Destination Port	208 + 4N	R/W	NA	000	
TRUNK”N”_ HASH32	Trunk Group N (0..7) Hash 32 Destination Port	209 + 4N	R/W	NA	000	
TRUNK”N”_ HASH54	Trunk Group N (0..7) Hash 54 Destination Port	20A + 4N	R/W	NA	000	
TRUNK”N”_ HASH76	Trunk Group N (0..7) Hash 76 Destination Port	20B + 4N	R/W	NA	000	

Table 9 - Register Description

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
MULTICAST_ HASH”N”-0	Multicast hash result N (0..7) mask byte 0	228 + 2N	R/W	NA	0FF	
MULTICAST_ HASH”N”-1	Multicast hash result N (0..7) mask byte 1	229 + 2N	R/W	NA	0FF	
3. CPU Port Configuration						
MAC0	CPU MAC Address byte 0	300	R/W	NA	000	
MAC1	CPU MAC Address byte 1	301	R/W	NA	000	
MAC2	CPU MAC Address byte 2	302	R/W	NA	000	
MAC3	CPU MAC Address byte 3	303	R/W	NA	000	
MAC4	CPU MAC Address byte 4	304	R/W	NA	000	
MAC5	CPU MAC Address byte 5	305	R/W	NA	000	
INT_MASK0	Interrupt Mask 0	306	R/W	NA	000	
INTP_MASK”N”	Interrupt Mask for MAC Port 2N, 2N+1	310+N (310 –314)	R/W	NA	000	
RQS	Receive Queue Select	323	R/W	NA	000	
RQSS	Receive Queue Status	324	RO	NA	NA	
MAC01	Increment MAC port 0,1 address	325	R/W	NA	000	
MAC23	Increment MAC port 2,3 address	326	R/W	NA	000	
MAC45	Increment MAC port 4,5 address	327	R/W	NA	000	
MAC67	Increment MAC port 6,7 address	328	R/W	NA	000	
MAC9	Port 9 MAC address byte 5	329	R/W	NA	000	
CPUQINS[6:0]		330-6	R/W	NA	000	
CPUQINSRPT		337	RO	NA	NA	
CPUGRNHDL[1:0]		338-9	RO	NA	NA	
CPURLSINFO[4:0]		33A-E	R/W	NA	000	
CPUGRNCTR		33F	R/W	NA	000	
4. Search Engine Configurations						
AGETIME_LOW	MAC Address Aging Time Low	400	R/W	049	05C	
AGETIME_ HIGH	MAC Address Aging Time High	401	R/W	04A	000	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
SE_OPMODE	Search Engine Operating Mode	403	R/W	NA	000	
5. Global QOS Control						
QOSC	QOS Control	500	R/W	04B	000	
UCC	Unicast Congestion Control	510	R/W	068	006	
MCC	Multicast Congestion Control	511	R/W	069	006	
MCCTH	Multicast Congestion Threshold	512	R/W	NA	003	
RDRC0	WRED Drop Rate Control 0	513	R/W	090	000	
RDRC1	WRED Drop Rate Control 1	514	R/W	091	000	
RDRC2	WRED Drop Rate Control 2	515	R/W	NA	000	
SFCB	Share FCB Size	518	R/W	074	000	
C1RS	Class 1 Reserve Size	519	R/W	075	000	
C2RS	Class 2 Reserve Size	51A	R/W	076	000	
C3RS	Class 3 Reserve Size	51B	R/W	077	000	
AVPML	VLAN Priority Map Low	530	R/W	056	000	
AVPMM	VLAN Priority Map Middle	531	R/W	057	000	
AVPMH	VLAN Priority Map High	532	R/W	058	000	
AVDM	VLAN Discard Map	533	R/W	05C	000	
TOSPML	TOS Priority Map Low	540	R/W	059	000	
TOSPMM	TOS Priority Map Middle	541	R/W	05A	000	
TOSPMH	TOS Priority Map High	542	R/W	05B	000	
TOSDML	TOS Discard Map	543	R/W	05D	000	
USER_PROTOCOL_[7:0]	User Define Protocol 0~7	550 - 557	R/W	0B3 - 0BA	000	
USER_PROTOCOL_FORCE _DISCARD[7:0]	User Define Protocol 7 To 0 Force Discard Enable	558	R/W	0BB	000	
WLPP10	Well known Logic Port Priority for 1 and 0	560	R/W	0A8	000	
WLPP32	Well known Logic Port Priority for 3 and 2	561	R/W	0A9	000	
WLPP54	Well known Logic Port Priority for 5 and 4	562	R/W	0AA	000	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
WLPP76	Well-known Logic Port Priority for 7 & 6	563	R/W	0AB	000	
WLPE[7:0]	Well known Logic 7 To 0 Port Enable	564	R/W	0AC	000	
WLPFD[7:0]	Well Known Logic 7 To 0 Port Force Discard Enable	565	R/W	0AD	000	
USER_ PORT”N”_LOW	User Define Logical Port “N” Low (N=0-7)	570 + 2N	R/W	092-099	000	
USER_ PORT”N”_HIGH	User Define Logical Port “N” High	571 + 2N	R/W	09A-0A1	000	
USER_ PORT1:0_ PRIORITY	User Define Logic Port 1 and 0 Priority	590	R/W	0A2	000	
USER_ PORT3:2_ PRIORITY	User Define Logic Port 3 and 2 Priority	591	R/W	0A3	000	
USER_ PORT5:4_ PRIORITY	User Define Logic Port 5 and 4 Priority	592	R/W	0A4	000	
USER_ PORT7:6_PRIORITY	User Define Logic Port 7 and 6 Priority	593	R/W	0A5	000	
USER_PORT_ENABLE[7:0]	User Define Logic 7 To 0 Port Enable	594	R/W	0A6	000	
USER_PORT_FORCE_DISCARD[7:0]	User Define Logic 7 To 0 Port Force Discard Enable	595	R/W	0A7	000	
RLOWL	User Define Range Low Bit7:0	5A0	R/W	0AE	000	
RLOWH	User Define Range Low Bit 15:8	5A1	R/W	0AF	000	
RHIGHL	User Define Range High Bit 7:0	5A2	R/W	0B0	000	
RHIGHH	User Define Range High Bit 15:8	5A3	R/W	0B1	000	
RPRIORITY	User Define Range Priority	5A4	R/W	0B2	000	
6. MISC Configuration Register						
MII_OP0	MII Register Option 0	600	R/W	0BC	000	
MII_OP1	MII Register Option 1	601	R/W	0BD	000	
FEN	Feature Registers	602	R/W	0BE	010	
MIIC0	MII Command Register 0	603	R/W	N/A	000	
MIIC1	MII Command Register 1	604	R/W	N/A	000	
MIIC2	MII Command Register 2	605	R/W	N/A	000	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
MIIC3	MII Command Register 3	606	R/W	N/A	000	
MIID0	MII Data Register 0	607	RO	N/A	NA	
MIID1	MII Data Register 1	608	RO	N/A	NA	
USD	One micro second divider	609	R/W	N/A	000	
DEVICE	Device id and test	60A	R/W	N/A	002	
SUM	EEPROM Checksum Register	60B	R/W	0FF	000	
LHBTimer	Link heart beat time out timer	610	R/W	N/A	000	
LHBReg0	LHB control filed value[7:0]	611	R/W	N/A	000	
LHBReg1	LHB control filed value[15:8]	612	R/W	N/A	000	
fMACCReg0	Forced MAC control filed value[7:0]	613	R/W	N/A	000	
fMACCReg1	Forced MAC control filed value[15:8]	614	R/W	N/A	000	
FCB_BASE_ADDR0	FCB Base Address Register 0	620	R/W	0BF	000	
FCB_BASE_ADDR1	FCB Base Address Register 1	621	R/W	0C0	060	
FCB_BASE_ADDR2	FCB Base Address Register 2	622	R/W	0C1	000	
7. Port Mirroring Controls						
MIRROR_DEST_MAC0	Mirror Destination Mac Address 0	700	R/W	N/A	000	
MIRROR_DEST_MAC1	Mirror Destination Mac Address 1	701	R/W	N/A	000	
MIRROR_DEST_MAC2	Mirror Destination Mac Address 2	702	R/W	N/A	000	
MIRROR_DEST_MAC3	Mirror Destination Mac Address 3	703	R/W	N/A	000	
MIRROR_DEST_MAC4	Mirror Destination Mac Address 4	704	R/W	N/A	000	
MIRROR_DEST_MAC5	Mirror Destination Mac Address 5	705	R/W	N/A	000	
MIRROR_SRC_MAC0	Mirror Source Mac Address 0	706	R/W	N/A	000	
MIRROR_SRC_MAC1	Mirror Source Mac Address 1	707	R/W	N/A	000	
MIRROR_SRC_MAC2	Mirror Source Mac Address 2	708	R/W	N/A	000	
MIRROR_SRC_MAC3	Mirror Source Mac Address 3	709	R/W	N/A	000	
MIRROR_SRC_MAC4	Mirror Source Mac Address 4	70A	R/W	N/A	000	
MIRROR_SRC_MAC5	Mirror Source Mac Address 5	70B	R/W	N/A	000	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
MIRROR_CONTROL	Port Mirror Control Register	70C	R/W	N/A	000	
RMAC_MIRROR0	RMAC Mirror 0	710	R/W	N/A	000	
RMAC_MIRROR1	RMAC Mirror 1	711	R/W	N/A	000	
8. Per Port QOS Control						
FCR-N	Flooding Control Register N (0..9)	800-809	R/W	04C-055	000	
BMRC-N	Broadcast/Multicast Rate Control N (0..9)	820-829	R/W	05E-067	000	
PR100_N	Port Reservation for RMAC Ports (Port 0..7)	840-847	R/W	06A-071	006	‘d1536/16=‘d96, ‘d96>>4=’h6
PR100_CPU	Port Reservation for CPU Ports	848	R/W	073	006	‘d96
PRM	Port Reservation for MMAC Port	849	R/W	072	024	‘d96x6=‘d576, ‘d576>>4=’h24
PTH100_N	Port Threshold for RMAC Ports (Port 0..7)	860-867	R/W	0C2-0C9	003	½
PTH100_CPU	Port Threshold for CPU Port	868	R/W	0CB	003	½
PTHM	Port Threshold for MMAC Port	869	R/W	0CA	012	½
QOSC”N”	QOS Control (N=0 – 15)	880-88F	R/W	078-087	000	
	QOS Control (N=16 – 21)	890-895	R/W	NA	000	
	QOS Control (N=22 – 27)	896-89B	R/W	088-08D	000	
	QOS Control (N=28 – 39)	89C-8A7	R/W	NA	000	
E. System Diagnostic						
DTSRL	Test Register Low	E00	R/W	NA	000	
DTSRM	Test Register Medium	E01	R/W	NA	001	
TESTOUT0	Testmux Output [7:0]	E02	R/O	NA	NA	
TESTOUT1	Testmux Output [15:8]	E03	R/O	NA	NA	
MASK0	MASK Timeout 0	E10	R/W	0F6	000	
MASK1	MASK Timeout 1	E11	R/W	0F7	000	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
MASK2	MASK Timeout 2	E12	R/W	0F8	000	
MASK3	MASK Timeout 3	E13	R/W	0F9	000	
MASK4	MASK Timeout 4	E14	R/W	0FA	000	
BOOTSTRAP[2:0]	BOOTSTRAP Read Back	E80 - E82	RO	NA	NA	
PRTFSMST-N (0-9)	Ethernet Ports Status Read Back	E90-E99	RO	NA	NA	
PRTQOSST-N (0-7)	RMAC Port [7:0] QOS and Queue Status	EA0-EA7	RO	NA	NA	
PRTQOSST8A	CPU Port QOS and Queue Status	EA8	RO	NA	NA	
PRTQOSST8B	CPU Port QOS and Queue Status	EA9	RO	NA	NA	
PRTQOSST9A	MMAC Port QOS and Queue Status	EAA	RO	NA	NA	
PRTQUSST9B	MMAC Port QOS and Queue Status	EAB	RO	NA	NA	
CLASSQOSST	Class Buffer Status	EAC	RO	NA	NA	
PRTINTCTR	Buffer Interrupt Status	EAD	R/W	NA	000	
QMCTRL-N (0-9)	Ports Queue Control Status	EB0-EB9	R/W	NA	000	
QCTRL	Ports Queue Control	EBA	R/W	NA	000	
BMBISTR0	Memory bist result	EBB	R/O	NA	NA	
BMBISTR1	Memory bist result	EBC	R/O	NA	NA	
BMControl	Memory control	EBD	R/W	NA	00F	
BUFF_RST	Buffer Reset Pool	EC0	R/W	NA	000	
FCBHEADPTR0	FCB Head Pointer [7:0]	EC1	R/W	NA	000	
FCB_HEAD_PTR1	FCB Head Pointer [15:8]	EC2	R/W	NA	000	
FCB_TAIL_PTR0	FCB Tail Pointer [7:0]	EC3	R/W	NA	000	
FCB_TAIL_PTR1	FCB Tail Pointer [15:8]	EC4	R/W	NA	000	
FCB_NUM0	FCB Number [7:0]	EC5	R/W	NA	000	
FCB_NUM1	FCB Init Start and FCB Number [14:8]	EC6	R/W	NA	006	
BM_RLSFF_CTRL	Read control register	EC7	R/W	NA	000	
BM_RLSFF_INFO0	Bm_rlsfifo_info[7:0]	EC8	RO	NA	NA	
BM_RLSFF_INFO1	Bm_rlsfifo_info[15:8]	EC9	RO	NA	NA	
BM_RLSFF_INFO2	Bm_rlsfifo_info[23:16]	ECA	RO	NA	NA	

Table 9 - Register Description (continued)

Register	Description	CPU Addr (Hex)	R/W	I²C Addr (Hex)	Default	Notes
BM_RLSFF_INFO3	Bm_rlsfifo_info[31:24]	ECB	RO	NA	NA	
BM_RLSFF_INFO4	Bm_rlsfifo_info[39:32]	ECC	RO	NA	NA	
BM_RLSFF_INFO5	Fifo_cnt[2:0],Bm_rlsfifo_info[44:40]	ECD	RO	NA	NA	
F. System Control						
GCR	Global Control Register	F00	R/W	NA	000	
DCR	Device Control Register	F01	RO	NA	NA	
DCR1	Device Control Register 1	F02	RO	NA	NA	
DPST	Device Port Status Register	F03	R/W	NA	000	
DTST	Data read back register	F04	RO	NA	NA	
DA	DA Register	FFF	RO	NA	0DA	

Table 9 - Register Description (continued)

12.2 Directly Accessed Registers

12.2.1 INDEX_REG0

- Address bits [7:0] for indirectly accessed register addresses
- Address = 0 (write only)

12.2.2 INDEX_REG1 (only needed for 8-bit mode)

- Address bits [15:8] for indirectly accessed register addresses
- Address = 1 (write only)

12.2.3 DATA_FRAME_REG

- Data of indirectly accessed registers. (8 bits)
- Address = 2 (read/write)

12.2.4 CONTROL_FRAME_REG

- CPU transmit/receive switch frames. (8/16 bits)
- Address = 3 (read/write)
- Format:
  - Send frame from CPU: In sequence)

    Frame Data (size should be in multiple of 8-byte)

    8-byte of Frame status (Frame size, Destination port #, Frame O.K. status)
  - CPU Received frame: In sequence)

    8-byte of Frame status (Frame size, Source port #, VLAN tag)

    Frame Data

12.2.5 COMMAND&STATUS Register

- CPU interface commands (write) and status
- Address = 4 (read/write)
- When the CPU writes to this register

Bit [0]:	Set Control Frame Receive buffer ready, after CPU writes a complete frame into the buffer. This bit is self-cleared.
Bit [1]:	Set Control Frame Transmit buffer1 ready, after CPU reads out a complete frame from the buffer. This bit is self-cleared.
Bit [2]:	Set Control Frame Transmit buffer2 ready, after CPU reads out a complete frame from the buffer. This bit is self-cleared.
Bit [3]:	Set this bit to indicate CPU received a whole frame (transmit FIFO frame receive done), and flushed the rest of frame fragment, If occurs. This bit will be self-cleared.
Bit [4]:	Set this bit to indicate that the following Write to the Receive FIFO is the last one (EOF). This bit will be self-cleared.
Bit [5]:	Set this bit to re-start the data that is sent from the CPU to Receive FIFO (re-align). This feature can be used for software debug. For normal operation must be '0'.
Bit [7:6]:	Reserved. Must be '0'

When the CPU reads this register:

Bit [0]:	Control Frame receive buffer ready, CPU can write a new frame 1 – CPU can write a new control command 1 0 – CPU has to wait until this bit is 1 to write a new control command 1
Bit [1]:	Control Frame transmit buffer1 ready for CPU to read 1 – CPU can read a new control command 1 0 – CPU has to wait until this bit is 1 to read a new control command
Bit [2]:	Control Frame transmit buffer2 ready for CPU to read 1 – CPU can read a new control command 1 0 – CPU has to wait until this bit is 1 to read a new control command
Bit [3]:	Transmit FIFO has data for CPU to read (TXFIFO_RDY)
Bit [4]:	Receive FIFO has space for incoming CPU frame (RXFIFO_SPOK)
Bit [5]:	Transmit FIFO End Of Frame (TXFIFO_EOF)
Bit [7:6]:	Reserved

12.2.6 Interrupt Register

- Interrupt sources (8 bits)
- Address = 5 (read only)
- When CPU reads this register

Bit [0]:	CPU frame interrupt
Bit [1]:	Control Frame 1 interrupt. Control Frame receive buffer1 has data for CPU to read
Bit [2]:	Control Frame 2 interrupt. Control Frame receive buffer2 has data for CPU to read
Bit [6:3]:	Reserved
Bit [7]:	Device Timeout Detected interrupt
Note: This register is not self-cleared. After reading CPU has to clear the bit writing 0 to it.	

12.2.7 Control Command Frame Buffer1 Access Register

- Address = 6 (read/write)
- When CPU writes to this register, data is written to the Control Command Frame Receive Buffer
- When CPU reads this register, data is read from the Control Command Frame Transmit Buffer1

12.2.8 Control Command Frame Buffer2 Access Register

- Address = 7 (read only)
- When CPU reads this register, data is read from the Control Command Frame Transmit Buffer1

12.3 Indirectly Accessed Registers

12.3.1 (Group 0 Address) MAC Ports Group

12.3.1.1 ECR1Pn: Port N Control Register

I²C Address 000 - 009; CPU Address:0000+2xN (N = port number)

Accessed by CPU and I²C (R/W)

Bit [0]	1 - Flow Control Off 0 - Flow Control On (Default) <u>When Flow Control On:</u> • In half duplex mode, the MAC transmitter applies back pressure for flow control. • In full duplex mode, the MAC transmitter sends Flow Control frames when necessary. The MAC receiver interprets and processes incoming flow control frames. The Flow Control Frame Received counter is incremented whenever a flow control is received. <u>When Flow Control Off:</u> • In half duplex mode, the MAC transmitter does not assert flow control by sending flow control frame or jamming collision. • In full duplex mode, the Mac transmitter does not send flow control frames. The MAC receiver does not interpret or process the flow control frames. The Flow Control Frame Received counter is not incremented.

Bit [1]	1 - Half Duplex - Only in 10/100 mode 0 - Full Duplex (Default)
Bit [2]	1 - 10Mbps 0 - 100Mbps (Default)
Bit [4:3]	00 - Enable Auto-Negotiation This enables hardware state machine for auto-negotiation. (Default) 01 - Limited Disable Auto-Negotiation This disables hardware state machine for speed auto-negotiation (use ECR1Pn[2:0] for configuration). Hardware will still poll PHY for link status. 10 - Force Link Down Disable the port. Hardware does not talk to PHY. 11 - Force Link Up The configuration in ECR1Pn[2:0] is used for (speed/duplex/flow control) setup. Hardware does not talk to PHY.
Bit [5]	Asymmetric Flow Control Enable. 0 – Disable asymmetric flow control (Default) 1 – Enable Asymmetric flow control When this bit is set and flow control is on (bit[0] = 0), the device does not send out flow control frames, but it's receiver interprets and processes flow control frames.
Bit [7:6]	SS - Spanning tree state (802.1D spanning tree protocol) 00 - Blocking: Frame is dropped 01 - Listening: Frame is dropped 10 - Learning: Frame is dropped. Source MAC address is learned. 11 - Forwarding: Frame is forwarded. Source MAC address is learned. (Default)

12.3.1.2 ECR2Pn: Port N Control Register

I²C Address: 00A-013; CPU Address:0001+2xN (N = port number)

Accessed by CPU and I²C (R/W)

Bit[0]:	Filter untagged frame 0: Disable (Default) 1: All untagged frames from this port are discarded or follow security option when security is enable
Bit[1]:	Filter Tag frame 0: Disable (Default) 1: All tagged frames from this port are discarded or follow security option when security is enable
Bit[2]:	Learning Disable 0: Learning is enabled on this port (Default) 1: Learning is disabled on this port
Bit[3]:	Rate control timer select 0: 10 microsecond refreshing time (Default) 1: 1 millisecond refreshing time
Bit[4]	Reserved

Bit [5]	Do not change VLAN tag. This control over PVMAPnn_3 bit[2]. If this bit is set, no tag will be replaced nor removed. (Default 0) 1: Enable 0: Disable
Bit[7:6]	Security Enable. The ZL50409 checks the incoming data for one of the following conditions: • If the source MAC address of the incoming packet is in the MAC table and is defined as secure address but the ingress port is not the same as the port associated with the MAC address in the MAC table. • A MAC address is defined as secure when its entry at MAC table has static status and bit 0 is set to 1. MAC address bit 0 (the first bit transmitted) indicates whether the address is unicast or multicast. As source addresses are always unicast bit 0 is not used (always 0). ZL50409 uses this bit to define secure MAC addresses. • If the port is set as learning disable and the source MAC address of the incoming packet is not defined in the MAC address table or the MAC address is not associated to the ingress port. If any one of the conditions is met, the packet is forwarded based on these setting. 00 – Disable port security, forward packets as usual. (Default) 01 – Discard violating packets 10 – Forward violating packets as usual and also to the CPU for inspection 11 – Forward violating packets to the CPU for inspection As well, if: • the port is configured to filter untagged frames and an untagged frame arrives, or • the port is configured to filter tagged frames and a tagged frame arrives, or • the packet has the source mac address on the source mac address filter list, or • the packet has the destination mac address on the destination mac address filter list, the packet will be handled according to: 0X – Discard violating packets 1X – Forward violating packets to CPU for inspection

12.3.1.3 ECR3Pn: Port N Control Register

I²C Address: 014-01D; CPU Address:0080+2xN (N = port number)

Accessed by CPU and I²C (R/W)

Bit[0]:	Enable receiving short frame < 64B 0: Disable (Default) 1: Allow receiving short frame with correct CRC.
Bit[1]:	Enable receiving long frame > 1522 0: Disable (Default) 1: Allow receiving long frame that smaller than “BUF_LIMIT” value
Bit[2]:	Enable pad frame to 64B when transmitted 1: Disable (Default) 0: Allow padding to 64B
Bit[3]:	Enable compress preamble 1: Only one byte preamble+SFD 0: Send standard preamble (Default)

Bit[6:4]	Number of bytes removed from the IFG. (Default 0)
Bit[7]	Link Heart Beat Transmit (RMAC ports only) 0: Disable (Default) 1: Enable

12.3.1.4 ECR4Pn: Port N Control Register

I²C Address: 01E-027; CPU Address:0081+2xN (N = port number)

Accessed by CPU and I²C (R/W)

Port 0 – 7 and Port 9: (RMAC Ports & MMAC Port)

Bit[0]:	Enable TXCLK output. Active high 0: Disable (Default) 1: Txclk pin becomes output in GPSI or MII mode
Bit[1]:	Enable RXCLK output. Active high 0: Disable (Default) 1: Rxclk pin becomes output in GPSI or MII mode
Bit[2]:	Internal loopback. 0: Disable (Default) 1: Enable In this mode, the packet is looped back in the MAC layer before going out of the chip. You must force linkup at full duplex as well. External loopback is another level of system diagnostic which involves the PHY device to loopback the packet.
Bit[4:3]:	RMAC interface mode: 00 - GPSI mode 01 - MII mode 10 - Reserved 11 - RMII mode (Default) MMAC interface mode (Port 9): 11 - MII mode (Default)
Bit[5]:	Frame loopback. 0: Disable frame from sending back to its source port. (Default) 1: Allow frame to send back to its source port In a regular ethernet switch, a packet should never be receive and forwarded to the same port. Setting the bit allows it to happen. This is not the same as an ingress MAC loopback. The destiniation MAC address has to be stored (learned) in the MCT and associated with the originating source port. The frame loopback will only work for unicast packets.
Bit[6]:	Link Heart Beat Receive (RMAC ports only) 0: Disable (Default). Also clears all MAC LHB status. 1: Enable
Bit[7]:	Soft reset. 0: Normal operation (Default) 1: Reset. Not self clearing.

Port 8: (CPU port)

Bit[1:0]:	Reserved
Bit[2]:	Enable special write to 2 registers in a single write operation. 0: Disable (Default) 1: Enable Should be enabled only in serial mode and disabled in 8/16-bit mode.
Bit[4:3]:	Enable insertion of 2-byte CPU information in CPU frame packet in Serial + MII mode 00: No information is inserted (Default) 01: Insert 2-byte of CPU information 10: Reserved 11: Insert 6-byte of padding + 2-byte of CPU information In port-based VLAN mode, the CPU MII interface must be in "No information is inserted" mode (ECR4P8[4:3]='00'). In tagged-based VLAN mode, the CPU MII interface supports all three modes (0,2,8 bytes insertion).
Bit[5]:	Enable frame loop back. In a regular ethernet switch, a packet should never receive and forwarded to the same port. Setting the bit allows it to happen. 0: Disable frame from send back to its source port. (Default) 1: Allow frame to send back to its source port
Bit[6]:	Reserved
Bit[7]:	Soft reset. 0: Normal operation (Default) 1: Reset. Not self clearing.

12.3.1.5 BUF_LIMIT – Frame Buffer Limit

CPU Address:h036

Accessed by CPU (R/W)

Bit[6:0]:	Frame Buffer Limit (max 4KB). Multiple of 64 bytes (Default 0x40)
Bit[7]:	Reserved

12.3.1.6 FCC – Flow Control Grant Period

CPU Address:h037

Accessed by CPU (R/W)

Bit[2:0]:	Flow Control Grant Period (Default 0x3)
Bit[7:3]:	Reserved

12.3.2 (Group 1 Address) VLAN Group

12.3.2.1 AVTCL – VLAN Type Code Register Low

I²C Address 028; CPU Address:h100

Accessed by CPU and I²C (R/W)

Bit[7:0]:	VLANType_LOW: Lower 8 bits of the VLAN type code (Default 0)

12.3.2.2 AVTCH – VLAN Type Code Register High

I²C Address 029; CPU Address:h101

Accessed by CPU and I²C (R/W)

Bit[7:0]:	VLANType_HIGH: Upper 8 bits of the VLAN type code (Default is 0x81)

12.3.2.3 PVMAP00_0 – Port 0 Configuration Register 0

I²C Address 02A, CPU Address:h102

Accessed by CPU and I²C (R/W)

In Port Based VLAN Mode

Bit[7:0]:	VLAN Mask for port 0 (Default 0xFF)

This register indicates the legal egress ports. A “1” on bit 7 means that the packet can be sent to port 7. A “0” on bit 7 means that any packet destined to port 7 will be discarded. This register works with registers 1 to form a 10 bit mask to all egress ports.

In Tag based VLAN Mode

Bit[7:0]:	PVID [7:0] (Default is 0xFF)

This is the default VLAN tag. It works with configuration register PVMAP00_1 [7:5] [3:0] to form a default VLAN tag. If the received packet is untagged, then the packet is classified with the default VLAN tag. If the received packet has a VLAN ID of 0, then PVID is used to replace the packet’s VLAN ID.

12.3.2.4 PVMAP00_1 – Port 0 Configuration Register 1

I²C Address h34, CPU Address:h103

Accessed by CPU and I²C (R/W)

In Port based VLAN Mode

Bit[1:0]:	VLAN Mask for ports 9 to 8 (Default 0x3)
Bit[7:2]:	Reserved (Default 0x3F)

In Tag based VLAN Mode

Bit[3:0]:	PVID [11:8] (Default is 0xF)

Bit [4]:	Untrusted Port. **(Default is 1)** This register is used to change the VLAN priority field of a packet to a predetermined priority. 1: VLAN priority field is changed to Bit[7:5] at ingress port 0: Keep VLAN priority field
Bit [7:5]:	Untag Port Priority **(Default 0x7)**

12.3.2.5 PVMAP00_3 – Port 0 Configuration Register 3

I²C Address h3E, CPU Address:h105

Accessed by CPU and I²C (R/W)

In Port Based VLAN Mode

Bit [2:0]:	Reserved
Bit [5:3]:	Default Transmit priority. Used when Bit[7]=1 (Default 0) Transmit Priority Level 0 (Lowest) Transmit Priority Level 1 Transmit Priority Level 2 Transmit Priority Level 3 (Highest)
Bit [6]:	Default Discard priority. Used when Bit[7]=1 0 – Discard Priority Level 0 (Lowest) (Default) 1 – Discard Priority Level 1(Highest)
Bit [7]:	Enable Fix Priority (Default 0) 0 - Disable. All frames are analysed. Transmit Priority and Discard Priority are based on VLAN Tag, TOS or Logical Port. 1 - Enable. Transmit Priority and Discard Priority are based on values programmed in bit [6:3]

In Tag-based VLAN Mode

Bit [0]:	Not used
Bit [1]:	Ingress Filter Enable 0 - Disable Ingress Filter. Packets with VLAN not belonging to source port are forwarded, if destination port belongs to the VLAN. Symmetric VLAN. (Default) 1 - Enable Ingress Filter. Packets with VLAN not belonging to source port are filtered. Asymmetric VLAN.
Bit [2]:	Force untag out (VLAN tagging is based on 802.1q rule)**.** 0 - Disable (Default) 1 - Force untagged output. All packets transmitted from this port are untagged. This bit is used when this port is connected to legacy equipment that does not support VLAN tagging.
Bit [5:3]:	Default Transmit priority. Used when Bit[7]=1 (Default 0) Transmit Priority Level 0 (Lowest) Transmit Priority Level 1 Transmit Priority Level 2 Transmit Priority Level 3 (Highest)

Bit [6]:	Default Discard priority. Used when Bit[7]=1 0 – Discard Priority Level 0 (Lowest) (Default) 1 – Discard Priority Level 1(Highest)
Bit [7]:	Enable Fix Priority (Default 0) 0 - Disable. All frames are analysed. Transmit Priority and Discard Priority are based on VLAN Tag, TOS or Logical Port. 1 - Enable. Transmit Priority and Discard Priority are based on values programmed in bit [6:3]

12.3.2.6 PVMAPnn_0,1,3 – Ports 1~9 Configuration Registers

PVMAP01_0,1,3 I²C Address h2B,35,3F; CPU Address:h106,107,109 (Port 1)

PVMAP02_0,1,3 I²C Address h2C,36,40; CPU Address:h10A, 10B, 10D (Port 2)

PVMAP03_0,1,3 I²C Address h2D,37,41; CPU Address:h10E, 10F, 111 (Port 3)

PVMAP04_0,1,3 I²C Address h2E,38,42; CPU Address:h112, 113, 115 (Port 4)

PVMAP05_0,1,3 I²C Address h2F,39,43; CPU Address:h116, 117, 119 (Port 5)

PVMAP06_0,1,3 I²C Address h30,3A,44; CPU Address:h11A, 11B, 11D (Port 6)

PVMAP07_0,1,3 I²C Address h31,3B,45; CPU Address:h11E, 11F, 121 (Port 7)

PVMAP08_0,1,3 I²C Address h32,3C,46; CPU Address:h122, 123, 125 (Port CPU)

PVMAP09_0,1,3 I²C Address h33,3D,47; CPU Address:h126, 127, 129 (Port MMAC)

12.3.2.7 PVMODE

I²C Address: h048, CPU Address:h170

Accessed by CPU and I²C (R/W)

Bit [0]:	VLAN Mode 1: Tag based VLAN Mode 0: Port based VLAN Mode (Default)
Bit [1]:	Slow learning (Default = 0) Same function as SE_OP MODE bit[7]. Either bit can enable the function; both need to be turned off to disable the feature.
Bit [2]:	Disable dropping frames with destination MAC addresses 0180C2000001 to 0180C200000F 0: Drop all frames in the range (Default) 1: Treats frames as multicast
Bit [3]:	Flooding control in secure mode 0: Enable - Learning disabled port will not receive any flooding packets (Default) 1: Disable
Bit [4]:	Support MAC address 0 0: MAC address 0 is not learned. (Default) 1: MAC address 0 is learned.

Bit [5]:	Disable spanning tree packet to CPU in managed mode 1: Received spanning tree packet is forwarded as multicast. 0: Received spanning tree packet is forwarded to CPU. (Default)
Bit [6]:	IP Multicast Enable 0: Disable (default) 1: Enable In general, this bit is equal to ^FEN[4].
Bit [7]:	Enable logical port match in secure mode 0: Disable (Default) 1: Enable - When Well Known or User Define logical port force discard enabled, force any IP packet with logical port number matching logical port numbers to CPU.

12.3.3 (Group 2 Address) Port Trunking Groups

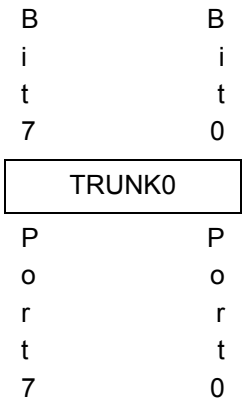
Trunk Group – Up to eight RMAC ports can be selected for each trunk group.

12.3.3.1 TRUNKn– Trunk Group 0~7

CPU Address:h200+N (N = trunk group)

Accessed by CPU (R/W)

Bit [7:0] Port 7-0 bit map of trunk N. (Default 0)



12.3.3.2 TRUNKn_HASH10 – Trunk group 0~7 hash result 1/0 destination port number

CPU Address:h208+4xN (N = trunk group)

Accessed by CPU (R/W)

Bit [3:0]	Hash result 0 destination port number (Default 0)
Bit [7:4]	Hash result 1 destination port number (Default 0)

12.3.3.3 TRUNKn_HASH32 – Trunk group 0~7 hash result 3/2 destination port number

CPU Address:h209+4xN (N = trunk group)

Accessed by CPU (R/W)

Bit [3:0]	Hash result 2 destination port number (Default 0)
Bit [7:4]	Hash result 3 destination port number (Default 0)

12.3.3.4 TRUNKn_HASH54 – Trunk group 0~7 hash result 5/4 destination port number

CPU Address:h20A+4xN (N = trunk group)

Accessed by CPU (R/W)

Bit [3:0]	Hash result 4 destination port number (Default 0)
Bit [7:4]	Hash result 5 destination port number (Default 0)

12.3.3.5 TRUNKn_HASH76 – Trunk group 0~7 hash result 7/6 destination port number

CPU Address:h20B+4xN (N = trunk group)

Accessed by CPU (R/W)

Bit [3:0]	Hash result 6 destination port number (Default 0)
Bit [7:4]	Hash result 7 destination port number (Default 0)

12.3.4 Multicast Hash Registers

Multicast Hash registers are used to distribute multicast traffic. 16 registers are used to form a 8-entry array; each entry has 10 bits, with each bit representing one port. Any port not belonging to a trunk group should be programmed with 1. Ports belonging to the same trunk group should only have a single port set to “1” per entry. The port set to “1” is picked to transmit the multicast frame when the hash value is met.

Hash Value =0	HASH0-1	HASH0-0
Hash Value =1	HASH1-1	HASH1-0
Hash Value =2	HASH2-1	HASH2-0
Hash Value =3	HASH3-1	HASH3-0
Hash Value =4	HASH4-1	HASH4-0
Hash Value =5	HASH5-1	HASH5-0
Hash Value =6	HASH6-1	HASH6-0
Hash Value =7	HASH7-1	HASH7-0

Port 9 Port 8 Port 7 Port 0

12.3.4.1 MULTICAST_HASHn-0 – Multicast hash result 0~7 mask byte 0

CPU Address:h228+2xN (N = hash value)

Accessed by CPU (R/W)

Bit[7:0]:	Port 7-0 bit map for multicast hash. (Default 0xFF)

12.3.4.2 MULTICAST_HASHn-1 – Multicast hash result 0~7 mask byte 1

CPU Address:h229+2xN (N = hash value)

Accessed by CPU (R/W)

Bit[1:0]:	Port 9-8 bit map for multicast hash. (Default 0x3)
Bit[5:2]:	Reserved (Default 0xF)
Bit[7:6]:	MULTICAST_HASH0-1 Hash Select. The hash algorithm selected is valid for all trunks (Default 00) 00 - Use Source and Destination Mac Address for hashing 01 - Use Source Mac Address for hashing 10 - Use Destination Mac Address for hashing 11 - Use Source Port Number for hashing MULTICAST_HASH[7:1]-1 Reserved (Default 0x3)

12.3.5 (Group 3 Address) CPU Port Configuration Group

5 0 (MC bit)

MAC5	MAC4	MAC3	MAC2	MAC1	MAC0

MAC5 to MAC0 registers form the CPU MAC address. When a packet with destination MAC address match MAC [5:0], the packet is forwarded to the CPU.

12.3.5.1 MAC0 – CPU Mac address byte 0

CPU Address:h300

Accessed by CPU (R/W)

Bit[7:0]:	Byte 0 of the CPU MAC address (Default 0)

12.3.5.2 MAC1 – CPU Mac address byte 1

CPU Address:h301

Accessed by CPU (R/W)

Bit[7:0]:	Byte 1 of the CPU MAC address (Default 0)

12.3.5.3 MAC2 – CPU Mac address byte 2

CPU Address:h302

Accessed by CPU (R/W)

Bit[7:0]:	Byte 2 of the CPU MAC address (Default 0)

12.3.5.4 MAC3 – CPU Mac address byte 3

CPU Address:h303

Accessed by CPU (R/W)

Bit[7:0]:	Byte 3 of the CPU MAC address (Default 0)

12.3.5.5 MAC4 – CPU Mac address byte 4

CPU Address:h304

Accessed by CPU (R/W)

Bit[7:0]:	Byte 4 of the CPU MAC address (Default 0)

12.3.5.6 MAC5 – CPU Mac address byte 5

CPU Address:h305

Accessed by CPU (R/W)

Bit[7:0]:	Byte 5 of the CPU MAC address (Default 0)

12.3.5.7 INT_MASK0 – Interrupt Mask

CPU Address:h306

Accessed by CPU (R/W)

The CPU can dynamically mask the interrupt when it is busy and doesn't want to be interrupted. (Default 0xFF)

- 1: Mask the interrupt
- 0: Unmask the interrupt (Enable interrupt)

Bit [0]:	CPU frame interrupt. CPU frame buffer has data for CPU to read
Bit [1]:	Control Command 1 interrupt. Control Command Frame buffer1 has data for CPU to read

Bit [2]:	Control Command 2 interrupt. Control command Frame buffer2 has data for CPU to read
Bit [6:3]:	Reserved
Bit [7]:	Device Timeout Detected interrupt

12.3.5.8 INTP_MASK0 – Interrupt Mask for MAC Port 0,1

CPU Address:h310

Accessed by CPU (R/W)

The CPU can dynamically mask the interrupt when it is busy and doesn't want to be interrupted (Default 0xFF)

- 1: Mask the interrupt
- 0: Unmask the interrupt

Bit [0]:	Port 0 statistic counter wrap around interrupt mask. An Interrupt is generated when a statistic counter wraps around. Refer to hardware statistic counter for interrupt sources
Bit [1]:	Port 0 link change mask
Bit [2]:	Port 0 module detect mask
Bit [3]:	Reserved
Bit [4]:	Port 1 statistic counter wrap around interrupt mask. An interrupt is generated when a statistic counter wraps around. Refer to hardware statistic counter for interrupt sources.
Bit [5]:	Port 1 link change mask
Bit [6]:	Port 1 module detect mask
Bit [7]:	Reserved

12.3.5.9 INTP_MASKn – Interrupt Mask for MAC Ports 2~9 Registers

INTP_MASK1 CPU Address:h311 (Ports 2,3)

INTP_MASK2 CPU Address:h312 (Ports 4,5)

INTP_MASK3 CPU Address:h313 (Ports 6,7)

INTP_MASK4 CPU Address:h314 (Port CPU,MMAC)

12.3.5.10 RQS – Receive Queue Select

CPU Address:h323

Accessed by CPU (RW)

Select which receive queue is used.

Bit [0]:	Select Queue 0
Bit [1]:	Select Queue 1
Bit[2]:	Select Queue 2
Bit[3]:	Select Queue 3
Bit[4]:	Select Multicast Queue 0

Bit[5]:	Select Multicast Queue 1
Bit[6]:	Select Multicast Queue 2
Bit[7]:	Select Multicast Queue 3

Note: Strict priority applies between different selected queues (UQ3>UQ2>UQ1>UQ0>MQ3>MQ2>MQ1>MQ0).

12.3.5.11 RQSS – Receive Queue Status

CPU Address:h324

Accessed by CPU (RO)

CPU receive queue status

Bit[3:0]:	Unicast Queue 3 to 0 not empty
Bit[7:4]:	Multicast Queue 3 to 0 not empty

12.3.5.12 MAC01 – Increment MAC port 0,1 address

CPU Address:h325

Accessed by CPU (RW)

Bit[2:0]:	Bit[42:40] of Port 0 MAC address
Bit[3]:	Reserved
Bit[6:4]:	Bit [42:40] of Port 1 MAC address
Bit[7]:	Reserved

12.3.5.13 MAC23 – Increment MAC port 2,3 address

CPU Address:h326

Accessed by CPU (RW)

Bit[2:0]:	Bit [42:40] of Port 2 MAC address
Bit[3]:	Reserved
Bit[6:4]:	Bit [42:40] of Port 3 MAC address
Bit[7]:	Reserved

12.3.5.14 MAC45 – Increment MAC port 4,5 address

CPU Address:h327

Accessed by CPU (RW)

Bit[2:0]:	Bit [42:40] of Port 4 MAC address
Bit[3]:	Reserved
Bit[6:4]:	Bit [42:40] of Port 5 MAC address
Bit[7]:	Reserved

12.3.5.15 MAC67 – Increment MAC port 6,7 address

CPU Address:h328

Accessed by CPU (RW)

Bit[2:0]:	Bit [42:40] of Port 6 MAC address
Bit[3]:	Reserved
Bit[6:4]:	Bit [42:40] of Port 7 MAC address
Bit[7]:	Reserved

12.3.5.16 MAC9 –Increment MAC port 9 address

CPU Address:h329

Accessed by CPU (RW)

Bit[7:0]:	Bit[47:40] of Port 9 MAC address

12.3.6 CPUQINS0 - CPUQINS6 -- CPU Queue Insertion Command

CPU Address:h330-336

Accessed by CPU, (R/W)

55 0

CQ6	CQ5	CQ4	CQ3	CQ2	CQ1	CQ0

CPU Queue insertion command

Bit[9:0]:	Destination Map (MMAC, CPU, port 7-0).
Bit[13:10]	Priority
Bit[20:14]	Number of granules for the frame
Bit[35:21]	Tail pointer
Bit[50:36]	Header Pointer
Bit[51]	Multicast frame (has to be one if more than one destination port)
Bit[54:52]	Reserved
Bit[55]	Command valid (will be processed on the rising edge of the signal)

12.3.7 CPUQINSRPT – CPU Queue Insertion Report

CPU Address:h337

Accessed by CPU, (RO)

CPU command queue status

Bit [0]:	The command is under processing.
Bit [1]:	Insertion Fail (May be due to queue full, WRED or filtering)

12.3.8 CPUGRNHDL0- CPUGRNHDL1 – CPU Allocated Granule Pointer

CPU Address:h338-339

Accessed by CPU, (RO)

15	0
CG1	CG0

CPU Queue insertion command

Bit [14:0]:	Granule pointer.
Bit [15]:	Pointer valid

12.3.9 CPURLSINFO0- CPURLSINFO4 – Receive Queue Status

CPU Address:h33A-33E

Accessed by CPU, (R/W)

				0
CR4	CR3	CR2	CR1	CR0

CPU Queue insertion command

Bit[14:0]:	Header pointer
Bit[30:15]	Tail pointer
Bit[38:32]	Number of granules for the release

12.3.10 CPUGRNCTR – CPU Granule Control

CPU Address:h33f

Accessed by CPU, (R/W)

CPU receive queue status

Bit [0]:	Allocate granule to the CPU if set to one. Otherwise, do not allocate any resource.
Bit [1]:	Read allocated granule (at rising edge only)
Bit [2]:	Release info valid (will be processed at rising edge only)

12.3.11 (Group 4 Address) Search Engine Group

12.3.11.1 AGETIME_LOW – MAC address aging time Low

I²C Address h049; CPU Address:h400

Accessed by CPU and I²C (R/W)

Used in conjuction with AGETIME_HIGH. The ZL50409 removes the MAC address from the data base and sends a Delete MAC Address Control Command to the CPU.

Bit[7:0]:	Low byte of the MAC address aging timer (Default 0x5C)

12.3.11.2 AGETIME_HIGH –MAC address aging time High

I²C Address h04A; CPU Address h401

Accessed by CPU and I²C (R/W)

Bit[7:0]:	High byte of the MAC address aging timer (Default 0)

The default setting of AGETIME_LOW/HIGH provides 300 seconds aging time. Aging time is based on the following equation:

{AGETIME_HIGH,AGETIME_LOW} X (# of MAC entries in the memory X 800µsec). Number of MAC entries = 4K.

12.3.11.3 SE_OPMODE – Search Engine Operation Mode

CPU Address:h403

Accessed by CPU (R/W)

Note: ECR2[2] enable/disable learning for each port.

Bit [0]:	Reserved. Must be 0.
Bit [1]:	Protocol filtering mode 0 – Inclusive (Default) 1 – Exclusive
Bit [2]:	Report control 1 – Disable report MAC address deletion 0 – Report MAC address deletion (MAC address is deleted from MCT after aging time) (Default)
Bit [3]:	Delete Control 1 – Disable aging logic from removing MAC during aging 0 – MAC address entry is removed when it is old enough to be aged (Default) However, a report is still sent to the CPU in both cases, when bit[2] = 0
Bit [4]:	Enable RSVP Packet trapping 1- Enable RSVP Packet trapping. IP Multicast also needs to be enabled for this function. 0- Disable RSVP Packet trapping.
Bit [5]	1 - Report ARP packet to CPU 0 - No ARP packet reporting (Default)
Bit [6]:	Disable MCT speed-up aging 1 – Disable speed-up aging when MCT resource is low. 0 – Enable speed-up aging when MCT resource is low. (Default)
Bit [7]:	Slow Learning 1– Enable slow learning. Learning is temporary disabled when search demand is high 0 – Learning is performed independent of search demand (Default)

12.3.12 (Group 5 Address) Buffer Control/QOS Group

12.3.12.1 QOSC – QOS Control

I²C Address h04B; CPU Address:h500

Accessed by CPU and I²C (R/W)

Bit [0]:	Enable TX rate control (on RMAC ports only) 1 – Enable 0 – Disable (Default)
Bit [1]:	Enable RX rate control (on RMAC ports only) 1 – Enable 0 – Disable (Default)
Bit [4:2]:	Reserved
Bit [5]:	Select VLAN tag or TOS (IP packets) to be preferentially picked to map transmit priority and drop priority 0 – Select VLAN Tag priority field over TOS (Default) 1 – Select TOS over VLAN tag priority field
Bit [6]:	Select TOS bits for Priority 0 – Use TOS [4:2] bits to map the transmit priority (Default) 1 – Use TOS [7:5] bits to map the transmit priority
Bit [7]:	Select TOS bits for Drop priority 0 – Use TOS [4:2] bits to map the drop priority (Default) 1 – Use TOS [7:5] bits to map the drop priority

12.3.12.2 UCC – Unicast Congestion Control

I²C Address h068, CPU Address: 510

Accessed by CPU and I²C (R/W)

Bit [7:0]:	Number of frame count. Used for best effort dropping at B% when destination port's best effort queue reaches UCC threshold and shared pool is all in use. Granularity is 16 granule (Default 0x6)

12.3.12.3 MCC – Multicast Congestion Control

I²C Address h069, CPU Address: 511

Accessed by CPU and I²C (R/W)

Bit [7:0]:	In multiples of 16 granules (granularity). Used for triggering MC flow control when destination port's multicast best effort queue reaches MCC threshold. (Default 0x6)

12.3.12.4 MCCTH – Multicast Threshold Control

CPU Address: 512

Accessed by CPU (R/W)

Bit [7:0]:	Threshold on the multicast granule count. Exceeding the threshold consider as multicast resource low and the new multicast will be dropped at B% or flow control is triggered if enabled. (Default: 0x3)

12.3.12.5 RDRC0 – WRED Rate Control 0

I²C Address 090, CPU Address 513

Accessed by CPU and I^2C (R/W)

Bits[3:0]:	Corresponds to the frame drop percentage Y% for WRED. Granularity 6.25%.
Bits[7:4]:	Corresponds to the frame drop percentage X% for WRED. Granularity 6.25%.
See Programming QoS Registers application note for more information	

12.3.12.6 RDRC1 – WRED Rate Control 1

I²C Address 091, CPU Address 514

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Corresponds to the best effort frame drop percentage B%, when shared pool is all in use and destination port best effort queue reaches UCC. Granularity 6.25%.
Bits[7:4]:	Corresponds to the frame drop percentage Z% for WRED. Granularity 6.25%.
See Programming QoS Registers application note for more information	

12.3.12.7 RDRC2 – WRED Rate Control 2

CPU Address 515

Accessed by CPU (R/W)

Bits[3:0]:	Corresponds to the frame drop percentage RB%, for rate control. Granularity 6.25%.
Bits[7:4]:	Corresponds to the frame drop percentage RA% for rate control. Granularity 6.25%.

12.3.12.8 SFCB – Share FCB Size

I²C Address h074, CPU Address 518

Accessed by CPU and I²C (R/W)

Bits [7:0]:	Expressed in multiples of 16 granules. Buffer reservation for shared pool.

12.3.12.9 C1RS – Class 1 Reserve Size

I²C Address h075, CPU Address 519

Accessed by CPU and I²C (R/W)

Bits [7:0]:	Class 1 FCB Reservation

Buffer reservation for class 1. Granularity 16 granules**. (Default 0)**

12.3.12.10 C2RS – Class 2 Reserve Size

I²C Address h076, CPU Address 51A

Accessed by CPU and I²C (R/W)

Bits [7:0]:	Class 2 FCB Reservation

Buffer reservation for class 2. Granularity 16 granules**. (Default 0)**

12.3.12.11 C3RS – Class 3 Reserve Size

I²C Address h077, CPU Address 51B

Accessed by CPU and I²C (R/W)

Bits [7:0]:	Class 3 FCB Reservation

Buffer reservation for class 3. Granularity 16 granules**. (Default 0)**

12.3.12.12 AVPML – VLAN Tag Priority Map

I²C Address h056; CPU Address:h530

Accessed by CPU and I²C (R/W)

Registers AVPML, AVPMM, and AVPMH allow the eight VLAN Tag priorities to map into eight Internal level transmit priorities. Under the Internal transmit priority, seven is the highest priority where as zero is the lowest. This feature allows the user the flexibility of redefining the VLAN priority field. For example, programming a value of 7 into bit 2:0 of the AVPML register would map packet VLAN priority 0 into Internal transmit priority 7. The new priority is used inside the 2604. When the packet goes out it carries the original priority.

Bit [2:0]:	Priority when the VLAN tag priority field is 0 (Default 0)
Bit [5:3]:	Priority when the VLAN tag priority field is 1 (Default 0)
Bit [7:6]:	Priority when the VLAN tag priority field is 2 (Default 0)

12.3.12.13 AVPMM – VLAN Priority Map

I²C Address h057, CPU Address:h531

Accessed by CPU and I²C (R/W)

Map VLAN priority into eight level transmit priorities:

Bit [0]:	Priority when the VLAN tag priority field is 2 (Default 0)
Bit [3:1]:	Priority when the VLAN tag priority field is 3 (Default 0)
Bit [6:4]:	Priority when the VLAN tag priority field is 4 (Default 0)
Bit [7]:	Priority when the VLAN tag priority field is 5 (Default 0)

12.3.12.14 AVPMH – VLAN Priority Map

I²C Address h058, CPU Address:h532

Accessed by CPU and I²C (R/W)

Map VLAN priority into eight level transmit priorities:

Bit [1:0]:	Priority when the VLAN tag priority field is 5 (Default 0)
Bit [4:2]:	Priority when the VLAN tag priority field is 6 (Default 0)
Bit [7:5]:	Priority when the VLAN tag priority field is 7 (Default 0)

12.3.12.15 AVDM – VLAN Discard Map

I²C Address h05C, CPU Address:h533

Accessed by CPU and I²C (R/W)

Map VLAN priority into frame discard when low priority buffer usage is above threshold

Bit [0]:	Frame drop priority when VLAN Tag priority field is 0 (Default 0)
Bit [1]:	Frame drop priority when VLAN Tag priority field is 1 (Default 0)
Bit [2]:	Frame drop priority when VLAN Tag priority field is 2 (Default 0)
Bit [3]:	Frame drop priority when VLAN Tag priority field is 3 (Default 0)
Bit [4]:	Frame drop priority when VLAN Tag priority field is 4 (Default 0)
Bit [5]:	Frame drop priority when VLAN Tag priority field is 5 (Default 0)
Bit [6]:	Frame drop priority when VLAN Tag priority field is 6 (Default 0)
Bit [7]:	Frame drop priority when VLAN Tag priority field is 7 (Default 0)

12.3.12.16 TOSPML – TOS Priority Map

I²C Address h059, CPU Address:h540

Accessed by CPU and I²C (R/W)

Map TOS field in IP packet into eight level transmit priorities

Bit [2:0]:	Priority when the TOS field is 0 (Default 0)
Bit [5:3]:	Priority when the TOS field is 1 (Default 0)
Bit [7:6]:	Priority when the TOS field is 2 (Default 0)

12.3.12.17 TOSPMM – TOS Priority Map

I²C Address h05A, CPU Address:h541

Accessed by CPU and I²C (R/W)

Map TOS field in IP packet into eight level transmit priorities

Bit [0]:	Priority when the TOS field is 2 (Default 0)
Bit [3:1]:	Priority when the TOS field is 3 (Default 0)
Bit [6:4]:	Priority when the TOS field is 4 (Default 0)
Bit [7]:	Priority when the TOS field is 5 (Default 0)

12.3.12.18 TOSPMH – TOS Priority Map

I²C Address h05B, CPU Address:h542

Accessed by CPU and I²C (R/W)

Map TOS field in IP packet into eight level transmit priorities:

Bit [1:0]:	Priority when the TOS field is 5 (Default 0)
Bit [4:2]:	Priority when the TOS field is 6 (Default 0)
Bit [7:5]:	Priority when the TOS field is 7 (Default 0)

12.3.12.19 TOSDML – TOS Discard Map

I²C Address h05D, CPU Address:h543

Accessed by CPU and I²C (R/W)

Map TOS into frame discard when low priority buffer usage is above threshold

Bit [0]:	Frame drop priority when TOS field is 0 (Default 0)
Bit [1]:	Frame drop priority when TOS field is 1 (Default 0)
Bit [2]:	Frame drop priority when TOS field is 2 (Default 0)
Bit [3]:	Frame drop priority when TOS field is 3 (Default 0)
Bit [4]:	Frame drop priority when TOS field is 4 (Default 0)
Bit [5]:	Frame drop priority when TOS field is 5 (Default 0)
Bit [6]:	Frame drop priority when TOS field is 6 (Default 0)
Bit [7]:	Frame drop priority when TOS field is 7 (Default 0)

12.3.12.20 USER_PROTOCOL_[7:0] – User Define Protocol 0~7

I²C Address h0B3-0BA, CPU Address:h550-557

Accessed by CPU and I²C (R/W)

(Default 00) This register is duplicated eight times from PROTOCOL 0~7 and allows the CPU to define eight separate protocols.

Bits[7:0]:	User Define Protocol

12.3.12.21 USER_PROTOCOL_FORCE_DISCARD[7:0] – User Define Protocol 0~7 Force Discard

I²C Address h0BB, CPU Address 558

Accessed by CPU and I²C (R/W)

Bits[0]:	Enable Protocol 0 Force Discard 1 – Enable 0 – Disable
Bits[1]:	Enable Protocol 1 Force Discard
Bits[2]:	Enable Protocol 2 Force Discard
Bits[3]:	Enable Protocol 3 Force Discard
Bits[4]:	Enable Protocol 4 Force Discard
Bits[5]:	Enable Protocol 5 Force Discard
Bits[6]:	Enable Protocol 6 Force Discard
Bits[7]:	Enable Protocol 7 Force Discard

12.3.13 User Defined Logical Ports and Well Known Ports

The ZL50409 supports classifying packet priority through layer 4 logical port information. It can be setup by 8 Well Known Ports, 8 User Defined Logical Ports, and 1 User Defined Range. The 8 Well Known Ports supported are:

- 23
- 512
- 6000
- 443
- 111
- 22555
- 22
- 554

Their respective priority can be programmed via Well_Known_Port [7:0] priority register. Well_Known_Port_ Enable can individually turn on/off each Well Known Port if desired.

Similarly, the User Defined Logical Port provides the user programmability to the priority, plus the flexibility to select specific logical ports to fit the applications. The 8 User Logical Ports can be programmed via User_Port 0-7 registers. Two registers are required to be programmed for the logical port number. The respective priority can be programmed to the User_Port [7:0] priority register. The port priority can be individually enabled/disabled via User_Port_Enable register.

The User Defined Range provides a range of logical port numbers with the same priority level. Programming is similar to the User Defined Logical Port. Instead of programming a fixed port number, an upper and lower limit need to be programmed, they are: {RHIGHH, RHIGHL} and {RLOWH, RLOWL} respectively. If the value in the upper limit is smaller or equal to the lower limit, the function is disabled. Any IP packet with a logical port that is less than the upper limit and more than the lower limit will use the priority specified in RPRIORITY.

12.3.13.1 WELL_KNOWN_PORT[1:0]_PRIORITY- Well Known Logic Port 1 and 0 Priority

I²C Address h0A8, CPU Address 560

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for Well known port 0 (23 for telnet)
Bits[7:4]:	Priority setting, transmission + dropping, for Well known port 1 (512 for TCP/UDP)

12.3.13.2 WELL_KNOWN_PORT[3:2]_PRIORITY- Well Known Logic Port 3 and 2 Priority

I²C Address h0A9, CPU Address 561

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for Well known port 2 (6000 for XWIN)
Bits[7:4]:	Priority setting, transmission + dropping, for Well known port 3 (443 for HTTP sec)

12.3.13.3 WELL_KNOWN_PORT[5:4]_PRIORITY- Well Known Logic Port 5 and 4 Priority

I²C Address h0AA, CPU Address 562

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for Well known port 4 (111 for sun remote procedure call)
Bits[7:4]:	Priority setting, transmission + dropping, for Well known port 5 (22555 for IP Phone call setup)

12.3.13.4 WELL_KNOWN_PORT[7:6]_PRIORITY- Well Known Logic Port 7 and 6 Priority

I²C Address h0AB, CPU Address 563

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for Well known port 6 (22 for ssh)
Bits[7:4]:	Priority setting, transmission + dropping, for Well known port 7 (554 for rtsp)

12.3.13.5 WELL_KNOWN_PORT_ENABLE[7:0] – Well Known Logic Port 0 to 7 Enables

I²C Address h0AC, CPU Address 564

Accessed by CPU and I²C (R/W)

Bits[0]:	Enable Well Known Port 0 Priority 1 – Enable 0 – Disable
Bits[1]:	Enable Well Known Port 1 Priority
Bits[2]:	Enable Well Known Port 2 Priority
Bits[3]:	Enable Well Known Port 3 Priority
Bits[4]:	Enable Well Known Port 4 Priority
Bits[5]:	Enable Well Known Port 5 Priority
Bits[6]:	Enable Well Known Port 6 Priority
Bits[7]:	Enable Well Known Port 7 Priority

12.3.13.6 WELL_KNOWN_PORT_FORCE_DISCARD[7:0] – Well Known Logic Port 0~7 Force Discard

I²C Address h0AD, CPU Address 565

Accessed by CPU and I²C (R/W)

Bits[0]:	Enable Well Known Port 0 Force Discard 1 – Enable 0 – Disable
Bits[1]:	Enable Well Known Port 1 Force Discard
Bits[2]:	Enable Well Known Port 2 Force Discard
Bits[3]:	Enable Well Known Port 3 Force Discard
Bits[4]:	Enable Well Known Port 4 Force Discard
Bits[5]:	Enable Well Known Port 5 Force Discard
Bits[6]:	Enable Well Known Port 6 Force Discard
Bits[7]:	Enable Well Known Port 7 Force Discard

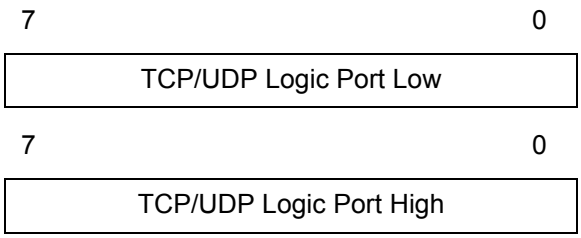
12.3.13.7 USER_PORT[7:0]_[LOW/HIGH] – User Define Logical Port 0~7

I²C Address h092-099(Low); CPU Address 570+2xN(Low) (N = logical port number)

I²C Address h09A-0A1(High); CPU Address 571+2xN(High)

Accessed by CPU and I²C (R/W)

(Default 00) This register is duplicated eight times from PORT 0 through PORT 7 and allows the CPU to define eight separate ports.



12.3.13.8 USER_PORT_[1:0]_PRIORITY - User Define Logic Port 1 and 0 Priority

I²C Address h0A2, CPU Address 590

Accessed by CPU and I²C (R/W)

The chip allows the CPU to define the priority

Bits[3:0]:	Priority setting, transmission + dropping, for logic port 0
Bits[7:4]:	Priority setting, transmission + dropping, for logic port 1 (Default 00)

12.3.13.9 USER_PORT_[3:2]_PRIORITY - User Define Logic Port 3 and 2 Priority

I²C Address h0A3, CPU Address 591

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for logic port 2
Bits[7:4]:	Priority setting, transmission + dropping, for logic port 3 (Default 00)

12.3.13.10 USER_PORT_[5:4]_PRIORITY - User Define Logic Port 5 and 4 Priority

I²C Address h0A4, CPU Address 592

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for logic port 4
Bits[7:4]:	Priority setting, transmission + dropping, for logic port 5 (Default 00)

12.3.13.11 USER_PORT_[7:6]_PRIORITY - User Define Logic Port 7 and 6 Priority

I²C Address h0A5, CPU Address 593

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Priority setting, transmission + dropping, for logic port 6
Bits[7:4]:	Priority setting, transmission + dropping, for logic port 7 (Default 00)

12.3.13.12 USER_PORT_ENABLE[7:0] – User Define Logic Port 0 to 7 Enables

I²C Address h0A6, CPU Address 594

Accessed by CPU and I²C (R/W)

Bits[0]:	Enable User Port 0 Priority 1 – Enable 0 – Disable
Bits[1]:	Enable User Port 1 Priority
Bits[2]:	Enable User Port 2 Priority
Bits[3]:	Enable User Port 3 Priority
Bits[4]:	Enable User Port 4 Priority
Bits[5]:	Enable User Port 5 Priority
Bits[6]:	Enable User Port 6 Priority
Bits[7]:	Enable User Port 7 Priority

12.3.13.13 USER_PORT_FORCE_DISCARD[7:0] – User Define Logic Port 0~7 Force Discard

I²C Address h0A7, CPU Address 595

Accessed by CPU and I²C (R/W)

Bits[0]:	Enable User Port 0 Force Discard 1 – Enable 0 – Disable
Bits[1]:	Enable User Port 1 Force Discard
Bits[2]:	Enable User Port 2 Force Discard
Bits[3]:	Enable User Port 3 Force Discard
Bits[4]:	Enable User Port 4 Force Discard
Bits[5]:	Enable User Port 5 Force Discard
Bits[6]:	Enable User Port 6 Force Discard
Bits[7]:	Enable User Port 7 Force Discard

12.3.13.14 RLOWL – User Define Range Low Bit 7:0

I²C Address h0AE, CPU Address: 5A0

Accessed by CPU and I²C (R/W)

Bits[7:0]:	Lower 8 bit of the User Define Logical Port Low Range

12.3.13.15 RLOWH – User Define Range Low Bit 15:8

I²C Address h0AF, CPU Address: 5A1

Accessed by CPU and I²C (R/W)

Bits[7:0]:	Upper 8 bit of the User Define Logical Port Low Range

12.3.13.16 RHIGHL – User Define Range High Bit 7:0

I²C Address h0B0, CPU Address: 5A2

Accessed by CPU and I²C (R/W)

Bits[7:0]:	Lower 8 bit of the User Define Logical Port High Range

12.3.13.17 RHIGHH – User Define Range High Bit 15:8

I²C Address h0B1, CPU Address: 5A3

Accessed by CPU and I²C (R/W)

Bits[7:0]:	Upper 8 bit of the User Define Logical Port High Range

12.3.13.18 RPRIORITY – User Define Range Priority

I²C Address h0B2, CPU Address: 5A4

Accessed by CPU and I²C (R/W)

RLOW and RHIGH form a range for logical ports to be classified with priority specified in RPRIORITY.

Bits[0]:	Drop Priority (inclusive only)
Bit[3:1]	Transmit Priority (inclusive only)
Bit[5:4]	Reserved
Bit[7:6]	00 - No Filtering 01 - Exclusive Filtering (x<=RLOW or x>=RHIGH) 10 - Inclusive Filtering (RLOW<x<RHIGH) 11 - Invalid

12.3.14 (Group 6 Address) MISC Group

12.3.14.1 MII_OP0 – MII Register Option 0

I²C Address 0BC, CPU Address:h600

Accessed by CPU and I²C (R/W)

Bit[4:0]:	Vendor specified link status register address (null value means don't use it) (Default 00). This is used if the Linkup bit position in the PHY is non-standard
Bit[6:5]	Reserved
Bits [7]:	Half duplex flow control feature 0 = Half duplex flow control always enable 1 = Half duplex flow control by negotiation

12.3.14.2 MII_OP1 – MII Register Option 1

I²C Address 0BD, CPU Address:h601

Accessed by CPU and I²C (R/W)

Bits[3:0]:	Duplex bit location in vendor specified register
Bits[7:4]:	Speed bit location in vendor specified register (Default 00)

12.3.14.3 FEN – Feature Register

I²C Address 0BE, CPU Address:h602)

Accessed by CPU and I²C (R/W)

Bits [0]:	Statistic Counter 0 – Disable (Default) 1 – Enable (all ports) When statistic counter is enable, an interrupt control frame is generated to the CPU, every time a counter wraps around. This feature requires an external CPU.
Bits[1]:	Reserved
Bit [2]:	Support DS EF Code. 0 – Disable (Default) 1 – Enable (all ports) When 101110 is detected in DS field (TOS[7:2]), the frame priority is set for 110 and drop is set for 0.
Bit [3]:	Enable VLAN ID hashing 0 – Disable (Default) 1 – Enable

Bit [4]:	Disable IP Multicast Support 0 – Enable IP Multicast Support (Must also set PVMODE[6]=1) 1 – Disable IP Multicast Support (Default) When enable, IGMP packets are identified by search engine and are passed to the CPU for processing. IP multicast packets are forwarded to the IP multicast group members according to the VLAN port mapping table.
Bit [5]:	Report to CPU 0 – Disable (Default) 1 – Enable When disable new VLAN port association report, new MAC address report or aging reports are disable for all ports. When enable, register SE_OPEMODE is used to enable/disable selectively each function.
Bit [6]:	MII Management State Machine 0: Enable (Default) 1: Disable This bit must be set so that there is no contention on the MDIO bus between MII Managment state machine and MIIC & MIID PHY register accesses.
Bit [7]:	MCT Link List structure 0 – Enable (Default) 1 – Disable

12.3.14.4 MIIC0 – MII Command Register 0

CPU Address:h603

Accessed by CPU (R/W)

Bits[7:0]:	MII Command Data [7:0]

Note: Before programming MII command: set FEN[6], check MIIC3, making sure no RDY, and no VALID; then program MII command.

12.3.14.5 MIIC1 – MII Command Register 1

CPU Address:h604

Accessed by CPU (R/W)

Bits[7:0]:	MII Command Data [15:8]

Note: Before programming MII command: set FEN[6], check MIIC3, making sure no RDY and no VALID; then program MII command.

12.3.14.6 MIIC2 – MII Command Register 2

CPU Address:h605

Accessed by CPU (R/W)

Bit [4:0]	REG_AD – Register PHY Address
Bit [6:5]	OP – Operation code “10” for read command and “01” for write command
Bit [7]	Reserved

Note: Before programming MII command: set FEN[6], check MIIC3, making sure no RDY and no VALID; then program MII command.

12.3.14.7 MIIC3 – MII Command Register 3

CPU Address:h606

Accessed by CPU (R/W)

Bits [4:0]	PHY_AD – 5 Bit PHY Address
Bit [5]	Reserved
Bit [6]	VALID – Data Valid from PHY (Read Only)
Bit [7]	RDY – Data is returned from PHY (Read Only)

Note: Before programming MII command: set FEN[6], check MIIC3, making sure no RDY and no VALID; then program MII command. Writing this register will initiate a serial management cycle to the MII management interface.

12.3.14.8 MIID0 – MII Data Register 0

CPU Address:h607

Accessed by CPU (RO)

Bits[7:0]:	MII Data [7:0]

12.3.14.9 MIID1 – MII Data Register 1

CPU Address:h608

Accessed by CPU (RO)

Bits[7:0]:	MII Data [15:8]

12.3.14.10 USD – One Micro Second Divider

CPU Address:h609

Accessed by CPU (R/W)

Bits[5:0]:	Divider to get one micro second from M_CLK (only used when not in standard RMII mode) In a MII or GPSI system, a 50MHz M_CLK may not be available. The system designer can decide to use another frequency on the M_CLK signal. To compensate for this, this register is required to be programmed. For example. If 20MHz is used on M_CLK, to compensate for the difference, this register is programmed with 20 to provide 1usec for internal reference.
Bits[7:6]:	Reserved

12.3.14.11 DEVICE Mode

CPU Address:h60A

Accessed by CPU (R/W)

Bit[0]:	Reserved
Bit[1]:	CPU Interrupt Polarity 0: Negative Polarity 1: Positive Polarity (Default)
Bit[4:2]:	Reserved
Bit [7:5]:	DEVICE ID (Default 0). This is for stacking operation. This is the stack ID for loop topology.

12.3.14.12 CHECKSUM - EEPROM Checksum

I²C Address 0FF, CPU Address:h60B

Accessed by CPU and I²C (R/W)

Bit [7:0]:	Checksum content (Default 0)

This register is used in unmanaged mode only. Before requesting that the ZL50409 updates the EEPROM device, the correct checksum needs to be calculated and written into this checksum register.

The checksum formula is:

$$\sum_{i=0}^{FF} I^2C \text{ register} = 0$$

When the ZL50409 boots from the EEPROM the checksum is calculated and the value must be zero. If the checksum is not zeroed the ZL50409 does not start and pin CHECKSUM_OK is set to zero.

12.3.14.13 LHBTimer – Link Heart Beat Timeout Timer

CPU Address:h610

Accessed by CPU (R/W)

In slot time (512 bit time). LHB packet will be sent out to the remote device if no other packet is transmitted in this period. The receiver will trigger LHB timeout interrupt if not receiving any good packet in two of this period.

12.3.14.14 LHBReg0, LHBReg1 - Link Heart Beat OpCode

CPU Address:h611, h612

Accessed by CPU (R/W)

The LHB frame uses MAC control frame format (same as flow control frame.) The register here defines the operation code. (flow control frame has h0001).

12.3.14.15 fMACCReg0, fMACCReg1

CPU Address:h613, h614

Accessed by CPU (R/W)

The registers define the operation code if MAC control frame is forced out by processor.

12.3.14.16 FCB Base Address Register 0

I²C Address 0BF, CPU Address:h620

Accessed by CPU and I²C (R/W)

Bit [7:0]	FCB Base address bit 7:0 (Default 0)

12.3.14.17 FCB Base Address Register 1

I²C Address 0C0, CPU Address:h621

Accessed by CPU and I²C (R/W)

Bit [7:0]	FCB Base address bit 15:8 (Default 0x60)

12.3.14.18 FCB Base Address Register 2

I²C Address 0C1, CPU Address:h622

Accessed by CPU and I²C (R/W)

Bit [7:0]	FCB Base address bit 23:16 (Default 0)

12.3.15 (Group 7 Address) Port Mirroring Group

12.3.15.1 MIRROR CONTROL – Port Mirror Control Register

CPU Address 70C

Accessed by CPU (R/W) (Default 00)

Bit [3:0]:	Destination port to be mirrored to.
Bit[4]	Mirror Flow from MIRROR_SRC_MAC[5:0] to MIRROR_DEST_MAC[5:0]
Bit[5]	Mirror Flow from MIRROR_DEST_MAC[5:0] to MIRROR_SRC_MAC[5:0]
Bit [6]:	Mirror when address is destination
Bit [7]:	Mirror when address is source

12.3.15.2 MIRROR_DEST_MAC[5:0] – Mirror Destination Mac Address 0~5

CPU Address 700-705

Accessed by CPU (R/W)

DEST_MAC5	DEST_MAC4	DEST_MAC3	DEST_MAC2	DEST_MAC1	DEST_MAC0
[47:40] (Default 00)	[39:32] (Default 00)	[31:24] (Default 00)	[23:16] (Default 00)	[15:8] (Default 00)	[7:0] (Default 00)

12.3.15.3 MIRROR_SRC _MAC[5:0] – Mirror Destination Mac Address 0~5

CPU Address 706-70B

Accessed by CPU (R/W)

SRC_MAC5	SRC_MAC4	SRC_MAC3	SRC_MAC2	SRC_MAC1	SRC_MAC0
[47:40] (Default 00)	[39:32] (Default 00)	[31:24] (Default 00)	[23:16] (Default 00)	[15:8] (Default 00)	[7:0] (Default 00)

12.3.15.4 RMAC_MIRROR0 – RMAC Mirror 0

CPU Address 710

Accessed by CPU (R/W)

Bit [2:0]:	Source port to be mirrored
Bit [3]:	Mirror path 0: Receive 1: Transmit
Bit [6:4]:	Destination port for mirrored traffic
Bit [7]:	Mirror enable

12.3.15.5 RMAC_MIRROR1 – RMAC Mirror 1

CPU Address 711

Accessed by CPU (R/W)

Bit [2:0]:	Source port to be mirrored
Bit [3]:	Mirror path 0: Receive 1: Transmit
Bit [6:4]:	Destination port for mirrored traffic
Bit [7]:	Mirror enable

12.3.16 (Group 8 Address) Per Port QOS Control

12.3.16.1 FCRn – Port 0~9 Flooding Control Register

I²C Address h04C-055; CPU Address:h800+N (N = port number)

Accessed by CPU and I²C (R/W)

Bit [3:0]:	U2MR: Unicast to Multicast Rate. Units in terms of time base defined in bits [6:4]. This is used to limit the amount of flooding traffic from Port N. The value in U2MR specifies how many packets are allowed to flood within the time specified by bit [6:4]. To disable this function, program U2MR to 0. (Default = 0)
Bit [6:4]:	Time Base for Unicast to Multicast, Multicast and Broadcast rate control of Port N: (Default = 000) 000 = 100us 001 = 200us 010 = 400us 011 = 800us 100 = 1.6ms 101 = 3.2ms 110 = 6.4ms 111 = 12.8ms
Bit [7]:	Reserved

12.3.16.2 BMRCn - Port 0~9 Broadcast/Multicast Rate Control

I²C Address h05E-067, CPU Address:h820+N (N = port number)

Accessed by CPU and I²C (R/W)

This broadcast and multicast rate defines for Port N, the number of packets allowed to be forwarded within a specified time. Once the packet rate is reached, packets will be dropped. To turn off the rate limit, program the field to 0. Time base is based on register FCR0 [6:4]

Bit [3:0]:	Multicast Rate Control. Number of multicast packets allowed within the time defined in bits 6 to 4 of the Flooding Control Register (FCRn). (Default 0).
Bit [7:4]:	Broadcast Rate Control. Number of broadcast packets allowed within the time defined in bits 6 to 4 of the Flooding Control Register (FCRn). (Default 0)

12.3.16.3 PR100_n – Port 0~7 Reservation

I²C Address h06A-071, CPU Address 840+N (N = port number)

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. (Default 0x6)

12.3.16.4 PR100_CPU – Port CPU Reservation

I²C Address h073, CPU Address 848

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. (Default 0x6)

12.3.16.5 PRM – Port MMAC Reservation

I²C Address h072, CPU Address 849

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. (Default 0x24)

12.3.16.6 PTH100_n – Port 0~7 Threshold

I²C Address h0C2-0C9, CPU Address 860+N (N = port number)

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. More than this number used on a source port will trigger ether random drop or flow control (Default 0x3)

12.3.16.7 PTH100_CPU – Port CPU Threshold

I²C Address h0CB, CPU Address 868

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. More than this number used on a source port will trigger ether random drop or flow control (Default 0x3)

12.3.16.8 PTHM – Port MMAC Threshold

I²C Address h0CA, CPU Address 869

Accessed by CPU and I²C (R/W)

Expressed in multiples of 16 granules. More than this number used on a source port will trigger ether random drop or flow control (Default 0x12)

12.3.16.9 QOSC00, QOSC01 - Classes Byte Limit port 0

Accessed by CPU and I²C (R/W)

- QOSC00 – BYTE_L1 (I²C Address h078, CPU Address 880)
- QOSC01 – BYTE_L2 (I²C Address h079, CPU Address 881)

Multiple of 16 granules. The two numbers set the two level for WRED on the high priority queue. When the queue size exceeds the L1 threshold, received frame will subject to X% (high drop) or Y% (low drop) WRED. When the queue size exceeds L2 threshold, received frame will ether be filtered (high drop) or subject to Z% WRED.

12.3.16.10 QOSC02, QOSC15 - Classes Byte Limit port 1-7

I²C Address 07A-087, CPU Address:h882-88F

Accessed by CPU and I²C (R/W)

Same as QOSC00, QOSC01

12.3.16.11 QOSC16 - QOSC21 - Classes Byte Limit CPU port

Accessed by CPU (R/W):

- QOSC16 – BYTE_L11 Level 1 for queue 1 (CPU Address 890)
- QOSC17 – BYTE_L21 Level 2 for queue 1 (CPU Address 891)
- QOSC18 – BYTE_L12 Level 1 for queue 2 (CPU Address 892)
- QOSC19 – BYTE_L22 Level 2 for queue 2 (CPU Address 893)
- QOSC20 – BYTE_L13 Level 1 for queue 3 (CPU Address 894)
- QOSC21 – BYTE_L23 Level 2 for queue 3 (CPU Address 895)

Multiple of 16 granules. The two numbers set the two level for WRED on the high priority queue. When the queue size exceeds the L1 threshold, received frame will subject to X% (high drop) or Y% (low drop) WRED. When the queue size exceeds L2 threshold, received frame will ether be filtered (high drop) or subject to Z% WRED.

12.3.16.12 QOSC22 - QOSC27 - Classes Byte Limit MMAC port

Accessed by CPU and I²C (R/W)

- QOSC22 – BYTE_L11 Level 1 for queue 1 (I²C Address h088, CPU Address 896)
- QOSC23 – BYTE_L21 Level 2 for queue 1 (I²C Address h089, CPU Address 897)
- QOSC24 – BYTE_L12 Level 1 for queue 2 (I²C Address h08A, CPU Address 898)
- QOSC25 – BYTE_L22 Level 2 for queue 2 (I²C Address h08B, CPU Address 899)
- QOSC26 – BYTE_L13 Level 1 for queue 3 (I²C Address h08C, CPU Address 89A)
- QOSC27 – BYTE_L23 Level 2 for queue 3 (I²C Address h08D, CPU Address 89B)

Multiple of 16 granules. The two numbers set the two level for WRED on the high priority queue. When the queue size exceeds the L1 threshold, received frame will subject to X% (high drop) or Y% (low drop) WRED. When the queue size exceeds L2 threshold, received frame will ether be filtered (high drop) or subject to Z% WRED.

12.3.16.13 QOSC28 - QOSC31 - Classes WFQ Credit For MMAC

Accessed by CPU (R/W)

W3 – QOSC28[5:0] – CREDIT_C00 (CPU Address 89C)

W2 – QOSC29[5:0] – CREDIT_C01 (CPU Address 89D)

W1 – QOSC30[5:0] – CREDIT_C02 (CPU Address 89E)

W0 – QOSC31[5:0] – CREDIT_C03 (CPU Address 89F)

QOSC28 through QOSC31 represents one set of WFQ parameters for MMAC port. The granularity of the numbers is 1, and their sum must be 64. QOSC31 corresponds to W0 that is the highest priority, and QOSC27 corresponds to W3. Default scheduling method will be strict priority across all queues. Only when the bit 7 in the class is set, the queue will be scheduled as WFQ. The credit number also works as shaper credit if bit 6 is set. The queue with shaper enabled will be scheduled by strict priority when the token is available. The shaper setting override the NS setting.

Bit [5:0]:	Class scheduling credit
Bit [6]:	Shaper enable
Bit [7]:	Not strict priority apply

12.3.16.14 QOSC36 - QOSC39 - Shaper Control Port MMAC

Accessed by CPU (R/W)

W3 – QOSC36[7:0] – TOKEN_LIMIT_C00 (CPU Address 8A4)

W2 – QOSC37[7:0] – TOKEN_LIMIT_C01 (CPU Address 8A5)

W1 – QOSC38[7:0] – TOKEN_LIMIT_C02 (CPU Address 8A6)

W0 – QOSC39[7:0] – TOKEN_LIMIT_C03 (CPU Address 8A7)

QOSC36 through QOSC39 represents one set of token limit on the shaper of MMAC port. The granularity of the numbers is 64 bytes. The shaper is implemented as leaky bucket and the limit here works as bucket size. Since the hardware implementation can keep negative number, the limit can be as small as one and still can transmit oversized frame, as long as one byte token is available.

12.3.17 (Group E Address) System Diagnostic

NOTE: Device Manufacturing test registers.

12.3.17.1 DTSRL – Test Output Selection

CPU Address E00

Accessed by CPU (R/W)

Test group selection for testout[7:0].

12.3.17.2 DTSRM – Test Output Selection

CPU Address E01

Accessed by CPU (R/W)

Test group selection for testout[15:8].

12.3.17.3 TESTOUT0, TESTOUT1 – Testmux Output [7:0], [15:8]

CPU Address E02, E03

Accessed by CPU (RO)

12.3.17.4 MASK0-MASK4 – Timeout Reset Mask

CPU Address E10-E14

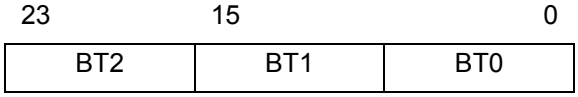
Accessed by CPU (R/W)

Disable timeout reset on selected state machine status.

12.3.17.5 BOOTSTRAP0 – BOOTSTRAP2

CPU Address E80-E82

Accessed by CPU (RO)



Bit [15:0]:	Bootstrap value from TSTOUT[15:0]: Bit [6:0]: TSTOUT[6:0] Bit [8:7]: Invert of TSTOUT[8:7] Bit [9]: TSTOUT[11] Bit [10]: TSTOUT[9] Bit [11]: TSTOUT[10] Bit [14:12]: TSTOUT[14:12] Bit [15]: Always 0
Bit [23:16]:	Bootstrap value from M[7:0]_TXEN Bit [16]: M0_TXEN Bit [17]: M1_TXEN ... Bit [23]: M7_TXEN

12.3.17.6 PRTFSMST0 – PRTFSMST9

CPU Address E90-E99

Accessed by CPU (RO)

Bit [0]:	TX FSM NOT idle for 5 sec
Bit [1]:	TX FIFO control NOT idle for 5 sec
Bit [2]:	RX SFD detection NOT idle for 5 sec
Bit [3]:	RXINF NOT idle for 5 sec
Bit [4]:	PTCTL NOT idle for 5 sec
Bit [5]:	Reserved
Bit [6]:	LHB frame detected
Bit [7]:	LHB receiving timeout

12.3.17.7 PRTQOSST0-PRTQOSST7

CPU Address EA0 - EA7

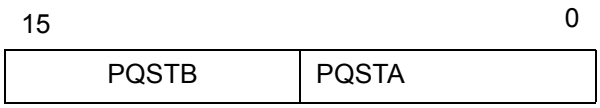
Accessed by CPU (RO)

Bit [0]:	Source port reservation low
Bit [1]:	No source port buffer left
Bit [2]:	Unicast congestion detected on best effort queue
Bit [3]:	Reserved
Bit [4]:	High priority queue reach L1 WRED level
Bit [5]:	High priority queue reach L2 WRED level
Bit [6]:	Low priority MC queue full
Bit [7]:	High priority MC queue full

12.3.17.8 PRTQOSST8A, PRTQOSST8B (CPU port)

CPU Address EA8 – EA9

Accessed by CPU (RO)

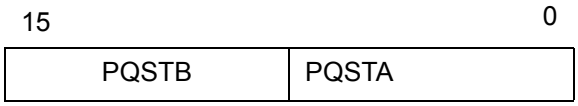


Bit [0]:	Source port reservation low
Bit [1]:	No source port buffer left
Bit [2]:	Unicast congestion detected on best effort queue
Bit [3]:	Reserved
Bit [4]:	priority queue 1 reach L1 WRED level
Bit [5]:	priority queue 1 reach L2 WRED level
Bit [6]:	priority queue 2 reach L1 WRED level
Bit [7]:	priority queue 2 reach L2 WRED level
Bit [8]:	priority queue 3 reach L1 WRED level
Bit [9]:	priority queue 3 reach L2 WRED level
Bit [10]:	priority 0 MC queue full
Bit [11]:	priority 1 MC queue full
Bit [12]:	priority 2 MC queue full
Bit [13]:	Priority 3 MC queue full
Bit [15:14]:	Reserved

12.3.17.9 PRTQOSST9A, PRTQOSST9B (MMAC port)

CPU Address EAA – EAB

Accessed by CPU (RO)



Bit [0]:	Source port reservation low
Bit [1]:	No source port buffer left
Bit [2]:	Unicast congestion detected on best effort queue
Bit [3]:	Reserved
Bit [4]:	Priority queue 1 reach L1 WRED level
Bit [5]:	Priority queue 1 reach L2 WRED level
Bit [6]:	Priority queue 2 reach L1 WRED level
Bit [7]:	Priority queue 2 reach L2 WRED level
Bit [8]:	Priority queue 3 reach L1 WRED level
Bit [9]:	Priority queue 3 reach L2 WRED level
Bit [10]:	Priority 0 MC queue full
Bit [11]:	Priority 1 MC queue full
Bit [12]:	Priority 2 MC queue full
Bit [13]:	Priority 3 MC queue full
Bit [15:14]:	Reserved

12.3.17.10 CLASSQOSST

CPU Address EAC

Accessed by CPU (RO)

Bit [0]:	No share buffer
Bit [1]:	No class 1 buffer
Bit [2]:	No class 2 buffer
Bit [3]:	No class 3 buffer
Bit [7:4]:	Reserved

12.3.17.11 PRTINTCTR

CPU Address EAD

Accessed by CPU (R/W)

Bit [0]:	Interrupt when source buffer low
Bit [1]:	Interrupt when no source buffer
Bit [2]:	Interrupt when UC congest
Bit [3]:	Interrupt when L1 WRED level
Bit [4]:	Interrupt when L2 WRED level
Bit [5]:	Interrupt when MC queue full
Bit [6]:	Interrupt when LHB timeout
Bit [7]:	Interrupt when no class buffer

12.3.17.12 QMCTRL[9:0]

CPU Address EB0 – EB9

Accessed by CPU (R/W)

Bit [0]:	Suspend port scheduling (no departure)
Bit [1]:	Reset queue
Bit [4:2]:	Reserved
Bit [5]:	Force out MAC control frame
Bit [6]:	Force out XOFF flow control frame
Bit [7]:	Force out XON flow control frame

12.3.17.13 QCTRL

CPU Address EBA

Accessed by CPU (R/W)

Bit [0]:	Stop QM FSM at idle
Bit [1]:	Stop MCQ FSM at idle
Bit [2]:	Stop new granule grant to any source
Bit [3]:	Stop release granule from any source
Bit [7:4]:	Reserved

12.3.17.14 BMBISTR0, BMBISTR1

CPU Address EBB, EBC

Accessed by CPU (RO)

12.3.17.15 BMControl

CPU Address EBD

Accessed by CPU (R/W)

Bit [3:0]:	Block Memory redundancy control 0: Use hardware detected value All others: Overwrite the hardware detected memory swap map
Bit [7:4]:	Reserved

12.3.17.16 BUFF_RST

CPU Address EC0

Accessed by CPU (R/W)

Bit [3:0]	Assign a value that the pool to be reset 0: port 0 pool 1: port 1 pool 2: port 2 pool 3: port 3 pool 4: port 4 pool 5: port 5 pool 6: port 6 pool 7: port 7 pool 8: port MMAC pool 9: shared pool 10: class 1 pool 11: class 2 pool 12: class 3 pool 13: multicast pool 14: cpu pool 15: reserved
Bit [4]	If this bit is 1, then all the pools are assigned
Bit [5]	Set 1 to reset the pools that are assigned
Bit [7:6]	Reserved

If CPU wants to reset pools again, CPU has to clear bit 5 and then set bit 5.

Note: Before CPU doing so, CPU should set QCTRL (CPU Address EBA) bit 2 and bit 3 to one. After reset the pools, CPU shall reprogram free granule link list (CPU address EC1, EC2, EC3, EC4, EC5, EC6). Then clear QCTRL (EBA).

12.3.17.17 FCB_HEAD_PTR0, FCB_HEAD_PTR1

CPU address EC1

Accessed by CPU (R/W)

Bit [7:0]	Fcb_head_ptr[7:0]. The head pointer of free granule link that CPU assigns.

CPU address EC2

Accessed by CPU (R/W)

Bit [6:0]	Fcb_head_ptr[14:8]. The head pointer of free granule link that CPU assigns.
Bit [7]	Set 1 to write

If CPU wants to write again, CPU has to clear bit 15 and then set bit 15.

12.3.17.18 FCB_TAIL_PTR0, FCB_TAIL_PTR1

CPU address EC3

Accessed by CPU (R/W)

Bit [7:0]	Fcb_tail_ptr[7:0]. The tail pointer of free granule link that CPU assigns.

CPU address EC4

Accessed by CPU (R/W)

Bit [6:0]	Fcb_tail_ptr[14:8]. The tail pointer of free granule link that CPU assigns.
Bit [7]	Set 1 to write

If CPU wants to write again, CPU has to clear bit 15 and then set bit 15.

12.3.17.19 FCB_NUM0, FCB_NUM1

CPU address EC5

Accessed by CPU (R/W)

Bit [7:0]	Fcb_number[7:0]. The total number of granules that CPU assigns.

CPU address EC6

Accessed by CPU (R/W)

Bit [6:0]	Fcb_number[14:8]. The total number of granules that CPU assigns.
Bit [7]	Set 1 to write

If CPU wants to write again, CPU has to clear bit 15 and then set bit 15.

Note: There are two ways to reprogram the free granules.

1. CPU links all the granules: CPU writes memory directly, at last write head pointer (address EC1, EC2), tail pointer (address EC3, EC4) and granule number (address EC5, EC6).
2. CPU tells Buffer Manager to link: CPU clear head pointer (address EC1, EC2), clear tail pointer (address EC3, EC4), then write granule number that tells Buffer Manager to link (address EC5, EC6).

12.3.17.20 BM_RLSFF_CTRL

CPU address EC7

Accessed by CPU (R/W)

Bit [0]	Read BM release FIFO.
Bit [7:1]	Reserved

The information of BM release FIFO is relocated to registers BM_RLSFF_INFO (address ECD, ECC, ECB, ECA, EC9 and EC8). If the FIFO is not empty, CPU can read out the next by setting the bit 0. Read only happens when bit 0 is changing from 0 to 1.

12.3.17.21 BM_RSLFF_INFO[5:0]

CPU address EC8

Accessed by CPU (RO)

Bit [7:0]	Rls_head_ptr[7:0].

CPU address EC9

Accessed by CPU (RO)

Bit [6:0]	Rls_head_ptr[14:8].
Bit [7]	Rls_tail_ptr[0]

CPU address ECA

Accessed by CPU (RO)

Bit [7:0]	Rls_tail_ptr[8:1]

CPU address ECB

Accessed by CPU (RO)

Bit [5:0]	Rls_tail_ptr[14:9]
Bit [7:6]	Rls_count[1:0]

CPU address ECC

Accessed by CPU (RO)

Bit [4:0]	Rls_count[6:2]
Bit [5]	If 1, then It is multicast packet.
Bit [7:6]	Rls_src_port[1:0[

CPU address ECD

Accessed by CPU (RO)

Bit [1:0]	Rls_src_port[3:2]
Bit [3:2]	Class[1:0]
Bit [4]	This release request is from QM directly.
Bit [7:5]	Entries count in release FIFO, 0 means FIFO is empty

12.3.18 (Group F Address) CPU Access Group

12.3.18.1 GCR - Global Control Register

CPU Address: hF00

Accessed by CPU (R/W)

Bit [0]:	Store configuration (Default = 0) Write ‘1’ followed by ‘0’ to store configuration into external EEPROM
Bit[1]:	Store configuration and reset (Default = 0) Write ‘1’ to store configuration into external EEPROM and reset chip
Bit[2]:	Start BIST (Default = 0) Write ‘1’ followed by ‘0’ to start the device’s built-in self-test. The result is found in the DCR register.
Bit[3]:	Soft Reset (Default = 0) Write ‘1’ to reset chip
Bit[4]:	Initialization Completed (Default = 0) This bit is reserved in unmanaged mode. In managed mode, the CPU writes this bit with ‘1’ to indicate initialization is completed and ready to forward packets. The '0' to '1' transition will toggle TSTOUT[2] from low to high.
Bit[7:5]:	Reserved

12.3.18.2 DCR - Device Status and Signature Register

CPU Address: hF01

Accessed by CPU (RO)

Bit [0]:	1: Busy writing configuration to I²C 0: Not busy (not writing configuration to I²C)
Bit[1]:	1: Busy reading configuration from I²C 0: Not busy (not reading configuration from I²C)
Bit[2]:	1: BIST in progress 0: BIST not running
Bit[3]:	1: RAM Error 0: RAM OK
Bit[5:4]:	Device Signature 10: ZL50409 device
Bit [7:6]:	Revision 00: Initial Silicon 01: Second Silicon

12.3.18.3 DCR1 - Device Status Register 1

CPU Address: hF02

Accessed by CPU (RO)

Bit [6:0]	Reserved
Bit [7]	Chip initialization completed

12.3.18.4 DPST – Device Port Status Register

CPU Address:hF03

Accessed by CPU (R/W)

Bit[4:0]:	Read back index register. This is used for selecting what to read back from DTST. (Default 00) - 5'b00000 - Port 0 Operating mode and Negotiation status - 5'b00001 - Port 1 Operating mode and Negotiation status - 5'b00010 - Port 2 Operating mode and Negotiation status - 5'b00011 - Port 3 Operating mode and Negotiation status - 5'b00100 - Port 4 Operating mode and Negotiation status - 5'b00101 - Port 5 Operating mode and Negotiation status - 5'b00110 - Port 6 Operating mode and Negotiation status - 5'b00111 - Port 7 Operating mode and Negotiation status - 5'b01000 - Port CPU Operating mode and Negotiation status - 5'b01001 - Port MMAC Operating mode and Negotiation status
Bit[7:5]:	Reserved

12.3.18.5 DTST – Data read back register

CPU Address: hF04

Accessed by CPU (RO)

This register provides various internal information as selected in DPST bit[4:0]. Refer to the PHY Control Application Note.

Bit[0]	Flow control enable
Bit[1]	Full duplex port
Bit[2]	Fast Ethernet port
Bit[3]	Link is down
Bit[4]	Auto negotiation enabled 1: Disable 0: Enable
Bit[5:6]	Reserved
Bit[7]	Module detected (for hot swap purpose)

12.3.18.6 DA – DA Register

CPU Address: hFFF

Accessed by CPU (RO)

Always return 8'h **DA**. Indicate the CPU interface or serial port connection is good.

Bit[7:0]	Always return DA

13.0 Characteristics and Timing

13.1 Absolute Maximum Ratings

Storage Temperature	-65C to +150C
Operating Temperature	-40C to +85C
Supply Voltage V_{CC} with Respect to V_{SS}	+3.0 V to +3.6 V
Supply Voltage V_{DD} with Respect to V_{SS}	+1.70 V to +2.00 V
Voltage on 5V Tolerant Input Pins	-0.5 V to (V_{CC} + 2.5 V)
Voltage on Other Pins	-0.5 V to (V_{DD} + 0.3 V)

Caution: Stress above those listed may damage the device. Exposure to the Absolute Maximum Ratings for extended periods may affect device reliability. Functionality at or above these limits is not implied.

13.2 DC Electrical Characteristics

V_{CC} = 3.0 V to 3.6 V (3.3v +/- 10%) $T_{AMBIENT}$ = -40 C to +85 C

V_{DD} = 1.8V +5% - 5%

13.3 Recommended Operating Conditions

Symbol	Parameter Description				
		Min	Typ	Max	Unit
f_{osc}	Frequency of Operation		45-50		MHz
I_{CC}	V_{CC} Supply Current – @ 50 MHz			50	mA
I_{DD}	V_{DD} Supply Current – @ 50 MHz			180	mA
V_{OH}	Output High Voltage (CMOS)	V_{CC} - 0.5			V
V_{OL}	Output Low Voltage (CMOS)			0.5	V
V_{IH}	Input High Voltage (TTL 5V tolerant)	V_{CC} x 70%		V_{CC} + 2.0	V
V_{IL}	Input Low Voltage (TTL 5V tolerant)			V_{CC} x 30%	V
I_{IL}	Input Leakage Current (0.1 V < V_{IN} < V_{CC}) (all pins except those with internal pull-up/pull-down resistors)			10	µA
I_{OL}	Output Leakage Current (0.1 V < V_{OUT} < V_{CC})			10	µA
C_{IN}	Input Capacitance			5	pF
C_{OUT}	Output Capacitance			5	pF
$C_{I/O}$	I/O Capacitance			7	pF
θ_{ja}	Thermal resistance with 0 air flow			24.3	C/W
θ_{ja}	Thermal resistance with 1 m/s air flow			20.0	C/W
θ_{ja}	Thermal resistance with 2m/s air flow			18.1	C/W
θ_{jc}	Thermal resistance between junction and case			4.6	C/W

13.4 AC Characteristics and Timing

13.4.1 Typical Reset & Bootstrap Timing Diagram

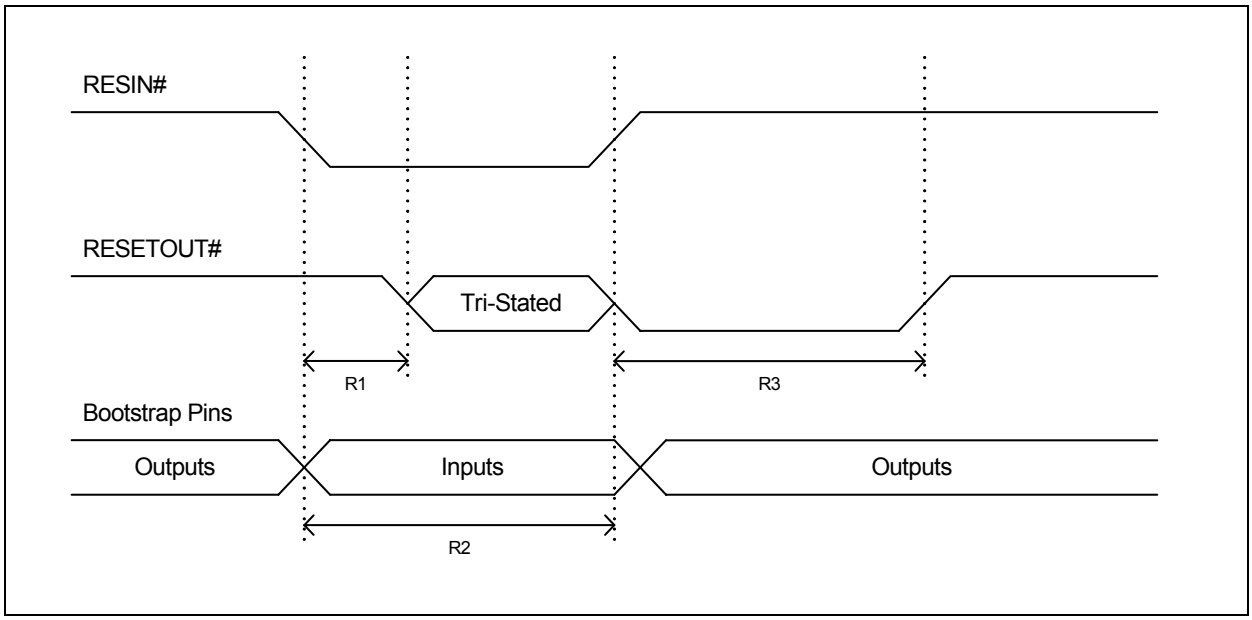


Figure 12 - Typical Reset & Bootstrap Timing Diagram

Symbol	Parameter	Min	Typ	Note:
R1	Delay until RESETOUT# is tri-stated		10ns	RESETOUT# state is then determined by the external pull-up/down resistor
R2	Bootstrap stabilization	1µs	10µs	Bootstrap pins sampled on rising edge of RESIN#
R3	RESETOUT# assertion		2ms	

Table 10 - Reset & Bootstrap Timing

13.4.2 Typical CPU Timing Diagram for a CPU Write Cycle

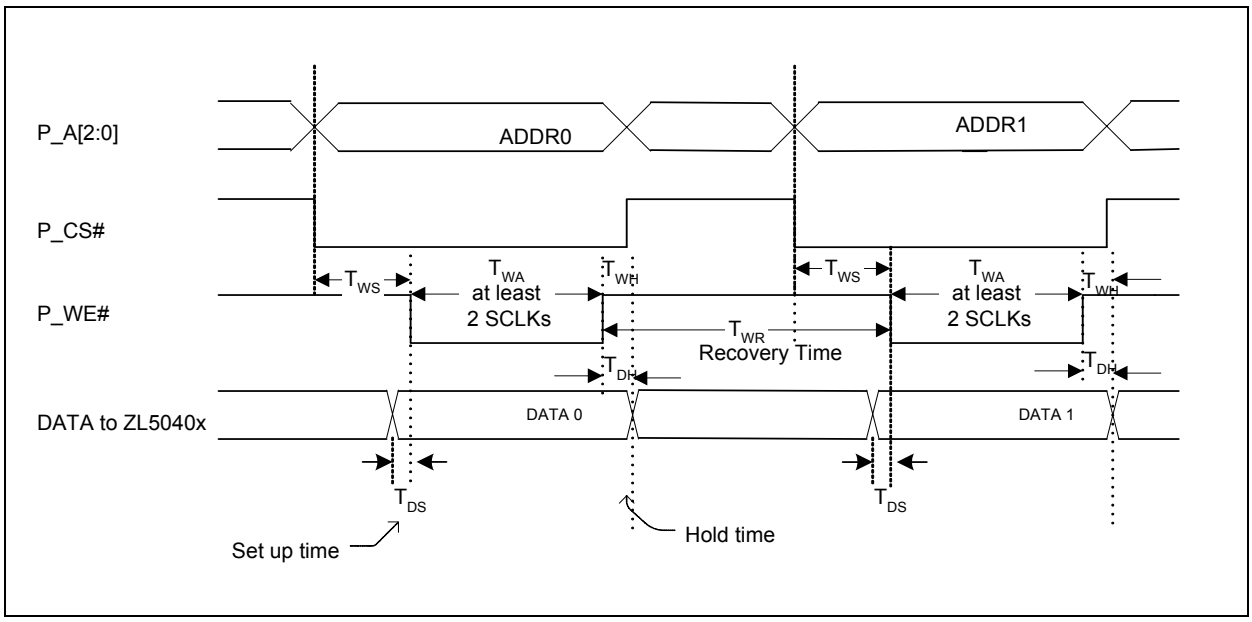


Figure 13 - Typical CPU Timing Diagram for a CPU Write Cycle

Description		(SCLK=100Mhz)		(SCLK=50Mhz)		Refer to Figure 13
Write Cycle	**Symbol**	**Min**	**Max**	**Min**	**Max**	
Write Set up Time	T_{WS}	10		10		
Write Active Time	T_{WA}	20		40		At least 2 SCLK
Write Hold Time	T_{WH}	2		2		
Write Recovery time	T_{WR}	30		60		At least 3 SCLK
Data Set Up time	T_{DS}	10		10		
Data Hold time	T_{DH}	2		2		

Table 11 - AC Characteristics - CPU Write Cycle

13.4.3 Typical CPU Timing Diagram for a CPU Read Cycle

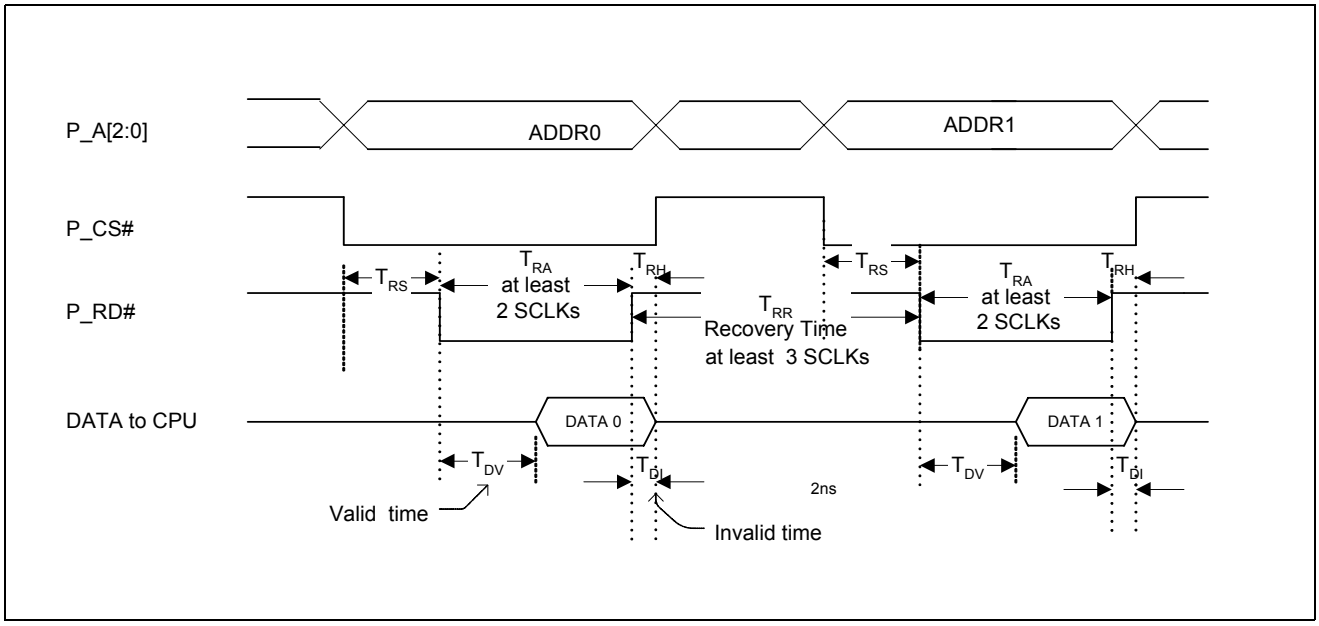


Figure 14 - Typical CPU Timing Diagram for a CPU Read Cycle

Description		(SCLK=100Mhz)		(SCLK=50Mhz)		Refer to Figure 14
Read Cycle	**Symbol**	**Min**	**Max**	**Min**	**Max**	
Read Set up Time	T_{RS}	10		10		
Read Active Time	T_{RA}	20		40		At least 2 SCLK
Read Hold Time	T_{RH}	2		2		
Read Recovery time	T_{RR}	30		60		At least 3 SCLK
Data Valid time	T_{Dv}		12		12	
Data Invalid time	T_{DI}		10		10	

Table 12 - AC Characteristics - CPU Read Cycle

13.4.4 Reduced Media Independent Interface

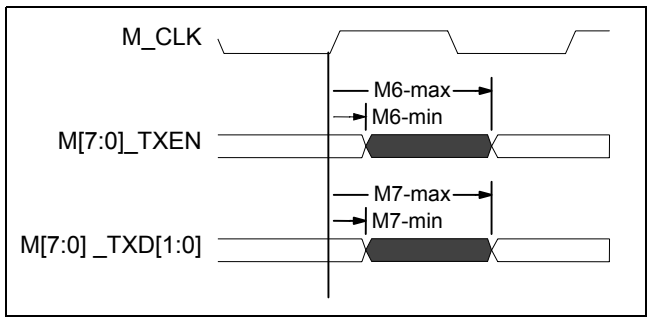


Figure 15 - AC Characteristics – Reduced media independent Interface (TX)

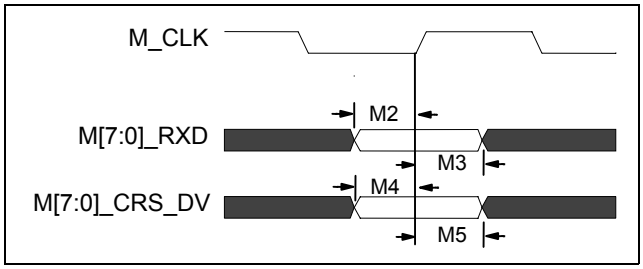


Figure 16 - AC Characteristics – Reduced Media Independent Interface (RX)

Symbol	Parameter	50MHz		Note:
		Min (ns)	Max (ns)	
M2	M[7:0]_RXD[1:0] Input Setup Time	4		
M3	M[7:0]_RXD[1:0] Input Hold Time	2		
M4	M[7:0]_CRS_DV Input Setup Time	4		
M5	M[7:0]_CRS_DV Input Hold Time	3		
M6	M[7:0]_TXEN Output Delay Time	2	11	C_L = 20 pF
M7	M[7:0]_TXD[1:0] Output Delay Time	2	11	C_L = 20 pF

Table 13 - AC Characteristics – Reduced Media Independent Interface

13.4.5 Media Independent Interface

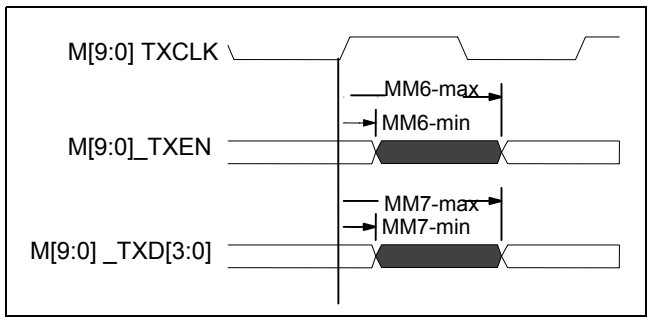


Figure 17 - AC Characteristics – Media independent Interface (TX)

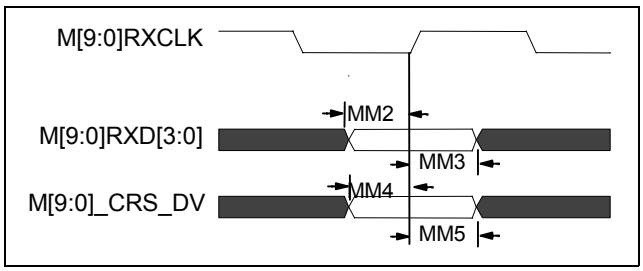


Figure 18 - AC Characteristics – Media Independent Interface (RX)

Symbol	Parameter	-25MHz		Note:
		Min (ns)	Max (ns)	
MM2	M[9,7:0]_RXD[3:0] Input Setup Time	4		
MM2	M[8]_RXD[3:0] Input Setup Time	10		CPU MII Interface
MM3	M[9:0]_RXD[3:0] Input Hold Time	2		
MM4	M[9,7:0]_CRS_DV Input Setup Time	4		
MM4	M[8]_CRS_DV Input Setup Time	10		CPU MII Interface
MM5	M[9:0]_CRS_DV Input Hold Time	2		
MM6	M[9:0]_TXEN Output Delay Time	2	14	C_L = 20 pF
MM7	M[9:0]_TXD[3:0] Output Delay Time	2	14	C_L = 20 pF

Table 14 - AC Characteristics –Media Independent Interface

13.4.6 General Purpose Serial Interface (7-wire)

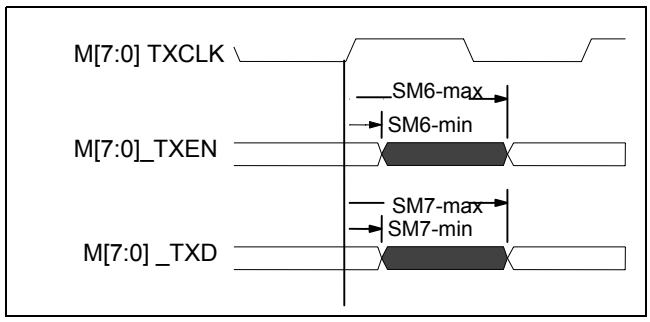


Figure 19 - AC Characteristics – General Purpose Serial Interface (TX)

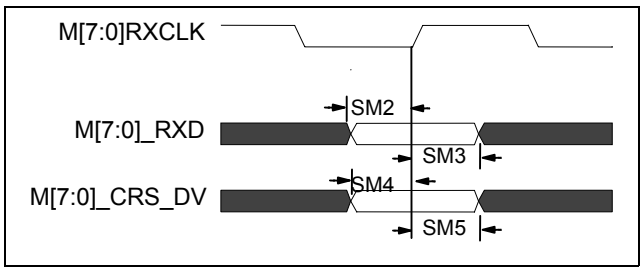


Figure 20 - AC Characteristics – General Purpose Serial Interface (RX)

Symbol	Parameter	-10MHz		Note:
		Min (ns)	Max (ns)	
SM2	M[7:0]_RXD Input Setup Time	4		
SM3	M[7:0]_RXD Input Hold Time	2		
SM4	M[7:0]_CRS_DV Input Setup Time	4		
SM5	M[7:0]_CRS_DV Input Hold Time	2		
SM6	M[7:0]_TXEN Output Delay Time	2	14	C_L = 20 pF
SM7	M[7:0]_TXD Output Delay Time	2	14	C_L = 20 pF

Table 15 - AC Characteristics –General Purpose Serial Interface

13.4.7 MDIO Input Setup and Hold Timing

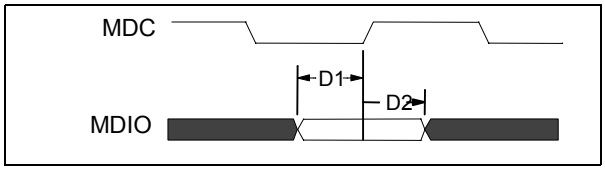


Figure 21 - MDIO Input Setup and Hold Timing

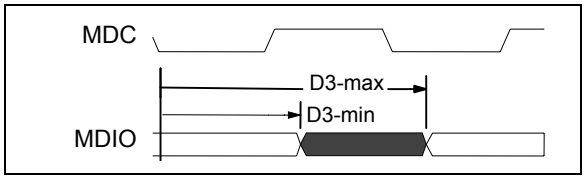


Figure 22 - MDIO Output Delay Timing

Symbol	Parameter	500KHz		Note:
		Min (ns)	Max (ns)	
D1	MDIO input setup time	10		
D2	MDIO input hold time	2		
D3	MDIO output delay time	1	20	C_L = 50pf

Table 16 - MDIO Timing

13.4.8 I²C Input Setup Timing

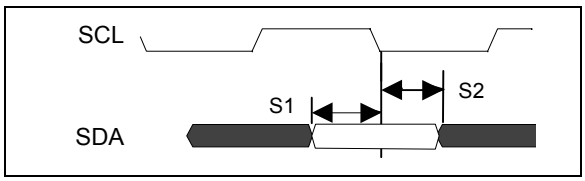


Figure 23 - I²C Input Setup Timing

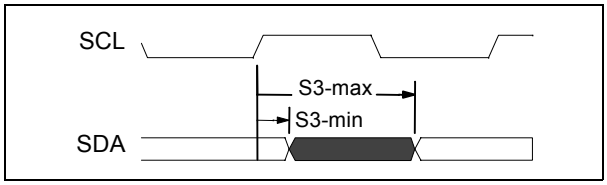


Figure 24 - I²C Output Delay Timing

Symbol	Parameter	50KHz		Note:
		Min (ns)	Max (ns)	
S1	SDA input setup time	20		
S2	SDA input hold time	1		
S3*	SDA output delay time	4 usec	6 usec	C_L = 30pf
* Open Drain Output. Low to High transistor is controlled by external pullup resistor.				

Table 17 - I²C Timing

13.4.9 Serial Interface Setup Timing

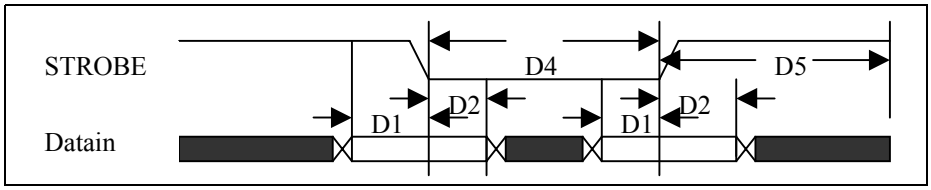


Figure 25 - Serial Interface Setup Timing

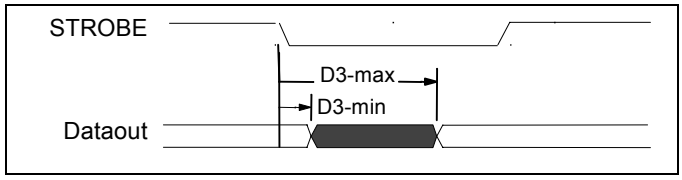


Figure 26 - Serial Interface Output Delay Timing

Symbol	Parameter	Min (ns)	Max (ns)	Note:
D1	Datain setup time	20		
D2	Datain hold time	3μs		Debounce on
D2	Datain hold time	20ns		Debounce off
D3	Dataout output delay time	1	50	C_L = 100pf
D4	Strobe low time	5μs		Debounce on
D4	Strobe low time	50ns		Debounce off
D5	Strobe high time	5μs		Debounce on
D5	Strobe high time	50ns		Debounce off

Table 18 - Serial Interface Timing

13.4.10 JTAG (IEEE 1149.1-2001)

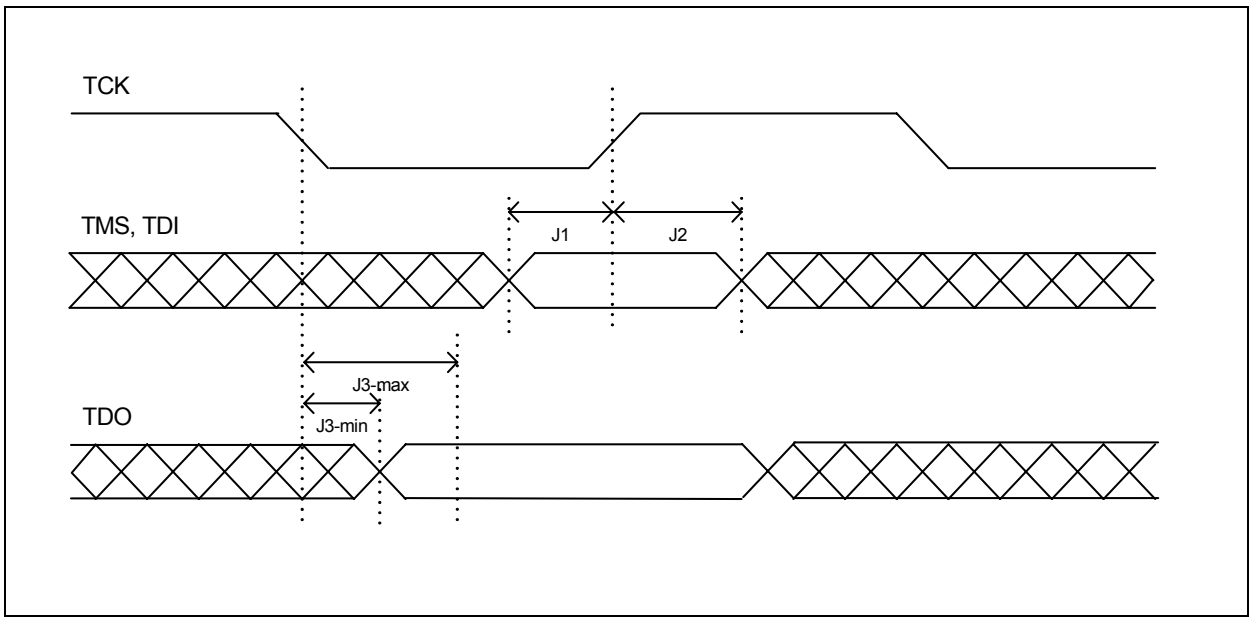
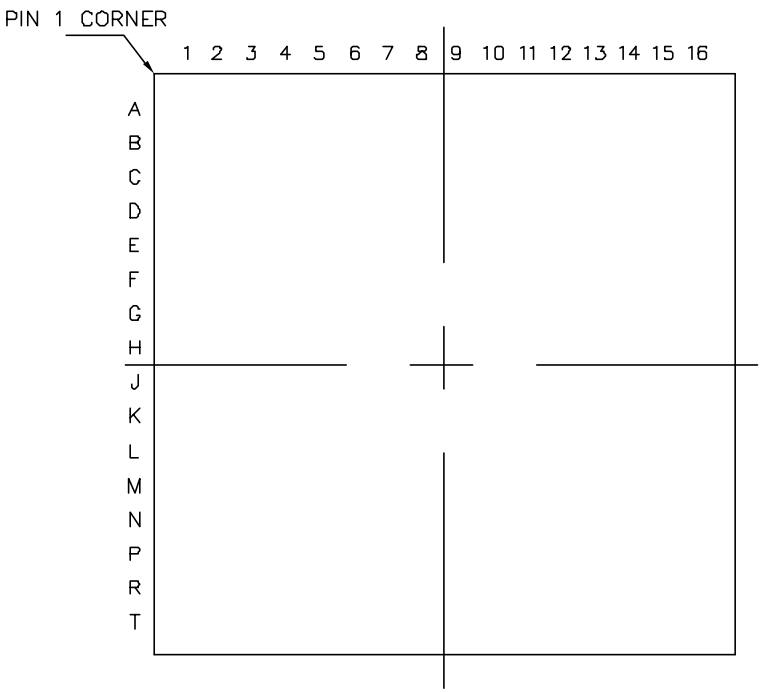


Figure 27 - JTAG Timing Diagram

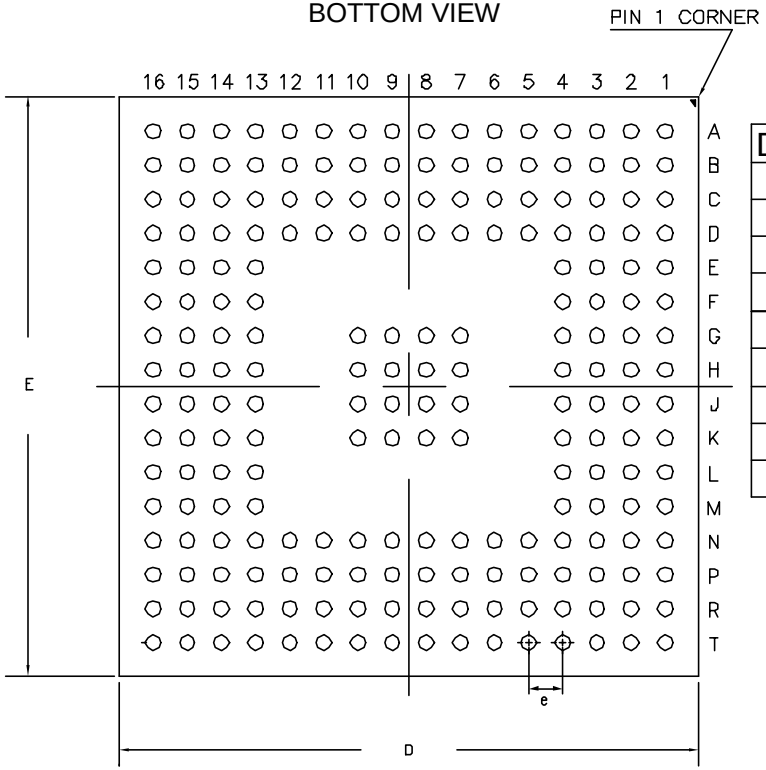
Symbol	Parameter	Min	Typ	Max	Units	Note:
	TCK frequency of operation	0	10	50	MHz	
	TCK cycle time	20			ns	
	TCK clock pulse width	10			ns	
	TRST# assert time	20		-	ns	TRST is an asynchronous signal
J1	TMS, TDI data setup time	3			ns	
J2	TMS, TDI data hold time	7			ns	
J3	TCK to TDO data valid	0		15	ns	

Table 19 - JTAG Timing

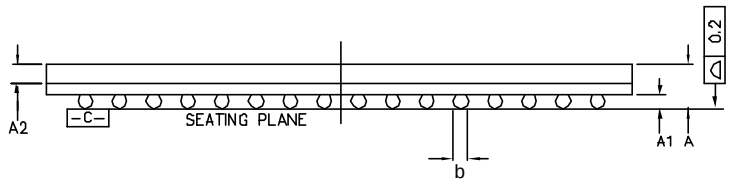
TOP VIEW



BOTTOM VIEW



Dimension	MIN	MAX
A	-	1.40
A1	0.30	0.50
A2	0.53 REF	
D	16.90	17.10
E	16.90	17.10
b	0.40	0.60
e	1.00	
N	208	
Conforms to JEDEC MO-192		



SIDE VIEW

NOTES: –
1. Controlling dimensions are in MM.
2. Seating plane is defined by the spherical crown of the solder balls.
3. Not to scale.
4. N is the number of solder balls
5. Substrate thickness is 0.36 MM.

ISSUE	1			
ACN	213730			
DATE	14Nov02			
APPRD.				

ZARLINK SEMICONDUCTOR

Previous package codes

Package Code GD

Package Outline for
208ball LBGA
17x17x1.4mm Max

GPD00802

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

www.zarlink.com

TECHNICAL DOCUMENTATION - NOT FOR RESALE