VT6516

16/12-PORT 10/100BASE-T/TX ETHERNET SWITCH CONTROLLER

REVISION 'E' DATASHEET
(Preliminary)

ISSUE 1: July 31, 1999

VIA Technologies, Inc.

PRELIMINARY RELEASE

Please contact VIA Technologies for the latest documentation.

Copyright Notice:

Disclaimer Notice:

Offices:

1045 Mission Court	8th Floor, No. 533
Fremont, CA 94539	Chung-Cheng Rd., Hsin-Tien
USA	Taipei, Taiwan ROC
Tel: (510) 683-3300	Tel: (886-2) 2218-5452
Fax: (510) 683-3301	Fax: (886-2) 2218-5453

Online Services:

BBS : 886-2-2186408
FTP : FTP.VIA.COM.TW
HTTP:**WWW.VIA.COM.TW** –or- WWW.VIATECH.COM

TABLE OF CONTENTS

FIGURES AND TABLES

REVERSION HISTORY

Reversion	Date	Reason for change	By
V0.90	2/18/1999	First release version	JeffreyChang
V0.91	6/2/1999	Add D version silicon features modification	JeffreyChang
V0.92	8/23/1999	Add E version silicon features modification	MurphyChen
V0.93	9/9/1999	Revision according to Weipin's, Kevin's, and Ruth's comments	MurphyChen

FEATURES

- Single chip 16/12 ports 10/100M Ethernet switch controller
  - Highly integrated single chip shared memory switch engine
  - With option for 16 RMII (Reduced Media Independent Interface) ports or 12 MII (Media Independent Interface) ports
  - Non-blocking layer 2 switch, 148,810 packets/sec on each 100Mbps Ethernet port
- Media Access Control (MAC)
  - Dual 192-bytes FIFO's of receive and transmit for each port
  - CRC generator for outgoing packets from CPU port
  - IEEE 802.3X compliant flow control for full duplex ports
  - Backpressure for half duplex ports
- Two switching mechanisms
  - Supports 'store and forward' switching without forwarding CRC-bad packets
  - Supports 'cut through' switching subject to long packets of length over 290 bytes for 100Mbps ports or of length over 98 bytes for 10Mbps ports
- Packet buffering
  - Glueless 64-bit interface to SDRAM as a packet buffer pool with size from 2M bytes (SGRAM) to 512 M bytes
  - 1536 bytes for each packet buffer
- External 32 bits SSRAM interface for forwarding table and memory link table
  - Link list structure initialized by software
  - 2K up to 32K unicast/multicast addresses table entries with VLAN information
  - Supports static entries for upper-layer multicast protocols, e.g. IGMP
- Advanced address recognition
  - Layer 2 MAC address recognition engine to enable wire-speed forwarding rate
  - Self learning mechanism
  - Supports multiple MAC address per-port from 2K up to 32K unicast/multicast addresses
- Switch management support
  - Supports port mirroring (Sniffer feature)
  - Supports spanning tree algorithm
  - Supports CPU direct access to SDRAM and SSRAM
  - Supports five statistical counters in each port
- Supports I^{2}C EEPROM interface for customized configuration
- Support port-grouping VLAN
  - Configurable server ports belonging to multiple VLAN groups
- Support port-based trunking
  - Three types of trunk grouping: one trunk group with 2 or 4 ports, two trunk groups each with 2 ports
  - Load balance according to MAC address and port number
- CPU interface VIA 8/16 bits ISA-like interface

- Chip initialization, auto-aging and spanning tree algorithm support by firmware
- Auto-sensing 10/100M media speed, duplex mode, and flow-control capability by firmware
- 50MHz internal reference clock rate
- 50~100MHz SDRAM clock rate, typically 83MHz
- 50~100MHz SSRAM clock rate, typically 83MHz
- Single +3.3V supply, 0.3μm standard CMOS technology
- 476 ball BGA package

BLOCK DIAGRAM
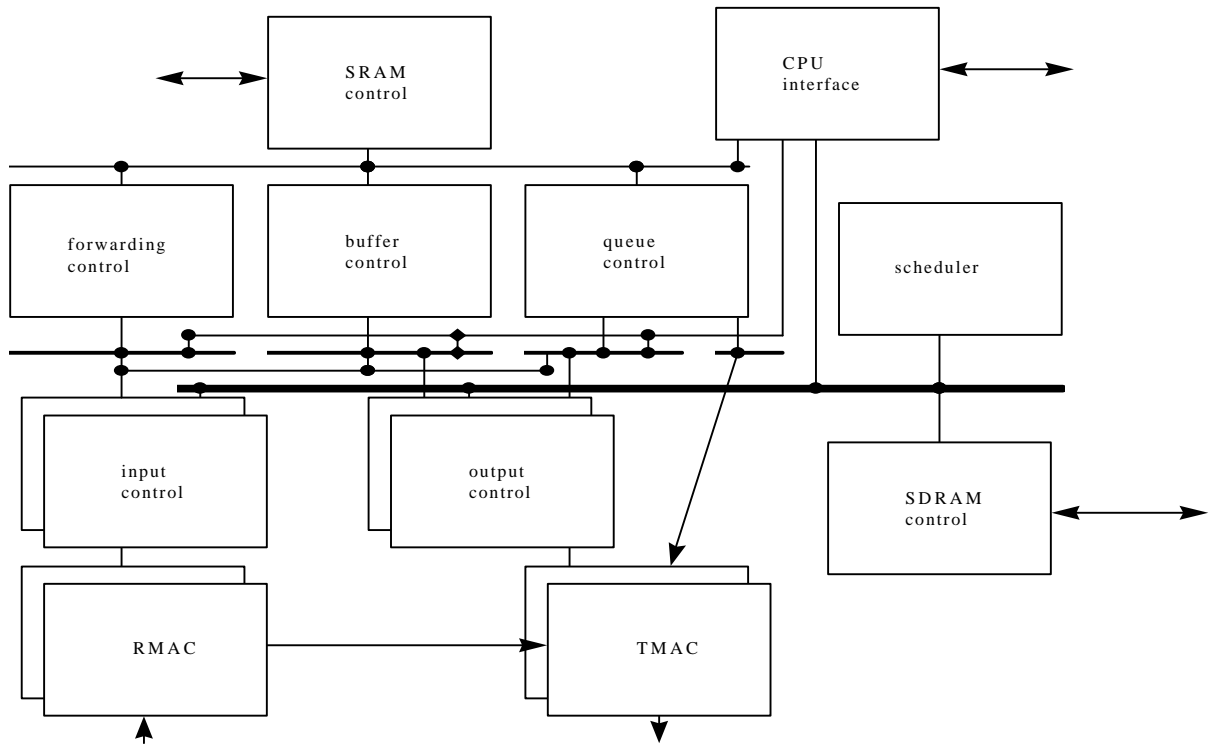
Figure 1: Block Diagram

BALL OUT DIAGRAM

RMII-mode Ball out Diagram

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
A	CSDV2	CRS.DV1	RXD1.1	TXEN1	TXD1.0	RXD0.0	MD1	MD3	MD5	MD7	MD9	MD11	MD45	MD47	RAS0	MA3	MA7	MA11	DCS2	DWE0	MD49	MD20	MD
B	RXD1.2	TXD1.2	TXD1.1	RXD0.1	CSDV0	TXD0.0	MD32	MD34	MD36	MD38	MD40	MD42	MD14	CAS1	MA0	MA4	MA8	BA0	DCS1	DWE1	MD18	MD52	MD
C	TXEN2	RXD0.2	TXD0.2	TXD0.1	TXEN0	RXD1.0	MD0	MD2	MD4	MD6	MD8	MD10	MD46	CAS0	MA1	MA5	MA9	BA1	DCS0	MD16	MD50	MD21	MD
D	TXD1.3	CSDV3	TXEN3	NC	NC	NC	MD33	GND	MD37	MD39	MD41	MD43	MD13	MD15	RAS1	MA2	MA6	MA10	DCS3	MD48	MD19	MD53	MD
E	RXD0.3	TXD0.3	RXD1.3	NC	VDDI	GNDI	GND	MD35	VDD	VDD	VDD	MD12	MD44	GND	GND	GND	VDD	VDD	VDD	MD17	MD51	GNDI	DC
F	RXD1.4	TXD0.4	RXD0.4	TXEN4	RCLK50	VDDI	GND	NC											NC	GND	GNDI	VDDI	MD
G	RXD0.5	CSDV4	TXD1.4	TXEN5	VDD	GND															GND	SCLK	SD
H	TXD1.5	RXD1.5	TXD0.5	NC	VDD	VDD															NC	SD23	SD
J	TXD0.6	RXD0.6	CSDV5	NC	VDD																	VDD	SD
K	CSDV6	TXD1.6	RXD1.6	TXEN6	NC																	VDD	SA
L	RXD1.7	TXD0.7	RXD0.7	NC	GND						GND	GND	GND	GND	GND	GND						VDD	SA
M	CSRV7	TXD1.7	TXEN7	NC	GND						GND	GND	GND	GND	GND	GND						SA10	SA
N	RXD1.8	TXD1.8	CSDV8	TXEN8	NC						GND	GND	GND	GND	GND	GND						SA15	SA
P	TXD0.8	RXD0.8	CSDV9	TXEN9	NC						GND	GND	GND	GND	GND	GND						GND	SD
R	TXD1.9	RXD1.9	TXD0.9	NC	VDD						GND	GND	GND	GND	GND	GND						GND	SD
T	RXD0.9	RXD0.10	TXD0.10	TXEN10	VDD						GND	GND	GND	GND	GND	GND						SD11	SD
U	RXD1.10	TXD1.10	CSDV10	NC	VDD																	VPP	SA
V	RXD0.11	TXD0.11	RXD1.11	TXEN11	VDD																	VPP	SO
W	TXD1.11	CSDV11	RXD0.12	TXEN12	NC	GND															GND	SCS3#	SC
Y	TXD0.12	RXD1.12	TXD1.12	NC	GND	GNDI															GNDI	HCLK	HD
AA	CSDV12	RXD0.13	TXD0.13	NC	GNDI	VDDI	VDD	VDD											GND	GND	GND	GNDI	HD
AB	RXD1.13	TXEN13	TXD1.13	NC	VDDI	VDD	VDD	NC	VPP	VPP	NC	NC	GND	GND	NC	NC	VPP	VPP	VPP	TEST12	VDDI	GND	HD
AC	CSDV13	CSDV14	TXD1.14	NC	EEC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	TEST7	TEST11	TEST16	TEST17	HD
AD	RXD1.14	TXD0.14	RXD1.15	TXEN15	EEIO	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	TEST3	TEST6	TEST10	TEST15	TEST20	TE
AE	RXD0.14	CSDV15	TXD0.15	MDC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	TEST2	TEST5	TEST9	TEST14	TEST19	TE
AF	TXEN14	TXD1.15	RXD0.15	MDIO	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	TEST1	TEST4	TEST8	TEST13	TEST18	TE

MII-mode Ballout Diagram

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
A	CRS1	RXD3_0	COL0	TXD1_0	RXD0_0	MD1	MD32	MD34	MD36	MD38	MD40	MD42	MD14	CAS1#	MA0	MA4	MA8	BA0	DCS1#	DWE1#	MD18	MD20	MD
B	RXD1_1	RXDV0	TXD3_0	RXD2_0	CRS0	TXD0_0	MD0	MD2	MD4	MD6	MD8	MD10	MD46	CAS0#	MA1	MA5	MA9	BA1	DCS0#	MD16	MD50	MD52	MD
C	TXEN1	TXD1_1	TXD0_0	TXD2_0	TXEN0	RXD1_0	MD33	MD35	MD37	MD39	MD41	MD43	MD13	MD15	RAS1#	MA2	MA6	MA10	DCS3#	MD48	MD19	MD21	MD
D	TXD3_1	RXD0_1	COL1	NC	NC	NC	MD3	VSS	MD7	MD9	MD11	MD12	MD44	RAS0#	MA3	MA7	MA11	DCS2#	DWE0#	MD17	MD51	MD53	MD
E	RXD2_1	RXDV1	RXD3_1	NC	VDD	GND	VSS	MD5	VCC	VCC	VCC	MD45	MD47	VSS	VSS	VSS	VCC	VCC	VCC	MD49	NC	GND	DC
F	RXD1_2	TXD2_1	RXD0_2	TXEN2	RCLK5_0	VDD	VSS	NC										NC	VSS	GND	VDD	MI	
G	RXD2_2	TXD0_2	TXD1_2	COL2	VCC	VSS															VSS	SCLK	SD
H	TXD3_2	CRS2	TXD2_2	RCLK2	VCC	VCC															NC	SD23	SD
J	TXD0_3	RXD3_2	RXDV2	TCLK2	VCC																	VCC	SD
K	RXD0_3	CRS3	TXD1_3	RXD1_3	TXEN3																	VCC	SA
L	RXD3_3	TXD2_3	RXD2_3	RCLK3	VSS						VSS	VSS	VSS	VSS	VSS	VSS						VCC	SA
M	RXDV3	TXD3_3	COL3	TCLK3	VSS						VSS	VSS	VSS	VSS	VSS	VSS						SA10	SA
N	TXD0_4	RXD1_4	TXD1_4	CRS4	TXEN4						VSS	VSS	VSS	VSS	VSS	VSS						SA15	SA
P	TXD3_4	RXD0_4	RXDV4	COL4	RCLK4						VSS	VSS	VSS	VSS	VSS	VSS						VSS	SD
R	RXD2_4	RXD3_4	TXD2_4	TCLK4	VCC						VSS	VSS	VSS	VSS	VSS	VSS						VSS	SD
T	RXD1_5	RXD0_5	TXD0_5	TXEN5	VCC						VSS	VSS	VSS	VSS	VSS	VSS						SD11	SD
U	RXD2_5	TXD1_5	CRS5	TCLK5	VCC																	VCC	SA
V	TXD3_5	TXD2_5	RXD3_5	COL5	VCC																	VCC	SO
W	TXD0_6	RXDV5	RXD0_6	TXEN6	RCLK5_6	VSS															VSS	SCS3#	SC
Y	CRS6	RXD1_6	TXD1_6	TCLK6	VSS	GND															GND	HCLK	HD
AA	RXD3_6	RXD2_6	TXD2_6	RCLK6	GND	VDD	VCC	VCC											VSS	VSS	VSS	GND	HD
AB	RXDV6	COL6	TXD3_6	TCLK7	VDD	VCC	VCC	NC	VCC	VCC	TCLK9	RXD3_10	VSS	VSS	CRS10	RCLK11	VCC	VCC	VCC	TEST12	VDD	VSS	HD
AC	RXD1_7	CRS7	TXD1_7	RCLK7	EEC	RCLK8	TCLK8	RXD3_9	RXD2_9	RCLK9	TXD2_10	RXD2_10	RCLK10	TCLK10	RXD3_11	RXDV11	TCLK11	CRS11	TEST7	TEST11	TEST16	TEST17	HD
AD	RXD0_7	TXD0_7	RXD3_7	COL7	EEIO	RXD0_8	TXD0_8	TXD3_8	RXD1_9	TXEN9	TXD3_9	RXD1_10	TXEN10	TXD2_10	RXD2_11	TXEN11	TXD2_11	TEST3	TEST6	TEST10	TEST15	TEST20	TE
AE	TXEN7	RXDV7	TXD2_7	MDC	RXD2_8	RXDV8	TXD1_8	COL8	RXD0_9	TXD0_9	COL9	RXD0_10	TXD0_10	TXD3_10	RXD1_11	TXD0_11	TXD3_11	TEST2	TEST5	TEST9	TEST14	TEST19	TE
AF	TXD3_7	RXD2_7	MDIO	RXD3_8	RXD1_8	TXEN8	TXD2_8	CRS8	RXDV9	TXD1_9	CRS9	RXDV10	TXD1_10	COL10	RXD0_11	TXD1_11	COL11	TEST1	TEST4	TEST8	TEST13	TEST18	TE

LOGIC SYMBOL
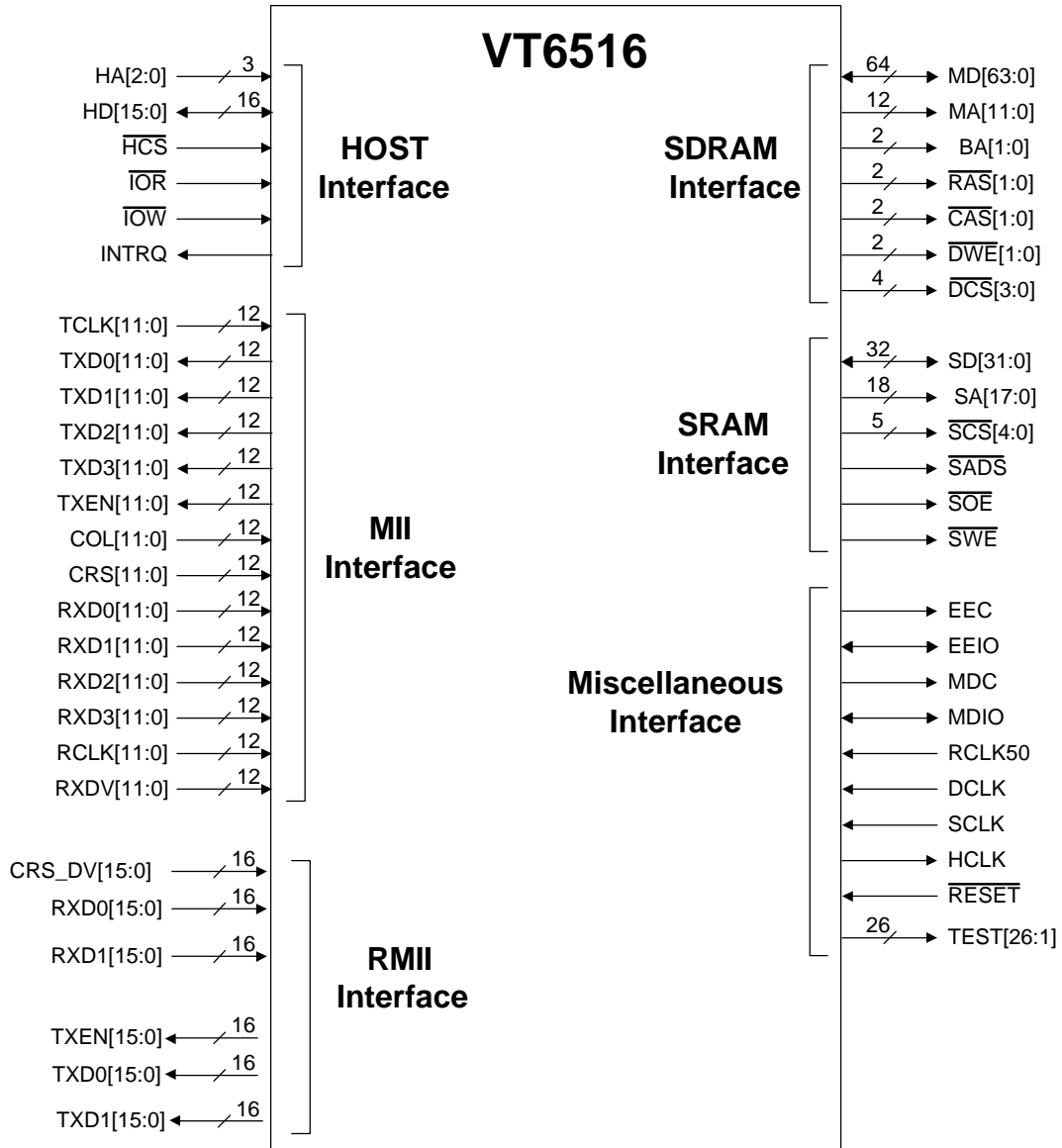
PIN DESCRIPTIONS

No.	Name	Type	Description
			SDRAM Interface
See Ball Table	MD[63:0]	I/O	**SDRAM Data:** 64-bit SDRAM data bus. These signals connect directly to the data input/output pins of the SDRAM devices.
See Ball Table	MA[11:0]	O	**SDRAM Address Bus:** 12-bit SDRAM data bus. These signals connect directly to the address input of the SDRAM devices.
See Ball Table	BA[1:0]	O	**Bank Identifier for Bank 0 and 1:**
See Ball Table	$\overline{RAS}$[1:0]	O	**Row Address Strobes for Bank 0 and 1:** DRAM row address strobes. $\overline{RAS}$[0] is used for Bank 0. $\overline{RAS}$[1] is used for Bank 1.
See Ball Table	$\overline{CAS}$[1:0]	O	**Column Address Strobes for Bank 0 and 1:** DRAM column address strobes. $\overline{CAS}$[0] is used for Bank 0. $\overline{CAS}$[1] is used for Bank 1.
See Ball Table	$\overline{DWE}$[1:0]	O	**DRAM Write Enable for Bank 0 and 1:**
See Ball Table	$\overline{DCS}$[3:0]	O	**DRAM Chip Select:** VT-3061A supports at most 4 SDRAM DIMM modules.
			SRAM Interface
See Ball Table	SD[31:0]	I/O	**SRAM Data:** 32-bit SRAM data bus. These signals connect directly to the data input/output pins of the SRAM devices.
See Ball Table	SA[17:0]	O	**SRAM Address Bus:** 18-bit SDRAM data bus. These signals connect directly to the address input of the SDRAM devices.
See Ball Table	$\overline{SCS}$[4:0]	O	**SRAM Chip Select:** SRAM Type — Chip Select Pins — Address Pins 32KBx32 — SCS[0] & SA[15] — SA[14:0] 64KBx32 — SCS[0] & SA[16] — SA[15:0] 128KBx32 — SCS[0] & SA[17] — SA[16:0] 256KBx32 — SCS[0] — SA[17:0]
See Ball Table	$\overline{SADS}$[1:0]	O	**Synchronous Processor Address Status**
See Ball Table	$\overline{SOE}$[1:0]	O	**Output Enable**
See Ball Table	$\overline{SWE}$[1:0]	O	**SRAM Write Enable:**
			Miscellaneous Interface

See Ball Table	EEC	O	**Serial EEPROM Interface Clock Output:**
			EEPROM Device Addressing in the demo board:
			PAGE 0 (EEPROM): Device Address = 1010 000 XXXXXXXX
			PAGE 1 (EEPROM): Device Address = 1010 001 XXXXXXXX
			PAGE 2 (EEPROM): Device Address = 1010 010 XXXXXXXX
			PAGE 3 (EEPROM): Device Address = 1010 011 XXXXXXXX
			PAGE 4 (SDRAM BANK-0): Device Address = 1010 100 XXXXXXXX
			PAGE 5 (SDRAM BANK-1): Device Address = 1010 101 XXXXXXXX
See Ball Table	EEIO	I/O	**Serial EEPROM Interface Data I/O**
See Ball Table	MDC	O	**Management Interface (MI) Clock Output**
See Ball Table	MDIO	I/O	**Management Interface (MI) Data I/O**
See Ball Table	RCLK50	I	**Main Reference Clock:**
See Ball Table	DCLK	I	**SDRAM Reference Clock:**
See Ball Table	SCLK	I	**SRAM Reference Clock**
See Ball Table	HCLK	O	**HOST Reference Clock**
			HCLK is determined by the strapping pins in SYSLED[3:1], i.e. the jump selection of J1[5-6, 3-4, 1-2]:
			J1[OFF,OFF,OFF] => 8MHz
			J1[OFF,OFF, ON] => 16MHz
			J1[OFF, ON, OFF] => 25MHz
			J1[OFF, ON, ON] => 4MHz
			J1[ON,OFF,OFF] => 33MHz
See Ball Table	$\overline{\text{RESET}}$	I	**SYSTEM RESET**
See Ball Table	SYSLED[26:0]	O	**SYSTEM Output Pins for LED:**
			SYSLED[8:0] are connected to pull-up IO PADs for strapping.
			SYSLED[25:9] are connected to IO PADs without pull up/down.
			All SYSLED[25:0] are

HOST Interface

See Ball Table	HA[2:0]	I	**HOST IDE-Interface Address Bus:**
			3'b000: command the switch that the whole 16-bit data in the HOST data bus HD[15:0] is valid for packet-data read/write.
			3'b001: command the switch that only the 8-bit data in the HOST data bus HD[15:0] is valid for internal registers read/write.
			3'b010: command the switch to write the low byte in the HOST data bus HD[15:0] into the low byte of the 16-bit switch address register for internal registers reference.
			3'b011: command the switch to write the low byte in the HOST data bus HD[15:0] into the high byte of the 16-bit switch address register for internal registers reference.
			3'b1xx: bus-idle command. Keep this address bus to be 3'b111 as the HOST has no access to VT-3061A.
See Ball Table	HD[15:0]	I/O	**HOST IDE-Interface Data Bus:**
			The whole 16-bit data bus is valid for packet data read/write. However, only the 8-bit data bus is valid for internal registers read/write.
See Ball Table	$\overline{\text{HCS}}$	I	**HOST Chip Select:**
			Active LOW. $\overline{\text{HCS}}$ must be asserted during the access of HOST IDE interface.
See Ball Table	$\overline{\text{IOR}}$	I	**IO READ:**
			High-to-Low Edge Trigger. $\overline{\text{IOR}}$ must be asserted from high to low to begin the read cycle of HOST IDE interface.
See Ball Table	$\overline{\text{IOW}}$	I	**IO READ:**
			High-to-Low Edge Trigger. $\overline{\text{IOW}}$ must be asserted from high to low to begin the write cycle of HOST IDE interface.
See Ball Table	INTRQ	O	**Interrupt Request:**
			Connected to the HOST external interrupt pin. It is asserted as the following four interrupt events happen:
			(1) MII Management Registers read/write command done
			(2) EEPROM read/write command done
			(3) Receiving a packet destined to HOST
			(4) Finishing transmission of a packet issued by HOST
			The interrupt cause is recorded in register IRQSTS[3:0] in address 2000H. To clear the individual interrupt, The corresponding register has to be written:
			(1) register CLR_PHY_INT in 1806H for PHY interrupt.
			(2) register CLR_EE_INT in 1C04H for EEPROM interrupt.
			(3) register CLR_RCV_INT in 6403H for packet-receiving interrupt.
			• register CLR_SENT_INT in 6411H for packet-sent interrupt.

MII Interface			
See Ball Table	TCLK[11:0]	I	**Transmit Clock for Port 0-11:**
			TCLK is driven by the PHY device. TCLK is a continuous clock that provides the timing reference for the transfer of the TXEN and TXD signals to the PHY. A PHY operating at 100Mbps must provide a TCLK frequency of 25MHz and a PHY operating at 10Mbps must provide a TCLK frequency of 2.5MHz.

See Ball Table	TXD<3:0>[11:0]	O	**Transmit Data for Port 0-11:** TXD is a bundle of 4 data signals (TXD<3:0>) that shall transition to the TCLK. For each TCLK period in which TXEN is asserted, TXD<3:0> are accepted for transmission by the PHY. TXD<0> is the least significant bit. While TXEN is de-asserted, TXD<3:0> shall have no effect upon the PHY, and the value of TXD<3:0> is unspecified.
See Ball Table	TXEN[11:0]	O	**Transmit Enable for Port 0-11:** TXEN shall transition synchronous to the TCLK. TXEN indicates the nibbles presenting on the MII for transmission. It shall be asserted synchronously with the first nibble of the preamble and shall remain asserted while all nibbles to be transmitted are presented to the MII.
See Ball Table	COL[11:0]	I	**Collision Detected for Port 0-11:** COL shall be asserted by the PHY asynchronously upon detection of a collision on the medium, and shall remain asserted while the collision condition persists.
See Ball Table	CRS[11:0]	I	**Carrier Sense for Port 0-11:** CRS shall be asserted by the PHY asynchronously upon detection of a non-idle medium or while TX_EN is asserted. CRS shall be de-asserted by the PHY asynchronously upon detection of idle conditions on both transmit and receive media. The PHY shall ensure that CRS remains asserted throughout the duration of a collision condition.
See Ball Table	RXD<3:0>[11:0]	I	**Receive Data for Port 0-11:** RXD is a bundle of 4 data signals (RXD<3:0>) that shall transition to the RCLK. For each RCLK period in which RXDV is asserted, RXD<3:0> from the PHY are accepted by the switch's MAC. RXD<0> is the least significant bit. While RXDV is de-asserted, RXD<3:0> shall have no effect upon the switch's MAC, and the value of RXD<3:0> is unspecified.
See Ball Table	RCLK[11:0]	I	**Receive Clock for Port 0-11:** RCLK is sourced from the PHY. RCLK is a continuous clock that provides the timing reference for the transfer of the RXDV and RXD signals from the PHY. A PHY operating at 100Mbps must provide a RCLK frequency of 25MHz and a PHY operating at 10Mbps must provide a RCLK frequency of 2.5MHz.
See Ball Table	RXDV[11:0]	I	**Receive Data Valid for Port 0-11:** RXDV is driven by the PHY to indicate the nibbles presenting on the MII for receiving. RXDV shall transition synchronous to the RCLK. It shall be asserted synchronously with the first nibble of the preamble and shall remain asserted while all nibbles to be received are presented to the MII.

Note: Some flat MII input pin when the VT6516 under the RMII application, please use 22 ohm resister pull down, refer to Table XXXX

RMII interface			
See Ball Table	CRS_DV[15:0]	I	Carries sense and data valid from port 15 to port 0 :
See Ball Table	RXD0[15:0]	I	Receive data zero from port 15 to port 0 :
See Ball Table	RXD1[15:0]	I	Receive data one from port 15 to port 0 :
See Ball Table	TXEN[15:0]	O	Transmit enable from port 15 to port 0 :
See Ball Table	TXD0[15:0]	O	Transmit data zero from port 15 to port 0 :
See Ball Table	TXD1[15:0]	O	Transmit data one from port 15 to port 0 :

Power Supply & Ground			
See Ball Table	VDD, VDDA	P	Positive 3.3V Supply: Supply power to Internal digital logic, Digital I/O pads, and TD, TX pads. Double bonding may be required.
See Ball Table	VSS, VSSA	G	Negative Supply: digital ground. Multiple bonding pads are required to separate core and I/O pads ground.

JUMPER STRAPPING

Jumper	Pin	Description
HOST Clock		
J1 [5-6], [3-4], [1-2]	SYSLED[3:1]	HOST Clock (HCLK) Rate Selection: J1[OFF,OFF,OFF] (SYSLED[3:1]==3'b111) => 8MHz J1[OFF,OFF, ON] (SYSLED[3:1]==3'b110) => 16MHz J1[OFF, ON, OFF] (SYSLED[3:1]==3'b101) => 25MHz J1[OFF, ON, ON] (SYSLED[3:1]==3'b100) => 4MHz J1[ON,OFF,OFF] (SYSLED[3:1]==3'b011) => 33MHz
PHY Mode		
J1 [7-8]	SYSLED[4]	PHY Device Selection: J1[OFF] (SYSLED[4]==1'b1) => RMII PHY J1[ON] (SYSLED[3:1]==1'b0) => MII PHY
SRAM Type		
J1 [11-12,9-10]	SYSLED[6:5]	SRAM Device Type Selection: J1[OFF,OFF] (SYSLED[6:5]==2'b11) => 64K x 32 SRAM J1[OFF,ON] (SYSLED[6:5]==2'b10) => 128K x 32 SRAM J1[ON,OFF] (SYSLED[6:5]==2'b01) => 32K x 32 SRAM

SECTION I FUNCTIONAL DESCRIPTIONS

1. GENERAL DESCRIPTION

The VT6516 is a switch engine chip implementation of a 16 ports 10/100M Ethernet switch system for IEEE 802.3 and IEEE 802.3u network. Each of individual port can be either auto-sensing or manually selected to run at 10Mbps or 100Mbps speed rate and under full or half duplex mode.

There are sixteen independent MACs within the VT6516 chip. The MAC controller controls the receiving, transmitting, and deferring of each individual port, and the MAC controller also provides framing, FCS checking, error handling, status indication and flow control function.

The VT6516 10/100M N-way switch port IC is wire-speed performance and low-cost packet switch; it can forward up to 148,810 packets/sec on each Ethernet port. The VT6516 support 12 ports MII or 16 ports RMII (reduce MII) interface for network interface,

The VT6516 used the simple 8/16 bits ISA-like interface to support initiation, expansion and management. The system CPU can access various registers inside VT6516 through a simple ISA-like CPU interface. The CPU can configure the switch by writing into the appropriate registers, or retrieve the status of the switch by reading the corresponding registers. The CPU can also access the register of external transceiver (PHY) device through the CPU interface.

The VT6516 supports new features including port based VLAN , 802.3X flow control, and the VT6516 also support the sniffer function to monitor network traffic in special ports.

2. THE VIA ETHER SWITCH ARCHITECTURE

The VT6516 switch engine uses the shared memory architecture. In order to improve the packet latency, VT6516 provides two methods for packet switching, one is cut-through, another is store-and-forwarding. A typical packet flow for Ethernet switch is described as follows in 4.5.

2.1 Switch initialization procedures

1. Test all of the on board components except the switch chip or access VIA the switch chip, including UART, LED, etc.
2. Switch SDRAM test --- switch chip SDRAM control hardware initialization, configuration, SDRAM size determination (VIA embedded EEPROM in SDRAM module) and read write test.
3. Switch SRAM test --- switch chip SDRAM control hardware initialization and read write test. Note that the SRAM size determination is VIA strapping.
4. Switch IO registers read write test.
5. Ethernet PHY registers read write test ---- the CPU read/write to PHY devices will go through PHY control in switch chip. Although they are outside components, but we test them as part of the switch chip.
6. Determine link table size; reset free buffer list pointers of bank 0 and 1; initialize free memory block counter. Note that permanent buffer management is controlled by allocating bit mask. They will be cleared automatically in the hardware reset or software reset.

2.2 Packet Switching Flow

1. After the switch microprocessor activates a port during initialization, the input control of that port pre-allocates one packet buffer from buffer pool. In the beginning, the buffer allocated will be from private buffer pool, but subsequent buffers may come from either private or public buffer pools.

2. When receive MAC (RMAC) receives a packet data from the network interface – either through MII or reduced MII (RMII) – it packs the data into 16-bit word then passes it to input control. If RMAC detects any error, it also notifies input control to stop forwarding process.

3. Input control extracts the destination MAC address from incoming data, passes it along to forwarding table control for forwarding decision. In the mean while, it packs 16-bit words into 64-bit quad-words, and saves it to an input FIFO before storing the packet data to SDRAM.

4. If the switch is configured to "store and forward" mode, input control queues the packet to the output queue of the destination port after input control is informed by RMAC that this is a good packet and it stores all packet data to SDRAM. If the switch is configured to "cut-through" mode, the input control queues the packet to the output queue of the destination port after enough amount of packet is stored in SDRAM to prevent output FIFO under-run.

5. After the whole packet is received and FCS is correct, input control pass the source MAC address of the packet to forwarding table control for address learning.

6. Output control of the outbound port de-queue the packet from output queue, and fetch packet data from SDRAM and save it into output FIFO. Then it notifies the transmit MAC (TMAC) of the new packet to transmit.

7. TMAC grabs 16-bit at a time from output control, adds preamble and SFD to the beginning of the packet, then send them out. Proper deferring is done if necessary to conform to 802.3 standard.

8. After the packet is successfully transmitted, TMAC notifies output control of the successful transmission. Output control then returns the packet to buffer pool.

3. INTERFACE DESCRIPTIONS

BUFFER MEMORY (SDRAM) INTERFACE AND TABLE (SSRAM) INTERFACE

VT6516 provides a 64-bit SDRAM/SGRAM interface for packet buffering and a 32-bit synchronous SRAM (SSRAM) interface for maintaining address table and various link lists. VT6516 uses SDRAM as packet buffers. Each packet buffer is a 1536-byte contiguous memory block in SDRAM, and corresponds to a 12-byte link node data structure in SSRAM. Except the first 128 link nodes, each link node can be part of an output queue, a free buffer link list, or held in input or output control. The first 128 link nodes are divided into 16 groups, each pre-assigned to a specific input control, and bit-mapped inside buffer control for faster allocate/free operation and reduce SSRAM usage.

    Initially, each input port control would request one packet buffer from its private buffer pool. Each time when a packet buffer is consumed by an incoming packet, the input port control will request another packet buffer to prepare for next packet. The common shared packet memory will be allocated only when there's no free permanent packet memory for that port. See Figure 3-4.

SRAM	DRAM

128 entries

12 bytes/entry

Permanent Buffer Table

12 bytes/entry

Free List Link Table

Address Table Entriers

1.5 K/Packet

…

1.5 K/Packet

128 blocks
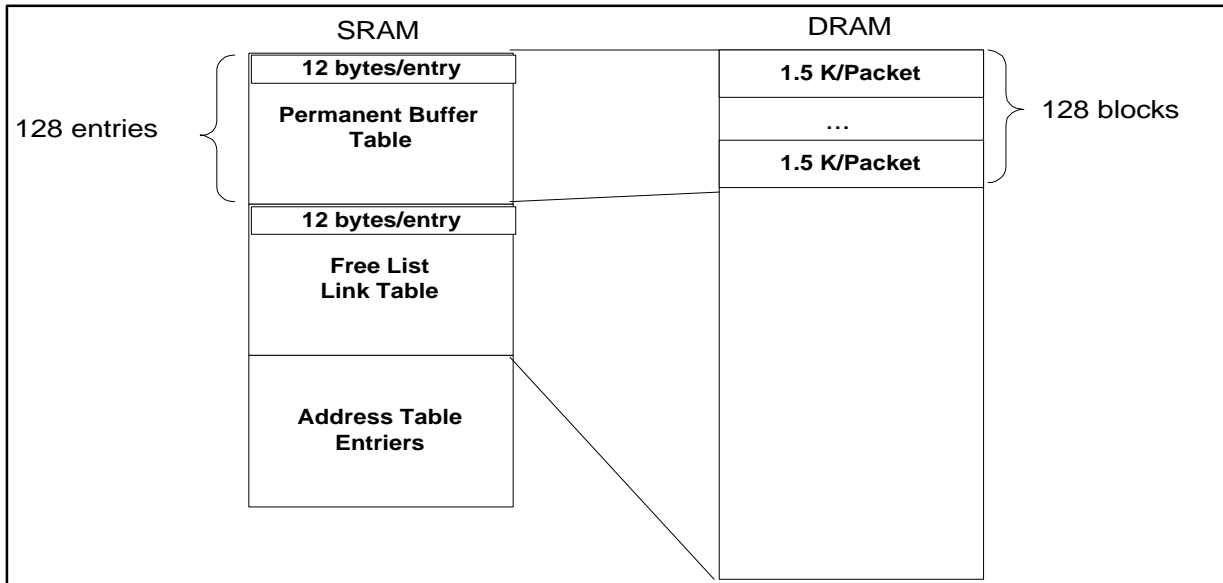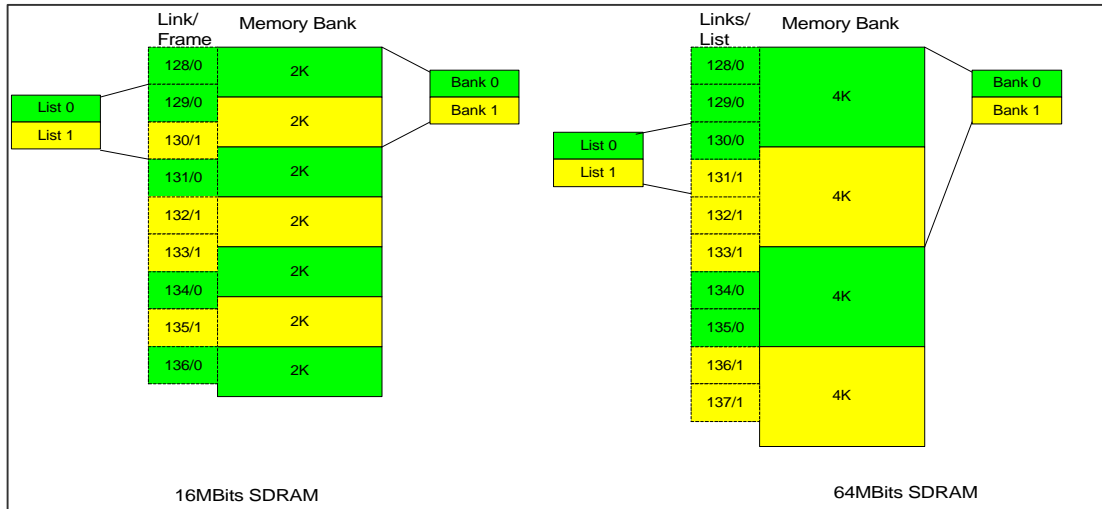
Figure 3-3

Following as the listing and figure 3-6 is the algorithm of initialization procedures for 2 bank free list of SDRAM.

For 16 Mbit SDRAM as following,
 -- Bank0 free link list:
   128, 129, 131, 134, 136, 137, 139, 142, 144, 145, 147, 150, 152, 153, 155, 158, 160,...
 -- Bank1 free link list:
   130, 132, 133, 135, 138, 140, 141, 143, 146, 148, 149, 151, 154, 156, 157, 159, 162, 164, ...

For 64 Mbit SDRAM as following,
 -- Bank0 free link list:
   128 , 129 , 130 , 134 , 135 , 139 , 140 , 141 , 144 , 145 , 146 , 150 , 151 , 155 , 156 , 157 , 160 , ...
 --Bank1 free link list:
   131 , 132 , 133 , 136 , 137 , 138 , 142 , 143 , 147 , 148 , 149 , 152 , 153 , 154 , 158 , 159 , 163 , …

Figure 3-6: Algorithm of Initialization of Free Link Lists.

```
#define SRAM_ADDR_REG0 0x2001
#define SRAM_ADDR_REG1 0x2002
#define SRAM_ADDR_REG2 0x2003
#define SRAM_DATA_REG0 0x2004
#define SRAM_DATA_REG1 0x2005
#define SRAM_DATA_REG2 0x2006
#define SRAM_DATA_REG3 0x2007
#define SRAM_CMD_REG   0x2008
#define SRAM_STATUS_REG 0x2009
#define SRAM_ACCESS_IDLE 0x01
#define NULL_PTR           0x7FFFF

void writeLinkEntry(int entryID, int nextID)
{
  reg_byte_write (SRAM_ADDR_REG0, entryID*3 & 0x0FF);
  reg_byte_cont_write (((entryID*3) >> 8) & 0x0FF);
  reg_byte_cont_write (((entryID*3) >> 16) & 0x0FF);
```

```
 reg_byte_cont_write (nextID & 0x0FF); // data bits [7:0]
 reg_byte_cont_write ((nextID >> 8) & 0x0FF); // data bits [15:8]
 reg_byte_cont_write ((nextID >> 16) & 0x0FF); // data bits [23:16]
 reg_byte_cont_write (0); // data bits [31:24]
 reg_byte_cont_write (0x02); // SRAM-write command
 while (reg_byte_read(SRAM_STATUS_REG) != SRAM_ACCESS_IDLE) {}
}

void initFreeList16Mb(int maxLinkEntryNo)
{ // note: for 16Mb SDRAM,
 //      Bank0 free list head pointer = 128
 //      Bank1 free list head pointer = 130
  int k; // k: current free entry id
  int b0, b1; // b0, b1: bank0/1 free list head entry id
  for(b0=b1= NULL_PTR, k= maxLinkEntryNo; k <=128; k--)
    if (((k * 3) % 8) < 4)
       { writeLinkEntry(b0,k); b0=k;}
    else { writeLinkEntry(b1,k); b1=k;}
}

void initFreeList64Mb(int maxLinkEntryNo)
{ // note: for 64Mb SDRAM,
 //      Bank0 free list head pointer = 128
 //      Bank1 free list head pointer = 131
  int k; // k: current free entry id
  int b0, b1; // b0, b1: bank0/1 free list head entry id
  for(b0=b1= NULL_PTR, k= maxLinkEntryNo; k <=128; k--)
    if (((k * 3) % 16) < 8)
       { writeLinkEntry(b0,k); b0=k;}
    else { writeLinkEntry(b1,k); b1=k;}
}
```

3.1.1 SDRAM interface

All frames received by the VT6516 will be stored into a common frame buffer memory, SDRAM.The SDRAM contains the packet buffers, each buffer is a 1536 (1.5K) bytes memory block. Each block is associated to an entry in link table in SRAM. The link entry includes a field (19 bits to support 512 MB) to point to next link entry. The figure 3-2 is buffer link list structure.

In order to provide the cost effective DRAM buffers, user can connect the 32 bits data SGRAM with VT6516 switch, there are two external buffer device using two double bank 128Kbits by 32 required. The following figure shows the minimum configuration of buffer memory and link/address memory. Note that the SGRAM physical memory hole is to accommodate the forwarding table into the SRAM link list hole.

Physical Buffer Memory
Allocation (eg, 2MB)

Physical SRAM Memory
Allocation (eg, 128K)
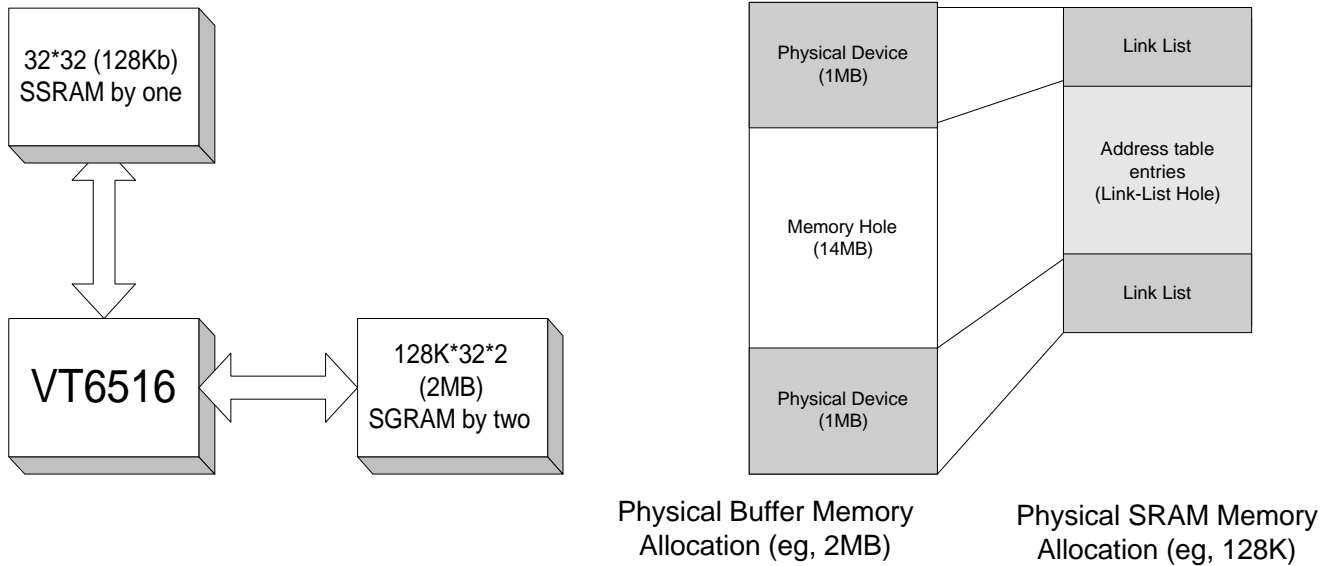
The detail initial step of VT6516 as following,
1. Forwarding table base = 683 * 3
2. SDRAM type equal to 16M bit
3. END0-3 = 2 (16MB)
4. Free list of SRAM have to be constructed by release public node in the sequence of buffers with blocks
   number
   10922, 10911, …10240, 681, 680, 128
   Note: The buffers numbered 682 to 10239 are located in the buffer memory hole, those buffers will be not
   put into the free list.
5. Free Memory count equal to 1364

Otherwise like this minimum configuration, the entries support for difference SRAM size using normal
Address table followed by free-list, the following table show the address entries support

Buffers	32*32 SSRAM	64*32	64*64
2MB	8K entries	8K~16K	8K~64K
4MB	8K entries	8K~16K	8K~64K

3.1.2 SRAM interface

The feature 3-1 is SSRAM structure map, the SSRAM contains the forwarding address entries, SDRAM buffers link list and permanent buffers table.
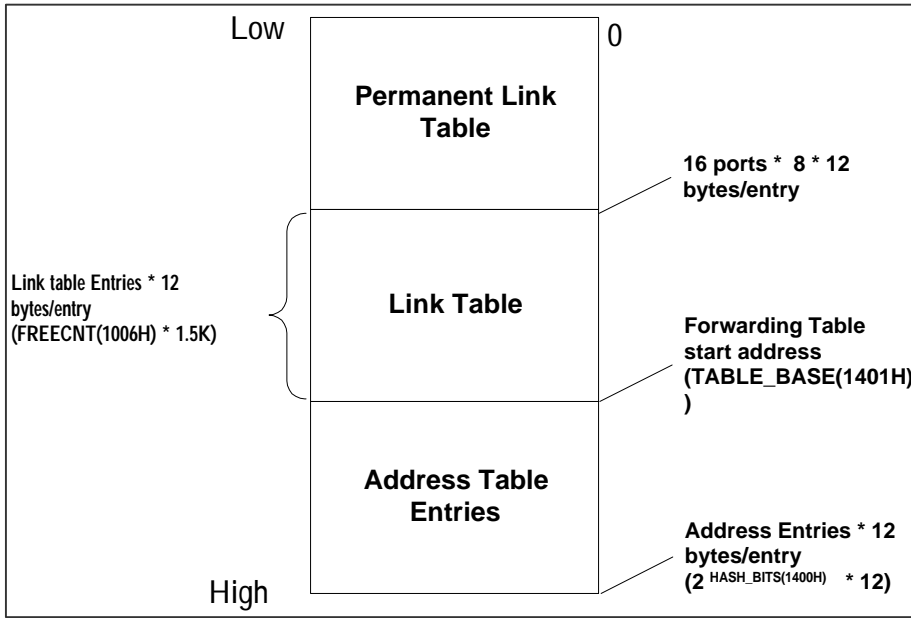

Figure 3-1 SRAM

95　　65	64　59	58	57　55	54　52	51	50　47	46　　36	35　19	18　0
reserved	VLAN ID	VLAN tage flag	frame tag type	priority	frame type	source port ID	packet byte count	port mask	pointer to next entry
								16 ports +1 cpu	

Figure 3-2 Free buffer link structure

Table 1-0 Free buffer link structure

bit 18-0	Next entry pointer
bit 35-19	port mask (bit[16]: CPU port + bit[15:0]: Ethernet ports 0~15)
bit 46-36	Packet byte count
bit 50-47	Source port ID
bit 51	~~Frame type~~ Reserved (zero)
Bit 54-52	Priority
Bit 57-55	~~Frame tag type~~ Reserved (zero)
bit 58	~~VLAN tag flag~~ Reserved (zero)
bit 64-59	VLAN ID
bit 95-65	reserved for future

95　　　　　64	63　58	57	56	55　54	53　37	36　　0
reserved	VLAN ID	VLAN tage flag	static/dynamic flag	age count	port mask	bit-47:11 MAC address
					16 ports +1 cpu	fixed, even for key length = 11~15

Figure 3-5 The Address table entries structure +

The address table structure as figure 3-5, The address table entries contains the MAC address information from bit 11 to bit 47, others bits 0~10 or bits 0~14 (2K~32K) as the address entries hashing index, the total address entries of device assigned by the HASH_BITS(1400H).

Initialization procedures include to set forwarding address table control in normal mode and to invalidate all forwarding table entries by setting the age-count field as 0. Configure port mask register (USER_PM) for broadcast MAC address. Configure port mask and MAC address pairs that allow any static MAC to port mask mapping.

Forwarding table entry has 96 bits, defined as follows:

Table 1-1 Address table structure

bit 36-0	High bits (bit 47-11) of MAC address
bit 53-37	port mask (bit[16]: CPU port + bit[15:0]: Ethernet ports 0~15)
bit 55-54	age count
bit 56	Static flag (0: dynamic entry, 1: static entry that can not be updated)
bit 57	~~VLAN tag flag~~ Reserved (zero)
bit 63-58	VLAN ID
bit 95-64	Reserved for future

And following is the algorithm for the initial the address entries;

```
#define SRAM_ADDR_REG0 0x2001
#define SRAM_ADDR_REG1 0x2002
#define SRAM_ADDR_REG2 0x2003
#define SRAM_DATA_REG0 0x2004
#define SRAM_DATA_REG1 0x2005
#define SRAM_DATA_REG2 0x2006
#define SRAM_DATA_REG3 0x2007
#define SRAM_CMD_REG   0x2008
#define SRAM_STATUS_REG 0x2009
#define SRAM_ACCESS_IDLE 0x01

void invalidateForwardEntry(int entryID)
{ // the entryID is starting from maxLinkEntryID with width of 96 bits
  reg_byte_write (SRAM_ADDR_REG0, (entryID*3+1) & 0x0FF);
  reg_byte_cont_write (((entryID*3+1) >> 8) & 0x0FF);
  reg_byte_cont_write (((entryID*3+1) >> 16) & 0x0FF);
  reg_byte_cont_write (nextID & 0x0FF); entry bits [32]
  reg_byte_cont_write ((nextID >> 8) & 0x0FF);
  reg_byte_cont_write ((nextID >> 16) & 0x0FF);
  reg_byte_cont_write ((nextID >> 16) & 0x0FF);
  reg_byte_cont_write (0x02); // SRAM-write command
  while (reg_byte_read(SRAM_STATUS_REG) != SRAM_ACCESS_IDLE) {}
}
```

3.1.3 CPU interface

The VT6516 support one ISA-like CPU interface, this CPU interface can cooperate with one simple microprocessor like 8031 or 8051. The CPU will access the switch control and status register to perform initialization and configurations. By the CPU interface, the frames of CPU port can be read/written from/into the buffer. The CPU interface can also be used to access the internal registers. The CPU interface also used to access the external PHY devices through the PHY control module.

The CPU firmware will perform following tasks,
- Read the configuration from switch register or from the EEPROM contains
- Initialize the switch followed by the configuration, those task including
    * DRAM initialization
    * SRAM initialization and link list construction
    * Program for each network ports for users manual setting or read the auto-negotiation result
- start switch to receive frames and forward frames
- decrease the learning address aging count
- polling the network port change event and change the switch MAC negotiation mode.

- Receiving the STP defined BPDU packets
- Blocking or re-start port due to STP
- Access the network management counter of each port

For a management switch the CPU also perform the management function like receiving and transmitting the SNMP frame.

3.1.4 Network interface

The VT6516 directly connect to 16 port RMII PHY or 12 port MII PHY device which compliant with IEEE standard (Please see IEEE 802.3u Fast Ethernet standard) . Each Fast Ethernet port has following characteristics:

- Capable of supporting both 10MBps and 100MBps data rates in half and full duplex modes.
- Provide a simple management interface (SMI) for port status
- Perform all functions of the IEEE 802.3 protocol such as frame formatting, frame stripping, collision handling, deferred, etc.
- Adjustable preamble ,SFD and inter frame gap (IFG).
- IEEE 802.3X flow control supported
- IEEE 802.1D spanning tree protocol support, and all port state of listen and block configurable

3.1.4.1 RMII interface

The VT6516 communicates with the external 10/100M Ethernet transceiver through the reduced MII (RMII) interface. The signals of RMII interface are described in Table-3-1

Table 3-1 RMII interface signals

Name	Type	Description
CRSDV	I	Carrier sense and Data valid
RXD[0-1]	I	Receive data bit 0 to 1 , data rate with 50MHz
TXEN	O	Transmit Enable
TXD[0-1]	O	Transmit Data bit 0 to 1

Figure 3-1 RMII timing diagram

(omitted)

3.1.4.2 MII interface

The VT6516 communicates with the external 10/100M Ethernet transceiver through standard MII interface, in this mode the VT6516 became 12 ports MII port due to the MII signal multiplexed with RMII signal. But the ports number of internal remained as 16 ports. The signals of MII interface are described in Table-3-2:

Table 3-2 MII interface signals

Name	Type	Description
TCLK	I	**Transmit Clock**
TXD[3:0]	O	**Transmit Data for**.
TXEN	O	**Transmit**
COL	I	**Collision Detected**
CRS	I	**Carrier Sense**
RXD[3:0]	I	**Receive Data**
RCLK	I	**Receive Clock**
RXDV	I	**Receive Data**

Figure 3-2 MII timing diagram

(omitted)

3.1.4.3 Flow control

Under full-duplex mode operation, if the buffer utilization of whole switch has exceeded the upper threshold and the permanent buffer has been used up, a pause frame with a pause time interval will be send to the sending port to stop it from sending new frame. If register- FMFCT not enable at this switch, the public buffer will used until no more buffers. Then further incoming frames will be dropped. The unit in pause time field of the flow control frame is slot time (512 bits).

The max possible waiting time should be the max packet memory size divided by lowest port speed, for example if 512MB is the max packet buffer size and 10Mb is the lowest speed, the 512M * 8 bits * 100ns = 409.6 seconds (8M slot time) is the max possible waiting time. The congestion factor is the max possible waiting time at current link load. The pause timer value is half of the max possible waiting time. If it is greater than the feasible max pause time, use all 1's in pause time value.

If the utilization of the public buffer of the switch drops below the lower threshold, a pause-frame with minimum frame interval of 0 will be sent to the linking ports the enable new frame transmission.

Under half duplex operation, if the buffer utilization of whole switch has exceeded the upper threshold and the permanent buffer has been used up, the port will perform back-pressure based flow control by sending a jam pattern on each incoming frame. If backpressure flow control of the port is not enable, the frame will be dropped. The flow control pause time is calculated by maintained the configuration of port speed of each port and the buffer size. With input of the free memory block count and congestion factor, it determines flow control on or off on an output port. If flow control is on, any new queue request from a input port to this output port will trigger a flow control frame sent to that request port by the output MAC that is notified by the packet flow control unit.

The flow control activity is triggered when the buffer utilization exceeds certain thresholds specified by the dedicated register FMFCT, Register- FMFCT is used to specify the upper and lower thresholds of reserved buffer slot for whole switch.

3.1.4.4 SMI interface

The VT6516 communicates with the external 10/100M PHY and access the PHY register through MDC, MDIO

3.1.4.5 Auto negotiation

The VT6516 communicates with the external 10/100M PHY and access the PHY register through MDC, MDIO

3.1.5 Serial EEPROM interface

4. FUNCTIONAL DESCRIPTION
4.1 Packet Reception and Address recognition

When VT6516 received frames from network, the input control module will receive packet from input MAC module, then get the output port mask from forwarding table control module, request packet buffer from buffer control, write packet from input FIFO to packet buffer scheduled by scheduler module, queue packet to the output queue through queue control module. And update the forwarding table by the source address of the received good packet.

Usually the source MAC address will be learned and stored to forwarding table. If VLAN is configured by user, the frame tag type and VLAN ID will also be learned. The source MAC address bit 47~11 and VLAN ID will be record in the forwarding table entry indexed by source MAC address bit 10~0 or 14~0.
The on chip multicast forwarding configuration registers mainly are for well-known addresses which are listened by CPU. External multicast addresses are for dynamically assigned. Also some static Mac addresses/port mask registers can be configured by CPU, these addresses will also be checked before look up the forward table.

4.2 Packet Forwarding and VLAN

The VT6516's queue control maintains all head and tail pointers for all output ports. Accept the request to queue and dequeue packets from input and output control.
Both queue and dequeue operations take only 1 SRAM access (3 words = 96 bits), because the tail node is stored in the internal register of the queue control
Usually, queue and dequeue operations to a specific output queue can be performed simultaneously. However, mutual exclusion is applied while only one node in this queue

Each port will maintain a packet counter, it increments when packet gets queued through the tail pointer, it decrements when packet de-queued through head pointer. The congestion factor is the queued packet count divided by port media speed. The congestion factor will be used for flow control and multicast, congestion factor should be roughly equal to the time it takes to transmit all the queued packets.
For multicast packet, based on congestion factor, the least congested output port will be queued first. The output control will queue the packet to next least congested output port when it is transmitted, the CPU port will always be last port to be transmitted if the corresponding CPU bit is set in the port mask.

The port speed will be used for cut through forwarding decision. If the packet length is 7ff, it implies the input control try to cut through, queue control will accept or reject by looking whether the input port speed is equal to the output port speed and the output don't have queued packets and any pending transmission. The faster output port (than input port speed) and CPU port is not able to cut through

Broadcast packet, multicast and look up miss packet will forward(multicast) to those ports which is configured by software, but default(dump switching hub) will be all ports(or all ports in that VLAN if VLAN is implemented) except CPU port. Broadcast, multicast packet will check the on chip broadcast forwarding configuration register and multicast forwarding configuration registers first, if multicast address not match any of the multicast forwarding configuration registers then it will look up the external SRAM forwarding table.

When request transfer to or from SDRAM through scheduler, the input control need to derive each burst starting address to bank0 or bank1 information for scheduler to utilize SDRAM bandwidth efficiently. When input FIFO is filled to 12x64 or page boundary or end of frame, the input port control will request DRAM access to write packet. Input FIFO size is 64 bits by 24.

After receiving the grant of queueing (cut-through or store-and-forward), even the bad packet has to be forwarded.

While cut-through, the input control will request the grant of cut-through counter bus for passing the cut through packet count from input port to output port as the whole packet has received.

4.2.1 Cross VLAN Server Port support

The VT6516 support Cross VLAN server port configuration, the following illation show the sample of server ports configuration by set the register of server port mask(14A0H~14A1H), and server ports only enable after the VLAN enabled. The multicast or broadcast frames received from one VLAN group will forward to any server ports and only forward to the ports with same VID.

4.3 Network Management Features

Flow control

The flow control activity is triggered when the buffer utilization exceeds certain thresholds specified by the dedicated register XXXX, Register-XXXX is used to specify the upper and lower thresholds of reserved buffer slot for whole switch.

Under full-duplex mode operation, if the buffer utilization of whole switch has exceeded the upper threshold and the permanent buffer has been used up, a flow control with a predefined pause time value will be sent to the source port to stop the input traffic. If flow control mechanism is not enabled, the public buffer will exhausted so that the further incoming frames will be dropped.

Under half duplex operation, if the buffer utilization of whole switch has exceeded the upper threshold and the permanent buffer has been used up, the port will perform back-pressure based flow control by sending a jam pattern on each incoming frame. If backpressure flow control of the port is not enable, the frame will be dropped.

Sniffer port

The VT6516 support sniffer function for user to monitor the network traffic. The Sniffer port enable can be set for any individual port of sixteen ports. And each sniffer port can set to monitor the traffic coming from any others fifteen port(monitor port). Any packets sent to the monitor ports or transmitted out of monitor port will be forwarded to sniffer port.

Spanning tree support

The VT6516 support the spanning tree protocol (STP). When spanning tree protocol support is enabled, frames from the CPU port having a DA value equal to reserved Bridge Management Group Address for BPDU will be forwarded to the port specified by the CPU.   Frames from other port with a DA equal to reserved Bridge Management Group Address for BPDU will be forwarded to the CPU port.

Every port of the VT6516 can be set to block and listen mode through the CPU interface. In the mode, incoming frames with DA value equal to the reserved Group address for BPDU will be forward to CPU port and other incoming frames with other DA value will be dropped. Outgoing frames with any DA value will be filtered expect DA equal to BPDU.

SECTION II REGISTER MAP

1. REGISTER TABLES

The VT6516 incorporates the required command/status registers and various counters for management purposes. Although the default values of the control registers are predefined in the usual way, there is still a requirement for CPU intervention. All registers are defined as 8 bits so that long registers have to be divided into pieces of 8 bits with the Little-Endian principle, i.e. the lower byte in the lower address. There are only eight registers that are directly accessible for CPU, called the CPU interface registers. They are located with memory mapping in the range of 8000H ~ 8007H for the microprocessor 8031 in the evaluation board. The other registers are called the internal registers that are referenced indirectly by the 16-bit address register with offset 02H ~ 03H in the CPU interface address table. While the 16-bit address register is set to reference to the specific 8-bit internal register, the following read or write operation to the 8-bit data register with offset 01H in the CPU interface address table will cause the specified internal register to be read or written indirectly. Besides, the address register will increase by one automatically to facilitate the successive read/write operation. If the internal register is of size less than 8 bits, the value 0's is always returned for the vacant register space and any write operations to them take no effect.

2 CPU INTERFACE REGISTERS MAP
*Note: register table base = 8000H for the evaluation board.

Description	Type	Offset	Function
Packet Data Register [15:0]	R/W	0H	According to the strapping mode of packet read/write data bus, two types are defined for 8-bit and 16-bit data bus, respectively. For 8-bit CPU, only the low byte of the Packet Data Register is used for packet read/write. For 16-bit CPU, the whole 16-bit Packet Data Register is used for packet read/write.
Data Register [7:0]	R/W	1H	The read or write operation to the 8-bit data register will cause the specified internal register (referenced by the Address Register) to be read or written indirectly. Besides, after the read/write operation, the Address Register will increase by one automatically to facilitate the successive read/write operation.
Address Register [7:0]	R/W	2H	The low-byte address register for the reference to an internal register with 16-bit address.
Address Register [15:8]	R/W	3H	The high-byte address register for the reference to an internal register.
TEST Register 0 [7:0]	W/O	4H	see the description in TEST Register 3
TEST Register 1 [7:0]	W/O	5H	see the description in TEST Register 3
TEST Register 2 [7:0]	W/O	6H	see the description in TEST Register 3
TEST Register 3 [7:0]	W/O	7H	

3 SWITCH INTERNAL REGISTERS MAP

Address (base/offset)	Register Description	Name	Bits	Default Value	R/W
0000H	**SDRAM**				
00H	SDRAM TYPE	SDRAMTYPE	[0]	0	R/W
01H	CAS Latency	CL	[1:0]	2	R/W
02H	SDRAM Operation Mode	RSDM	[3:0]	5	R/W
03H	DIM-Bank 0 Ending Address	END0A	[4:0]	0	R/W
04H	DIM-Bank 1 Ending Address	END1A	[4:0]	0	R/W
05H	DIM-Bank 2 Ending Address	END2A	[4:0]	0	R/W
06H	DIM-Bank 3 Ending Address	END3A	[4:0]	0	R/W
07H	SDRAM Command Drive Strength Configure	SDRAM_DR_CFG	[2:0]	0	R/W
08H	SDRAM Bank Interleaving Disable	BK_IL_DIS	[0]	0	R/W
0800H	**SRAM**				
00H	SRAM Read Command Interleave Disable	SRAM_READ_IL_DIS	[0]	0	R/W
0C00H	Queue control				
00-02H	Free Memory Flow Control Threshold register	FMFCT	[18:0]	0	R/W
03H	Cut Through Enable	CUT_THROUGH_EN	[0]	0	R/W
04H	CPU Port Speed Configuration	CPU_SPD_CFG	[2:0]	0	R/W
10-13H	Congestion Factor of Output Port 0	CONGEST_FCT0	[25:0]	0	R/O
14-17H	Congestion Factor of Output Port 1	CONGEST_FCT1	[25:0]	0	R/O
18-1BH	Congestion Factor of Output Port 2	CONGEST_FCT2	[25:0]	0	R/O
1C-1FH	Congestion Factor of Output Port 3	CONGEST_FCT3	[25:0]	0	R/O
20-23H	Congestion Factor of Output Port 4	CONGEST_FCT4	[25:0]	0	R/O

24-27H	Congestion Factor of Output Port 5	CONGEST_FCT5	[25:0]	0	R/O
28-2BH	Congestion Factor of Output Port 6	CONGEST_FCT6	[25:0]	0	R/O
2C-2FH	Congestion Factor of Output Port 7	CONGEST_FCT7	[25:0]	0	R/O
30-33H	Congestion Factor of Output Port 8	CONGEST_FCT8	[25:0]	0	R/O
34-37H	Congestion Factor of Output Port 9	CONGEST_FCT9	[25:0]	0	R/O
38-3BH	Congestion Factor of Output Port 10	CONGEST_FCT10	[25:0]	0	R/O
3C-3FH	Congestion Factor of Output Port 11	CONGEST_FCT11	[25:0]	0	R/O
40-43H	Congestion Factor of Output Port 12	CONGEST_FCT12	[25:0]	0	R/O
44-47H	Congestion Factor of Output Port 13	CONGEST_FCT13	[25:0]	0	R/O
48-4BH	Congestion Factor of Output Port 14	CONGEST_FCT14	[25:0]	0	R/O
4C-4FH	Congestion Factor of Output Port 15	CONGEST_FCT15	[25:0]	0	R/O
50-53H	Congestion Factor of Output Port 16 (CPU port)	CONGEST_FCT16	[25:0]	0	R/O
1000H	**Buffer control**				
00-02H	Bank 0 Free Pointer	FREE0_PT	[18:0]		R/O
03-05H	Bank 1 Free Pointer	FREE1_PT	[18:0]		R/O
06-08H	Free Memory Block Count	FREEMCNT	[18:0]		R/W
09H	CLEAR All Free Pointers (reset the free buffer pointers according to the SDRAM TYPE)	CFP	[0]		W/O
10H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 0	PORT0_MASK	[7:0]	0	R/O
11H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 1	PORT1_MASK	[7:0]	0	R/O
12H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 2	PORT2_MASK	[7:0]	0	R/O
13H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 3	PORT3_MASK	[7:0]	0	R/O
14H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 4	PORT4_MASK	[7:0]	0	R/O
15H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 5	PORT5_MASK	[7:0]	0	R/O
16H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 6	PORT6_MASK	[7:0]	0	R/O
17H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 7	PORT7_MASK	[7:0]	0	R/O

18H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 8	PORT8_MASK	[7:0]	0	R/O
19H	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 9	PORT9_MASK	[7:0]	0	R/O
1AH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 10	PORT10_MASK	[7:0]	0	R/O
1BH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 11	PORT11_MASK	[7:0]	0	R/O
1CH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 12	PORT12_MASK	[7:0]	0	R/O
1DH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 13	PORT13_MASK	[7:0]	0	R/O
1EH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 14	PORT14_MASK	[7:0]	0	R/O
1FH	PRIVATE MEMORY ALLOCATION BIT MASK for PORT 15	PORT15_MASK	[7:0]	0	R/O
1400H	**Forwarding table control**				
00H	bits of MAC address used as index for forwarding table	HASH_BITS	[2:0]	0	R/W
01-03H	starting SRAM address register for forwarding table base	TBL_BASE	[18:0]	0	R/W
04H	user configured forwarding mode	FWD_MODE	[1:0]	0	R/W
05-07H	user configured port mask	USER_PM	[16:0]	0	R/W
08-09H	port mask for packets sent by CPU	CPU_PM	[15:0]	0	R/W
0AH	CPU port related forwarding configuration.	CPU_FWD_CFG	[2:0]	0	R/W
0BH	port id of sniffer port.	SNIFFER_PID	[3:0]	0	R/W
0C-0EH	monitor port mask	MONITOR_PM	[16:0]	0	R/W
10H	high byte [14:8] of the MAC hash address to be aged	AGE_MAC	[6:0]	0	R/W
11H	low byte [7:0] of the MAC hash address to be aged.	AGE_MAC	[7:0]	0	R/W
12H	aging status	AGING_STATUS	[0]	0	R/O
20H	spanning tree state for PORT 0	PORT0_STP_STATE	[1:0]	0	R/W
21H	spanning tree state for PORT 1	PORT1_STP_STATE	[1:0]	0	R/W
22H	spanning tree state for PORT 2	PORT2_STP_STATE	[1:0]	0	R/W
23H	spanning tree state for PORT 3	PORT3_STP_STATE	[1:0]	0	R/W

24H	spanning tree state for PORT 4	PORT4_STP_STATE	[1:0]	0	R/W
25H	spanning tree state for PORT 5	PORT5_STP_STATE	[1:0]	0	R/W
26H	spanning tree state for PORT 6	PORT6_STP_STATE	[1:0]	0	R/W
27H	spanning tree state for PORT 7	PORT7_STP_STATE	[1:0]	0	R/W
28H	spanning tree state for PORT 8	PORT8_STP_STATE	[1:0]	0	R/W
29H	spanning tree state for PORT 9	PORT9_STP_STATE	[1:0]	0	R/W
2AH	spanning tree state for PORT 10	PORT10_STP_STATE	[1:0]	0	R/W
2BH	spanning tree state for PORT 11	PORT11_STP_STATE	[1:0]	0	R/W
2CH	spanning tree state for PORT 12	PORT12_STP_STATE	[1:0]	0	R/W
2DH	spanning tree state for PORT 13	PORT13_STP_STATE	[1:0]	0	R/W
2EH	spanning tree state for PORT 14	PORT14_STP_STATE	[1:0]	0	R/W
2FH	spanning tree state for PORT 15	PORT15_STP_STATE	[1:0]	0	R/W
80H	port 0 VLAN ID	PORT0_VID	[5:0]	0	R/W
82H	port 1 VLAN ID	PORT1_VID	[5:0]	0	R/W
84H	port 2 VLAN ID	PORT2_VID	[5:0]	0	R/W
86H	port 3 VLAN ID	PORT3_VID	[5:0]	0	R/W
88H	port 4 VLAN ID	PORT4_VID	[5:0]	0	R/W
8AH	port 5 VLAN ID	PORT5_VID	[5:0]	0	R/W
8CH	port 6 VLAN ID	PORT6_VID	[5:0]	0	R/W
8EH	port 7 VLAN ID	PORT7_VID	[5:0]	0	R/W
90H	port 8 VLAN ID	PORT8_VID	[5:0]	0	R/W
92H	port 9 VLAN ID	PORT9_VID	[5:0]	0	R/W
94H	port 10 VLAN ID	PORT10_VID	[5:0]	0	R/W

96H	port 11 VLAN ID	PORT11_VID	[5:0]	0	R/W
98H	port 12 VLAN ID	PORT12_VID	[5:0]	0	R/W
9AH	port 13 VLAN ID	PORT13_VID	[5:0]	0	R/W
9CH	port 14 VLAN ID	PORT14_VID	[5:0]	0	R/W
9EH	port 15 VLAN ID	PORT15_VID	[5:0]	0	R/W
A0-A1H	Server port mask	SRV_PM	[15:0]	0	R/W
A2H	VLAN related forwarding configuration	VLAN_FWD_CFG	[0]	0	R/W
1800H	**PHY control**				
00H	PHY ID	PHYID	[3:0]	0	W/O
01H	PHY register address	PHY_REG_ADDR	[4:0]	0	W/O
02-03H	PHY data register	PHYDATA	[15:0]		R/W
04H	PHY command register	PHYCMD	[0]		W/O
05H	PHY status register	PHYSTS	[1:0]	0	R/O
10H	PORT0 PHY Device Address	PORT0_PHY_ADDR	[4:0]	0	R/W
11H	PORT1 PHY Device Address	PORT1_PHY_ADDR	[4:0]	0	R/W
12H	PORT2 PHY Device Address	PORT2_PHY_ADDR	[4:0]	0	R/W
13H	PORT3 PHY Device Address	PORT3_PHY_ADDR	[4:0]	0	R/W
14H	PORT4 PHY Device Address	PORT4_PHY_ADDR	[4:0]	0	R/W
15H	PORT5 PHY Device Address	PORT5_PHY_ADDR	[4:0]	0	R/W
16H	PORT6 PHY Device Address	PORT6_PHY_ADDR	[4:0]	0	R/W
17H	PORT7 PHY Device Address	PORT7_PHY_ADDR	[4:0]	0	R/W
18H	PORT8 PHY Device Address	PORT8_PHY_ADDR	[4:0]	0	R/W
19H	PORT9 PHY Device Address	PORT9_PHY_ADDR	[4:0]	0	R/W
1AH	PORT10 PHY Device Address	PORT10_PHY_ADDR	[4:0]	0	R/W

1BH	PORT11 PHY Device Address	PORT11_PHY_ADDR	[4:0]	0	R/W
1CH	PORT12 PHY Device Address	PORT12_PHY_ADDR	[4:0]	0	R/W
1DH	PORT13 PHY Device Address	PORT13_PHY_ADDR	[4:0]	0	R/W
1EH	PORT14 PHY Device Address	PORT14_PHY_ADDR	[4:0]	0	R/W
1FH	PORT15 PHY Device Address	PORT15_PHY_ADDR	[4:0]	0	R/W
1C00H	**EEPROM control**				
00H	EEPROM word address	EEWDADDR	[7:0]		W/O
01H	EEPROM data	EEDATA	[7:0]		R/W
02H	EEPROM device address	EEDEVADDR	[7:0]		W/O
03H	EEPROM status register	EESTS	[2:0]		R/O
2000H	**CPU interface**				
00H	interrupt status register	IRQSTS	[3:0]	0	R/W
01H-03H	SRAM address register	SRAMADDR	[18:0]		R/W
04H-07H	SRAM data register	SRAMDATA	[31:0]		R/W
08H	SRAM command register	SRAMCMD	[1:0]		R/W
09H	SRAM status register	SRAMSTS	[1:0]	0	R/O
10H-13H	SDRAM address register	SDRAMADDR	[23:0]		R/W
14H-1BH	SDRAM data register	SDRAMDATA	[63:0]		R/W
1CH	SDRAM command register	SDRAMCMD	[1:0]		R/W
1DH	SDRAM status register	SDRAMSTS	[1:0]	0	R/O
20H	Write packet command	WR_PKT_CMD	[2:0]		W/O
21H	Packet Abort	ERR_ABORT	[0]		W/O
30H	bits [47:40] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:0]	0	R/W
31H	bits [39:32] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:0]	0	R/W
32H	bits [31:24] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:0]	0	R/W

33H	bits [23:16] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:0]	0	R/W
34H	bits [15:8] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:0]	0	R/W
35H	bits [7:4] of switch base MAC address [47:0]	SWITCH_MAC_BASE	[7:4]	0	R/W
40H	interrupt mask register	IRQSTS_MASK	[3:0]	4'b1111	R/W
50H	CPU Soft Reset for the whole switch chip reset	CPU_SOFT_RESET	[0]	1	R/W
51H	Revision Control Register	REVISION_CTL	[7:0]	0	R/O
2400H	**MAC & I/O Control Module of Port 0**				
00H	configurable preamble bytes	PREAM_CFG	[2:0]	7	R/W
01H	configurable frame gap in di bits for 1st interval	IFG_CFG	[5:0]	32	R/W
02H	Backoff configuration	BOFFCFG	[4:0]	5'b10000	R/W
03H	MAC media type configuration	MACCFG	[3:0]	0	R/W
04H	IO port enable	IO_CFG	[1:0]	0	R/W
10H-13H	received good packet count	RCV_GOOD_PKT	[31:0]	0	R/O
14H-17H	received bad packet count	RCV_BAD_PKT	[31:0]	0	R/O
18H-1BH	drop packet counter	DROP_PKT	[31:0]	0	R/O
1CH-1FH	sent good packet count	XMT_GOOD_PKT	[31:0]	0	R/O
20H-23H	sent bad packet counter	XMT_BAD_PKT	[31:0]	0	R/O
2800H	**MAC & I/O Control Module of Port 1**	as same as Port 0			
2C00H	**MAC & I/O Control Module of Port 2**	as same as Port 0			
3000H	**MAC & I/O Control Module of Port 3**	as same as Port 0			
3400H	**MAC & I/O Control Module of Port 4**	as same as Port 0			
3800H	**MAC & I/O Control Module of Port 5**	as same as Port 0			
3C00H	**MAC & I/O Control Module of Port 6**	as same as Port 0			
4000H	**MAC & I/O Control Module of Port 7**	as same as Port 0			
4400H	**MAC & I/O Control Module of Port 8**	as same as Port 0			
4800H	**MAC & I/O Control Module of Port 9**	as same as Port 0			
4C00H	**MAC & I/O Control Module of Port 10**	as same as Port 0			

5000H	**MAC & I/O Control Module of Port 11**	as same as Port 0				
5400H	**MAC & I/O Control Module of Port 12**	as same as Port 0				
5800H	**MAC & I/O Control Module of Port 13**	as same as Port 0				
5C00H	**MAC & I/O Control Module of Port 14**	as same as Port 0				
6000H	**MAC & I/O Control Module of Port 15**	as same as Port 0				
6400H	**CPU IO Control Module**					
00H	CPU packet read byte count register bits [7:0]	PKT_BYTE_CNT	[7:0]	0	R/O
01H	CPU packet read byte count register bits [10:8]	PKT_BYTE_CNT	[10:8]	0	R/O
02H	CPU packet read status register	RD_PKT_STATUS	[1:0]	0	R/O
03H	Packet source port ID	PKT_SRC_PORT	[3:0]	0	R/O
04H	CPU IO port configuration register	CPUIO_CFG	[1:0]	0	R/W
10H	CPU packet write status register	WR_PKT_STATUS	[2:0]	0	R/O

4. DETAIL OF SWITCH REGISTER
4.1 Registers of SDRAM Control Module

* Base Address: 0000H

Addres s (offset)	Function	Register Name	Bits	Defau lt Value	R/ W
00H	SDRAM TYPE: 0: 16Mbit SDRAM chip (default) 1: 64Mbit This register has to be specified before initialization of the buffer control because the Bank 1 free buffer pointer should have initial value 130 for 16Mbit SDRAM, or, initial value 131 for 64Mbit SDRAM.	SDRAMTYPE	[0]	0	R/W
01H	CAS Latency for read operation: 2'b00: latency 1 2'b01: latency 2 2'b10: latency 3 (default) This latency specifies the required delay between the CAS cycle and the first read cycle. Note that the CAS latency has to be specified before using RSDM in SDRAM initialization.	CL	[1:0]	2	R/W

02H	**SDRAM Operation Mode:** For the bits [2:0], the operation modes are defined as follows: 3'b000: Normal SDRAM Mode 3'b001: NOP Command Enable 3'b010: Precharge All Banks 3'b011: MSR Enable (Mode Register Set Enable) 3'b100: CBR Refresh Cycle Enable others: idle for power-up For the bit [3], it is called REFRESH_EN, defined as follows: 0: turn off hardware refresh cycle (default) 1: turn on hardware refresh cycle After the last refresh operation issued by software in the initialization cycle, software should enable the bit "REFRESH_EN" immediately to notify SDRAM control module 'sdramctl' to start generating refresh cycle periodically. The initialization of SDRAM control module is illustrated as follows: SDRAMTYPE ← 0 : 16Mb CL ← 1 : read latency = 2 (3) delay 1 s (4) RSDM ← 1 : NOP (5) delay 1 s (6) RSDM ← 2 : Precharge (7) delay 1 s (8) loop 7 times RSDM ← 4 : Refresh delay 1 s RSDM ← 1 : NOP delay 1 s (9) RSDM ← 0CH : Refresh & turn on hardware refresh (10) delay 1 s (11) RSDM ← 0BH : Mode Register Set Enable (12) delay 1 s (13) RSDM ← 08H : Normal SDRAM Mode (14) END0A ← 0x04 : DIM bank 0 ending address = 32MB (15) END1A ← 0x08 : DIM bank 1 ending address = 64MB (16) END2A ← 0x0C : DIM bank 2 ending address = 96MB (17) END3A ← 0x10 : DIM bank 3 ending address = 128MB	RSDM	[3:0]	5	R/W
03H	Bits [27:23] of DIMM Bank 0 Ending Address For the case that there are two 32MB SDRAM modules plugged in DIMM slot 0 and two 16MB SDRAM modules plugged in DIMM slot 1, assign the registers as follows END0A = 04H to indicate the ending address of DIMM Bank 0 is at 2^{25} (32MB) END1A = 08H to indicate the ending address of DIMM Bank 1 is at 2^{26} (64MB) END2A = 0AH to indicate the ending address of DIMM Bank 0 is at $2^{26}+2^{24}$ (80MB) END3A = 0CH to indicate the ending address of DIMM Bank 0 is at $2^{26}+2^{25}$ (96MB)	END0A	[4:0]	0	R/W
04H	Bits [27:23] of DIMM Bank 1 Ending Address (see END0A)	END1A	[4:0]	0	R/W
05H	Bits [27:23] of DIMM Bank 2 Ending Address (see END0A)	END2A	[4:0]	0	R/W
06H	Bits [27:23] of DIMM Bank 3 Ending Address (see END0A)	END3A	[4:0]	0	R/W

| 07H | **SDRAM Command Drive Strength Configure**
bit0: RDCSDV --- SDRAM Chip Select Drive Strength
bit1: RMADV --- SDRAM MA drive strength
       (including RAS,CAS,WE,MA,BA)
bit2: RMDDV --- SDRAM MD drive strength | SDRAM_DR_
CFG | [2:0] | 0 | R/W |
| 08H | **SDRAM Bank Interleaving Disable**
0: enable interleaving (default)
1: disable interleaving | BK_IL_DIS | [0] | 0 | R/W |

4.2 Registers of SRAM Control Module

* Base Address: 0800H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00H	SRAM Read Command Interleave Disable 0: enable interleaving (default) 1: disable interleaving	SRAM_READ_IL_DIS	[0]	0	R/W

4.3 Registers of Queue Control Module

* Base Address: 0C00H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00-02H	**Free Memory Flow Control Threshold register** As FREEMCNT(a register in buffer control) < FMFCT, the congestion control function will be triggered to command the TMAC module of the source port, destined to a congested port, to send out a flow control frame for full duplex mode, or to make back-pressure for half duplex mode. See the context about congestion control for details. Larger the threshold value more sensitive the congestion control mechanism, i.e. maybe poor utilization for packet buffers but larger packet loss rate. Smaller the threshold value less sensitive the congestion control mechanism, i.e. maybe good utilization for packet buffers but smaller packet loss rate. It depends on the network configuration and traffic pattern. The recommended threshold value is 256.	FMFCT	[18:0]	0	R/W
03H	Cut Through Enable 0: Disable Cut Through (default) 1: Enable Cut Through Note: REMEMBER to enable the cut-through function to improve the switching latency. For 100Mbps input port, the smallest latency for cut-through is 288 bytes time (288x8x10 ns). For 10Mbps input port, the smallest latency for cut-through is 96 bytes time (96x8x100 ns).	CUT_THROUGH_EN	[0]	0	R/W

04H	**CPU Port Speed Configuration** 3'b000: 1 Mbit (default) 3'b001: 5 Mbit 3'b010: 10 Mbit 3'b011: 20 Mbit 3'b100: 40 Mbit 3'b101: 50 Mbit 3'b110: 80 Mbit 3'b111: 100 Mbit This register is used to calculate the congestion factor of the CPU port, that is the quotient of the accumulated byte count of the CPU output queue to the specified CPU port speed. While the congestion control is triggered, the output ports with congestion factor larger than the average will enter into the congestion control mode.	CPU_SPD_CFG	[2:0]	0	R/W
10-13H	Congestion Factor of Output Port 0 The congestion factor, i.e. the quotient of the accumulated byte count of the output queue to the port speed, for each of 16 Ethernet ports is calculated by the flow control module. While the congestion control is triggered, the output ports with congestion factor larger than the average will enter into the congestion control mode.	CONGEST_FCT0	[25:0]	0	R/O
14-17H	Congestion Factor of Output Port 1	CONGEST_FCT1	[25:0]	0	R/O
18-1BH	**Congestion Factor of Output Port 2**	CONGEST_FCT2	[25:0]	0	R/O
1C-1FH	**Congestion Factor of Output Port 3**	CONGEST_FCT3	[25:0]	0	R/O
20-23H	**Congestion Factor of Output Port 4**	CONGEST_FCT4	[25:0]	0	R/O
24-27H	**Congestion Factor of Output Port 5**	CONGEST_FCT5	[25:0]	0	R/O
28-2BH	**Congestion Factor of Output Port 6**	CONGEST_FCT6	[25:0]	0	R/O
2C-2FH	**Congestion Factor of Output Port 7**	CONGEST_FCT7	[25:0]	0	R/O
30-33H	**Congestion Factor of Output Port 8**	CONGEST_FCT8	[25:0]	0	R/O
34-37H	**Congestion Factor of Output Port 9**	CONGEST_FCT9	[25:0]	0	R/O
38-3BH	**Congestion Factor of Output Port 10**	CONGEST_FCT10	[25:0]	0	R/O
3C-3FH	**Congestion Factor of Output Port 11**	CONGEST_FCT11	[25:0]	0	R/O
40-43H	**Congestion Factor of Output Port 12**	CONGEST_FCT12	[25:0]	0	R/O
44-47H	**Congestion Factor of Output Port 13**	CONGEST_FCT13	[25:0]	0	R/O
48-4BH	**Congestion Factor of Output Port 14**	CONGEST_FCT14	[25:0]	0	R/O

Address	Function	Register Name	Bits	Default Value	R/W
4C-4FH	**Congestion Factor of Output Port 15**	CONGEST_FCT15	[25:0]	0	R/O

50-53H	**Congestion Factor of Output Port 16 (CPU port)**
	
	There are 11 bits are used for reading FREEMCNT. Only 15 bits are used as CPU port's congestion factor. The read sequence of CONGEST_FCT16[14:0] is as follows:
	1. Read 0C50H to get the lowest byte.
	  CONGEST16[7:0] = HD[7:0]
	2. Read 0C51H to get the other 7 bits.
	  CONGEST16[14:8] = HD[6:0]
		CONGEST_FCT16	[25:0]	0	R/O

4.4 Registers of Buffer Control Module

* Base Address: 1000H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00-02H	**Bank 0 Free Pointer** This register is initialized according to SDRAMTYPE while the CFP is written. For 16/64Mbit SDRAM, its value is always 128 because the bank 0 free list follows the private buffer pool of buffer entries 0~127. The free buffers with starting address at the SDRAM even bank should be linked into this free list to improve the SDRAM bandwidth utilization. However, if the free buffers are misplaced, they will returned to the adequate free lists after their first release by the output port control. Internally, the free pointer refers to the ID of the 1st free buffer, rather than its physical address in SRAM (that is equal to ID*3).	FREE0_PT	[18:0]		R/O
03-05H	**Bank 1 Free Pointer** This register is initialized according to SDRAMTYPE while the CFP is written. For 16Mbit SDRAM, its value is 130. For 64Mbit SDRAM, its value is 131. The fixed buffer size is 1536 bytes. Because the page size is 2KB for 16Mbit SDRAM, the first public buffer of bank 1 is the 130th entry located at page 1. Because the page size is 4KB for 64Mbit SDRAM, the first public buffer of bank 1 is the 131st entry located at page 1.	FREE1_PT	[18:0]		R/O

06-08H	**Free Memory Block Count**
	
	It is an integer value <= sizeof(SDRAM) / 1.5KB. It has to be specified at the switch initialization stage.
	To fix the bug of reading FREEMCNT in VT3061A, the bit mapping for reading FREEMCNT is modified in VT3061B. The write sequence of FREEMCNT is also to write data to 1006H, 1007H, 1008H. However, the read sequence of FREEMCNT has to read data from
	(1) Read 1006H to get the lowest byte, and also lock the counter information, .i.e. FREEMCNT[7:0] = HD[7:0]
	(2) Read 0C52H to get the second byte, i.e.
	  FREEMCNT[15:8] = HD[7:0]
	(3) Read 0C53H to get the FREEMCNT[17:16]
	  FREEMCNT[17:16] = HD[1:0]
	(4) Read 0C51H to get the FREEMCNT[18]
	  FREEMCNT[18] = HD[7]
		FREEMCNT	[18:0]		R/W

09H	**CLEAR All Free Pointers** Write to this register will reset the two free buffer pointers according to the SDRAMTYPE. It is the only way to program the FREE0_PT and FREE1_PT. This command should be taken after the SDRAMTYPE has been specified.	CFP	[0]		W/O
10H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 0** Each bit corresponds to a private packet buffer. This mask register will be cleared to bit pattern 0000-0000 while system reset. The 8 private buffers for the port K are that of entry IDs (K*8) ~ (K*8+7). But, the CPU IO port has not private buffers.	PORT0_MASK	[7:0]	0	R/O
11H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 1**	PORT1_MASK	[7:0]	0	R/O
12H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 2**	PORT2_MASK	[7:0]	0	R/O
13H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 3**	PORT3_MASK	[7:0]	0	R/O
14H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 4**	PORT4_MASK	[7:0]	0	R/O
15H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 5**	PORT5_MASK	[7:0]	0	R/O
16H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 6**	PORT6_MASK	[7:0]	0	R/O
17H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 7**	PORT7_MASK	[7:0]	0	R/O
18H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 8**	PORT8_MASK	[7:0]	0	R/O
19H	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 9**	PORT9_MASK	[7:0]	0	R/O
1AH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 10**	PORT10_MASK	[7:0]	0	R/O
1BH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 11**	PORT11_MASK	[7:0]	0	R/O
1CH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 12**	PORT12_MASK	[7:0]	0	R/O
1DH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 13**	PORT13_MASK	[7:0]	0	R/O
1EH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 14**	PORT14_MASK	[7:0]	0	R/O
1FH	**PRIVATE MEMORY ALLOCATION BIT MASK for PORT 15**	PORT15_MASK	[7:0]	0	R/O

4.5 Registers of Forwarding Control Module

* Base Address: 1400H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00H	**bits of MAC address used as index for forwarding table** 3'b000: use MAC address bit 10-0 (default) 3'b001: use MAC address bit 11-0 3'b010: use MAC address bit 12-0 3'b011: use MAC address bit 13-0 3'b100: use MAC address bit 14-0 others, use MAC address bit 10-0 This register specifies the lookup hash key. For example, if the MAC address bits [14:0] is used as the hash key, there must be 32K 96-byte table entries necessary to be allocated in the upper part of SRAM for destination MAC lookup and source MAC learning.	HASH_BITS	[2:0]	0	R/W
01-03H	**starting SRAM address register for forwarding table base** The forwarding table should be located above the linked buffer entries in the SRAM. The starting address of the forwarding table is specified by TBL_BASE in unit of 32-bit word. The occupied size is determined by HASH_BITS. For example, if there are maximum 5461 buffers entries used for 8MB SDRAM, the minimum forwarding table base is 5461*3 because each linked buffer entry is of size 96 bits (3 words).	TBL_BASE	[18:0]	0	R/W

04H	**user configured Forwarding Mode**
	bit 0 – if using the specified forwarding mask without lookup
	    (default: 0, to take lookup without specified mask)
	bit 1 – if not forwarding packets destined to congested ports
	    (default: 0, not to filter packets by congestion factors)
	
	If FWD_MODE[0] = 1, the incoming packets would not be forwarded with table lookup. However, the USER_PM is used as the forwarding mask if the incoming packets are not from the CPU port. For broadcast & lookup-miss packets, the USER_PM is returned by Forwarding Control to IO Control as the lookup result. If FWD_MODE[1] = 1, the incoming packets would not be forwarded to the congested ports whose congestion factors are larger than 511.
		FWD_MODE	[1:0]	0	R/W

05-07H	**USER configured Port Mask** The USER_PM is used as the lookup result for the incoming packets from Ethernet ports in the following cases: FWD_MODE = 1 FWD_MODE = 0, VLAN is off, STP_STATE is "forward", and this is a broadcast packet or a lookup-miss packet	USER_PM	[16:0]	0	R/W
08-09H	**port mask for packets sent by CPU** The CPU_PM is used as the lookup result for the incoming packets from the CPU port without regard to packet's DMAC.	CPU_PM	[15:0]	0	R/W

0AH	**CPU port related forwarding configuration**
	bit 0 – enable forwarding broadcast packets with
	    DMAC=0xffffffffffff to CPU
	    (default = 0 : disable)
	bit 1 – enable forwarding spanning-tree packets to CPU
	    (default = 0 : disable)
	bit 2 – enable forwarding unicast packets with DMAC =
	    switch MAC base to CPU
	    (default = 0 : disable)
	
	The three register bits are used to enable/disable forwarding the
	above three types of frames to the CPU port. Note that for a
	lookup-miss packet, whether it will be forwarded to the CPU port is
	determined by the bit USER_PM[16], rather than this register.
		CPU_FWD_C
		FG
			[2:0]	0	R/W

0BH	**Sniffer Port ID** This register is valid only if MONITOR_PM is not all 0's. The default value of MONITOR_PM is all 0's to disable the Sniffer function.	SNIFFER_PID	[3:0]	0	R/W
0C-0EH	**Monitor Port Mask** This register is used to specify which ports to be monitored by Sniffer so that all packets forwarded from/to the monitored ports are also made a copy sent to the Sniffer port. The Sniffer function is enabled only if MONITOR_PM is not all 0's. The default value of MONITOR_PM is all 0's to disable the sniffer function.	MONITOR_P M	[16:0]	0	R/W
10H	**high byte [14:8] of the MAC hash address to be aged** AGE_MAC is in the hash-key format of bits [14:0]. The high byte [14:8] is stored in the AGE_MAC register of offset 10H. The low byte [7:0] is stored in the AGE_MAC register of offset 11H. A write to the register of offset 11H will trigger an aging operation that decreases by one the age count of the corresponding forwarding table entry. A forwarding table entry with age count = 0 is an invalid entry, i.e. this entry is available for the source MAC learning. AGE_MAC is reset to all 0's after aging.	AGE_MAC	[6:0]	0	R/W
11H	**low byte [7:0] of the MAC hash address to be aged** (see the above)	AGE_MAC	[7:0]	0	R/W
12H	Aging Status 0, idle or done (default) 1: aging in progress After an aging command is issued, the status is recorded in this register. The next age command can only be issued as the status changes from 1 (in-progress) to 0 (done).	AGING_STAT US	[0]	0	R/O

20H	**spanning tree state for PORT 0** 2'b00 – blocking state (default) 2'b01 – listening state 2'b10 - learning state 2'b11 – forwarding state The forwarding operation in each Ethernet port is controlled by its associated spanning tree state. In blocking or listening state, the incoming packets will not trigger any DMAC lookup operation and SMAC learning operation. In learning state, the incoming packets will not trigger DMAC lookup operation, but the SMAC learning operation will be triggered for CRC-OK packets. Only in forwarding state, an incoming packet will trigger DMAC lookup operation while the first 24 bytes are received, and it will trigger the SMAC learning operation while the whole packet is received with good CRC. For the 802.1d spanning tree algorithm, a blocked port for loop avoidance should enter the blocking state so that any incoming packets are filtered without forward. A normal port that does not cause any loop should be in the forwarding state.	PORT0_STP_ STATE	[1:0]	0	R/W
21H	**spanning tree state for PORT 1**	PORT1_STP_ STATE	[1:0]	0	R/W
22H	**spanning tree state for PORT 2**	PORT2_STP_ STATE	[1:0]	0	R/W
23H	**spanning tree state for PORT 3**	PORT3_STP_ STATE	[1:0]	0	R/W
24H	**spanning tree state for PORT 4**	PORT4_STP_ STATE	[1:0]	0	R/W
25H	**spanning tree state for PORT 5**	PORT5_STP_ STATE	[1:0]	0	R/W
26H	**spanning tree state for PORT 6**	PORT6_STP_ STATE	[1:0]	0	R/W
27H	**spanning tree state for PORT 7**	PORT7_STP_ STATE	[1:0]	0	R/W
28H	**spanning tree state for PORT 8**	PORT8_STP_ STATE	[1:0]	0	R/W
29H	**spanning tree state for PORT 9**	PORT9_STP_ STATE	[1:0]	0	R/W
2AH	**spanning tree state for PORT 10**	PORT10_STP _STATE	[1:0]	0	R/W
2BH	**spanning tree state for PORT 11**	PORT11_STP _STATE	[1:0]	0	R/W
2CH	**spanning tree state for PORT 12**	PORT12_STP _STATE	[1:0]	0	R/W
2DH	**spanning tree state for PORT 13**	PORT13_STP _STATE	[1:0]	0	R/W
2EH	**spanning tree state for PORT 14**	PORT14_STP _STATE	[1:0]	0	R/W
2FH	**spanning tree state for PORT 15**	PORT15_STP _STATE	[1:0]	0	R/W
80H	**port 0 VLAN ID** The VLAN feature is enabled only when all port VID's are configured to a valid (non-zero) VID.	PORT0_VID	[5:0]	0	R/W
82H	**port 1 VLAN ID**	**PORT1_VID**	**[5:0]**	**0**	**R/W**
84H	**port 2 VLAN ID**	**PORT2_VID**	**[5:0]**	**0**	**R/W**
86H	**port 3 VLAN ID**	**PORT3_VID**	**[5:0]**	**0**	**R/W**
88H	**port 4 VLAN ID**	**PORT4_VID**	**[5:0]**	**0**	**R/W**

8AH	port 5 VLAN ID	PORT5_VID	[5:0]	0	R/W
8CH	port 6 VLAN ID	PORT6_VID	[5:0]	0	R/W
8EH	port 7 VLAN ID	PORT7_VID	[5:0]	0	R/W
90H	port 8 VLAN ID	PORT8_VID	[5:0]	0	R/W
92H	port 9 VLAN ID	PORT9_VID	[5:0]	0	R/W
94H	port 10 VLAN ID	PORT10_VID	[5:0]	0	R/W
96H	port 11 VLAN ID	PORT11_VID	[5:0]	0	R/W
98H	port 12 VLAN ID	PORT12_VID	[5:0]	0	R/W
9AH	port 13 VLAN ID	PORT13_VID	[5:0]	0	R/W
9CH	port 14 VLAN ID	PORT14_VID	[5:0]	0	R/W
9EH	port 15 VLAN ID	PORT15_VID	[5:0]	0	R/W
A0-A1H	**Server Port Mask** The SRV_PM is used only when VLAN is enabled. As the VLAN feature is enabled (i.e. all port VID's are valid (non-zero)), the behavior is described for the following scenarios: A broadcast or lookup-miss packet will be forwarded to the ports of same VLAN ID. If CPU_FWD_CFG[0]=1, the broadcast packet will also be forwarded to CPU port. A unicast packet destined to different VLAN will be forwarded to CPU port if VLAN_FWD_CFG[0]=1. A unicast packet destined to another port in the same VLAN will be forwarded in a unicast manner. The SRV_PM should be set for the Server ports that respond to carry cross VLAN packets. It is recommended that all packets from the Server stations have not any embedded VID. Only the cross VLAN packets through the Server ports (the corresponding SRV_PM bits are on) within different VLAN domain must carry VID. In the source-MAC learning procedure, for packets with tagged VID, the corresponding forwarding table entry will have NOT the tagging bit on to make the outgoing packets destined to it with VID tagged. All valid forwarding table entries should have non-zero VID.	SRV_PM	[15:0]	0	R/W
A2H	**VLAN related forwarding configuration** This register bit is used to enable those packets destined to a different VLAN also to be forwarded to the CPU port. This scenario happens when the following conditions hold simultaneously: The destination MAC address is found in forwarding table (lookup hit), and this entry is not static. Note that if the entry is static, its priority is highest and the destination ports are fully determined by the port mask field in the entry so that forwarding to CPU is not necessary. The source VID differs from the destination VID.	VLAN_FWD_CFG	[0]	0	R/W

4.6 Registers of PHY Control Module

* Base Address: 1800H

Addres s (offset)	Function	Register Name	Bits	Defau lt Value	R/ W
00H	**PHY ID** This is used to specify which PHY device is the objective of the following MII commands. There are maximum 16 RMII PHY devices.	PHYID	[3:0]	0	W/ O
01H	**PHY register address** In each PHY device, there are maximum 32 MII management registers accessible by the CPU. The PHY_REG_ADDR register is used to specify which one is the objective of the following access command.	PHY_REG _ADDR	[4:0]	0	W/ O
02-03H	**PHY data register** Each PHY management register is 16 bits. Every data access to a PHY management register is in unit of 16 bits, stored in this register.	PHYDAT A	[15:0]		R/ W
04H	**PHY command register** 1: read 0: write Write 0 to this register will cause a write operation to the PHY management register (specified by the PHY_REG_ADDR) of the PHY device (specified by the PHYID). Write 1 to this register will cause a read operation. A read or write operation takes about 0.4 ms so that the CPU has to read the PHYSTS register periodically to check if the issued command is complete.	PHYCMD	[0]		W/ O
05H	**PHY status register** 2'b00: idle 2'b01: busy 2'b10: complete This register indicates the status of the PHY control module. Initially, the PHY control module is in the idle status. While a read or write command is issued by writing 1/0 to the PHYCMD register, PHYSTS becomes "busy" immediately, and goes into the "complete" status as this operation finishes. Then, a following "read status" command will cause it back to the "idle" status, or a following read/write command will cause it into the "busy" status.	PHYSTS	[1:0]	0	R/O
10H	**PORT0 PHY Device Address** The pair of 5-bit PHY device address and 5-bit register address forms a unique access address to a PHY device's register. Each PHY device has a unique device address that is identified by the PHYID. In the system initialization, the CPU should write PHY device addresses, corresponding to every PHY devices, to registers PORT[0..15]_PHY_ADDR, that may be recorded in the EEPROM or code ROM.	PORT0_P HY_ADD R	[4:0]	0	R/ W
11H	**PORT1 PHY Device Address**	PORT1_P HY_ADD R	[4:0]	0	R/ W
12H	**PORT2 PHY Device Address**	PORT2_P HY_ADD R	[4:0]	0	R/ W

13H	**PORT3 PHY Device Address**	PORT3_PHY_ADDR	[4:0]	0	R/W
14H	**PORT4 PHY Device Address**	PORT4_PHY_ADDR	[4:0]	0	R/W
15H	**PORT5 PHY Device Address**	PORT5_PHY_ADDR	[4:0]	0	R/W
16H	**PORT6 PHY Device Address**	PORT6_PHY_ADDR	[4:0]	0	R/W
17H	**PORT7 PHY Device Address**	PORT7_PHY_ADDR	[4:0]	0	R/W
18H	**PORT8 PHY Device Address**	PORT8_PHY_ADDR	[4:0]	0	R/W
19H	**PORT9 PHY Device Address**	PORT9_PHY_ADDR	[4:0]	0	R/W
1AH	**PORT10 PHY Device Address**	PORT10_PHY_ADDR	[4:0]	0	R/W
1BH	**PORT11 PHY Device Address**	PORT11_PHY_ADDR	[4:0]	0	R/W
1CH	**PORT12 PHY Device Address**	PORT12_PHY_ADDR	[4:0]	0	R/W
1DH	**PORT13 PHY Device Address**	PORT13_PHY_ADDR	[4:0]	0	R/W
1EH	**PORT14 PHY Device Address**	PORT14_PHY_ADDR	[4:0]	0	R/W
1FH	**PORT15 PHY Device Address**	PORT15_PHY_ADDR	[4:0]	0	R/W

4.7 Registers of EEPROM Control Module

* Base Address: 1C00H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00H	**EEPROM word address** For a 256-byte EEPROM device, an 8-bit data object is identified with this register. For a 512-byte EEPROM device, an 8-bit data object is identified with this register plus EEDEVADDR[1]. For a 1024-byte EEPROM device, an 8-bit data object is identified with this register plus EEDEVADDR[2:1], vice versa.	EEWDADDR	[7:0]		W/O
01H	**EEPROM data** Every data access to EEPROM is in unit of 8 bits, stored in this register.	EEDATA	[7:0]		R/W
02H	**EEPROM device address** bit 7-4 : device type id (EEPROM 1010) bit 3-1 : device id bit 0 : r/w command , value 0: write; value 1: read The triple of 4-bit PHY device type ID, 3-bit device ID, and 7-bit word address forms a unique access address to an 8-bit EEPROM data object. This register's bit 0 is used to specify the command type: 0 for write and 1 for read. A read or write operation takes about 0.4 ms so that the CPU has to read the EEYSTS register periodically to check if the issued command is "complete without error" or "ack error".	EEDEVADDR	[7:0]		W/O
03H	**EEPROM status register** 3'b000: idle 3'b001: busy 3'b010: complete without error 3'b100: ack error This register indicates the status of the EEPROM control module. Initially, the EEPROM control module is in the idle status. While a read or write command is issued by writing 1/0 to EEDEVADDR[0], EESTS becomes "busy" immediately, and goes into the "complete" status as this operation finishes or into the "ack error" as an acknowledge error happens. Then, a following "read status" command will cause it back to the "idle" status, or a following read/write command will cause it into the "busy" status.	EESTS	[2:0]		R/O

4.8 Registers of CPU Interface Module

* Base Address: 2000H

Address (offset)	Function	Register Name	Bits	Default Value	R/W

00H	**interrupt status register** bit 0: interrupt indication for read/write PHY command complete bit 1: interrupt indication for read/write EEPROM command complete or error bit 2: interrupt indication for CPU IO port receiving an incoming packet bit 3: interrupt indication for CPU IO port finishing the transmission of an outgoing packet To clear an interrupt, write 1 to the corresponding IRQSTS bit. However, write 0 will not cause any change on that interrupt.	IRQSTS	[3:0]	0	R/W
01H-03H	**SRAM address register** The data object addressed is in unit of 32 bits. The maximum allowable SRAM size is 1MB. For SRAM direct access, the CPU has to (1) set SRAMADDR & SRAMDATA, (2) issue read/write command by SRAMCMD, and (3) check command status by SRAMSTS.	SRAMADDR	[18:0]		R/W
04H-07H	**SRAM data register**	SRAMDATA	[31:0]		R/W
08H	**SRAM command register** 2'b00 : nop 2'b01 : read 2'b10 : write To make a direct read to SRAM, write SRAMCMD by 2'b01. To make a direct write to SRAM, write SRAMCMD by 2'b10. Read/Write will cause the SRAMSTS = "busy" immediately. As it is done, SRAMSTS = "done". A following read to SRAMSTS will clear it to "idle".	SRAMCMD	[1:0]		R/W
09H	**SRAM status register** 2'b01 : read/write command done 2'b10 : busy (read/write in progress) 2'b00 : idle	SRAMSTS	[1:0]	0	R/O
10H-13H	**SDRAM address register** The data object addressed is in unit of 64 bits. The maximum allowable SDRAM size is 128MB. For SDRAM direct access, the CPU has to (1) set SDRAMADDR & SDRAMDATA, (2) issue read/write command by SDRAMCMD, and (3) check command status by SDRAMSTS.	SDRAMADDR	[23:0]		R/W
14H-1BH	SDRAM data register	SDRAMDATA	[63:0]		R/W
1CH	**SDRAM command register** 2'b00 : nop 2'b01 : read 2'b10 : write To make a direct read to SDRAM, write SDRAMCMD by 2'b01. To make a direct write to SDRAM, write SDRAMCMD by 2'b10. Read/Write will cause the SDRAMSTS = "busy" immediately. As it is done, SDRAMSTS = "done". A following read to SDRAMSTS will clear it to "idle".	SDRAMCMD	[1:0]		R/W
1DH	**SDRAM status register** 2'b01 : read/write command done 2'b10 : busy (read/write in progress) 2'b00 : idle	SDRAMSTS	[1:0]	0	R/O

20H	**Write packet command** 3'b100 : end of frame with the remaining data size = 2 bytes 3'b101 : end of frame with the remaining data size = 1 byte (that is the low byte as using 16-bit write) 3'b000 : idle 3'b001 : start of frame for the next write 3'b010 : middle of frame for the next write 3'b011 : abort the unfinished packet write CPU should write this command register before repeatedly writing 8/16 bit packet data VIA the ISA/IDE bus (with a2, a1, a0 = 000).	WR_PKT_CMD	[2:0]		W/O
21H	**Packet Abort** Write this register to drop an incoming packet ready to be read by CPU.	ERR_ABORT	[0]		W/O
30H	**bits [47:40] of switch base MAC address [47:0]** Each port in the switch IC has a unique MAC address with the port ID as address bits [3:0] and the same MAC base bits [47:4], specified by the register SWITCH_MAC_BASE[47:4].	SWITCH_MAC_BASE	[7:0]	0	R/W
31H	**bits [39:32] of switch base MAC address [47:0]**	SWITCH_MAC_BASE	[7:0]	0	R/W
32H	**bits [31:24] of switch base MAC address [47:0]**	SWITCH_MAC_BASE	[7:0]	0	R/W
33H	**bits [23:16] of switch base MAC address [47:0]**	SWITCH_MAC_BASE	[7:0]	0	R/W
34H	**bits [15:8] of switch base MAC address [47:0]**	SWITCH_MAC_BASE	[7:0]	0	R/W
35H	**bits [7:4] of switch base MAC address [47:0]**	SWITCH_MAC_BASE	[7:4]	0	R/W
40H	**interrupt mask register** bit 0: PHY interrupt mask bit 1: EEPROM interrupt mask bit 2: packet received interrupt mask bit 3: packet sent interrupt mask The four interrupts can be masked individually. The value 0 indicates "Masked", and value 1 (default) indicates "Unmasked".	IRQSTS_MASK	[3:0]	4'b1111	R/W

| 50H | **CPU Soft Reset for the whole switch chip reset**
For Read
  0: soft reset in progress
  1: soft reset done
For Write,
  any value will trigger the whole chip reset

The soft reset is similar to power-on reset for the switch chip, except that it is asserted by writing any value to this register. The CPU soft reset has to take 16 RCLK50 cycles, i.e. 320ns, to make the switch chip being reset and ready to CPU. For 8MHz 8051 CPU that an instruction cycle is 1.5 s, it needs to wait for 4 CPU instruction cycles to continue after the soft reset. Or, CPU can read this register CPU_SOFT_RESET until value 1 is returned.
Note that reading this register will not cause the address register to increment automatically. So, consecutively reading from 2050H to 2051H should not be applied. That is, any reading to 2051H has to specify the address explicitly. | CPU_SOFT_RESET | [0] | 1 | R/W |
| 51H | **Revision ID Register**

This register is used to record the revision code. Its value is 0 for the first sample ICs. | REVISION_ID | [7:0] | 0 | R/O |

4.9 Registers of MAC/IO Control Module

* Base Address: 2400H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00H	configurable preamble bytes This register specifies the preamble length (0..7 bytes) for outgoing packets.	PREAM_CFG	[2:0]	7	R/W

01H	**configurable frame gap in di bits for 1st interval** This register specifies the 1st interval of the inter-frame gap (in unit of di bit) for outgoing packets, where the 2nd interval of the inter-frame gap is fixed as 16 di bits (i.e. 32 bits). The minimum inter-frame gate (IFG) defined in 802.3 is 96 bits (48 di bits). But for fast transmission, many manufactures use smaller IFG (minimum is 32 bits) in practical. The allowable IFG_CFG value is 0..63 di bits. Note that, TMAC only performs the carrier sense function during the 1st IFG interval, rather than the whole IFG. So, for the half duplex link, if an incoming packet arrives at the 2nd IFG interval, a collision with the ready-to-send outgoing packet will happen.	IFG_CFG	[5:0]	32	R/W

02H	Backoff configuration bit 0: CAP mode , mild solution for Capture effect bit 1: MBA mode , aggressive solution for Capture effect bit 2: EEFAST mode , drop the 2nd collided packet for testing purpose, accelerate the drop event bit 3: CRANDOM mode , use another random algorithm bit 4: OFSET , parameter for backoff timer For that CAP mode is enabled, the TMAC module will select the backoff time as b'10 or b'11 for the 2nd collision, i.e. the backoff time for the 2nd collision is 2 or 3 slot times, where a slot time is 512 bits time duration. For that MBA mode is enabled, the TMAC will select backoff time for 10th collision just as that for 5th collision backoff time for 11th collision just as that for 4th collision backoff time for 12th collision just as that for 3rd collision backoff time for 13th collision just as that for 2nd collision backoff time for 14th collision just as that for 1st collision backoff time for 15th collision as 0. For that EEFAST mode is enabled, the Output Control will drop the packet immediately as the second collision happens. For that CRANDOM mode is enabled, the TMAC will select the backoff time by using the alternative random algorithm that calculates the backoff time as that for 10th collision. For OFSET=1, the TMAC will follow the 802.3 standard backoff algorithm. For OFSET=0, the TMAC will select the backoff time for the 1st and 2nd collision as that of the 3rd collision, i.e. the possible the backoff time for the 1st and 2nd collision is ranged from 0 to 7 in unit of slot time.	BOFFCFG	[4:0]	5'b10 000	R/ W

03H	**MAC media type configuration**
	bit 0: SPD_10M , value 0: 100Mbps, value 1: 10Mbps
	bit 1: HALF_DPX , 1: half duplex, 0: full duplex
	bit 2: RCV_FC_DIS , 1: disable receive flow control frame
	        0: enable receive flow control frame
	bit 3: XMT_FC_DIS , 1: disable send flow control frame
	        0: enable send flow control frame
		MACCFG	[3:0]	0	R/W

04H	IO port enable bit 0: input port enable , 1: input enable, 0: input disable bit 1: output port enable , 1: output enable, 0: output disable	IO_CFG	[1:0]	0	R/W
10H-13H	received good packet count Accounting Event: receiving packets with CRC ok and packet size between 64 and 1522 (valid maximum packet size in spite of VLAN disabled or enabled). Note that the RMON/MIB counter will be locked during 4-byte continuous register-read, and the increment (if any) is deferred until read complete.	RCV_GOOD_PKT	[31:0]	0	R/O
14H-17H	received bad packet count Formal Definition: "The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol." Accounting Events: (1) receiving valid-length packets with CRC error, (2) receiving runt packets, (3) receiving over-length packets	RCV_BAD_PKT	[31:0]	0	R/O
18H-1BH	drop packet counter Formal Definition: "The total number of events in which packets were dropped by the probe due to lack of resources. Note that this number is not necessarily the number of packets dropped; it is just the number of times this condition has been detected." Accounting Event: input FIFO overrun due to SDRAM-bandwidth blocking or buffer starvation.	DROP_PKT	[31:0]	0	R/O
1CH-1FH	sent good packet count Accounting Event: store-and-forward transmission success without collision	XMT_GOOD_PKT	[31:0]	0	R/O

20H-23H	sent bad packet counter Formal Definition: "The number of outbound packets that could not be transmitted because of errors." Accounting Event: re-transmission due to collision or output FIFO underrun.	XMT_BAD_PKT	[31:0]	0	R/O
2800H	**MAC & I/O Control Module of Port 1**	as same as Port 0			
2C00H	**MAC & I/O Control Module of Port 2**	as same as Port 0			
3000H	**MAC & I/O Control Module of Port 3**	as same as Port 0			
3400H	**MAC & I/O Control Module of Port 4**	as same as Port 0			
3800H	**MAC & I/O Control Module of Port 5**	as same as Port 0			
3C00H	**MAC & I/O Control Module of Port 6**	as same as Port 0			
4000H	**MAC & I/O Control Module of Port 7**	as same as Port 0			
4400H	**MAC & I/O Control Module of Port 8**	as same as Port 0			
4800H	**MAC & I/O Control Module of Port 9**	as same as Port 0			
4C00H	**MAC & I/O Control Module of Port 10**	as same as Port 0			
5000H	**MAC & I/O Control Module of Port 11**	as same as Port 0			
5400H	**MAC & I/O Control Module of Port 12**	as same as Port 0			
5800H	**MAC & I/O Control Module of Port 13**	as same as Port 0			
5C00H	**MAC & I/O Control Module of Port 14**	as same as Port 0			
6000H	**MAC & I/O Control Module of Port 15**	as same as Port 0			

4.10 Registers of CPU IO Control Module

* Base Address: 6400H

Address (offset)	Function	Register Name	Bits	Default Value	R/W
00H	**CPU packet read byte count register bits [7:0]** CPU can check the incoming packet length VIA the 11-bit register PKT_BYTE_CNT [10:0] before starting to read it.	PKT_BYTE_CNT	[7:0]	0	R/O
01H	**CPU packet read byte count register bits [10:8]**	PKT_BYTE_CNT	[10:8]	0	R/O

02H	**CPU packet read status register** 2'b00: idle or packet read in progress 2'b01: packet received successfully 2'b10: packet received with error (CPU needs to read the same packet again)	RD_PKT_ STATUS	[1:0]	0	R/O
03H	Packet source port ID CPU can check the incoming packet's source port ID VIA the 3-bit register PKT_SRC_PORT before starting to read it. It is useful to the spanning tree algorithm.	PKT_SRC _PORT	[3:0]	0	R/O
04H	**CPU IO port configuration register** bit 0 : input port enable , 1: input enable, 0: input disable bit 1 : output port enable , 1: output enable, 0: output disable	CPUIO_C FG	[1:0]	0	R/ W
10H	**CPU packet write status register** bits [1:0] : packet write status 2'b00: idle or packet write in progress 2'b01: CPU sent packet successfully 2'b10: CPU sent packet unsuccessfully (CPU needs to re-write the packet again) bit 2: CPU Input Control is ready for CPU to write packets (It can be ready only after setting CPUIO_CFG[0] = 1.) 0: not ready (default) 1: ready	WR_PKT_ STATUS	[2:0]	0	R/O

SECTION III ELECTRICAL SPECIFICATIONS
ABSOLUTE MAXIMUM RATINGS

Parameter	Min	Max	Unit
Ambient operating temperature	0	70	^oC
Case temperature	0	100	^oC
Storage temperature	-55	125	^oC
Input voltage	-0.5	5.5	Volts
Output voltage (V_{CC} = 3.1 - 3.6V)	-0.5	V_{CC} + 0.5	Volts

Note: Stress above the conditions listed may cause permanent damage to the device. Functional operation of this device should be restricted to the conditions described under operating conditions.

DC CHARACTERISTICS
TA-0-70^oC, V_{CC}=3.3V+/-5%, GND=0V

Symbol	Parameter	Min	Max	Unit	Condition
V_{IL}	Input low voltage	-0.50	0.8	V	
V_{IH}	Input high voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output low voltage	-	0.45	V	I_{OL}=4.0mA
V_{OH}	Output high voltage	2.4	-	V	I_{OH}=-1.0mA
I_{IL}	Input leakage current	-	+/-10	uA	$0<V_{IN}<V_{CC}$
I_{OZ}	Tristate leakage current	-	+/-20	uA	$0.45<V_{OUT}<V_{CC}$
I_{CC}	Power supply current	-	**TBD**	mA	

AC CHARACTERISTICS
AC timing specifications provided are based on external zero-pf capacitance load. Min/Max cases are based on the following table:

Parameter	Min	Max	Unit
3.3V power (Vcc)	3.135	3.465	V
Temperature	0	95	^oC

- CPU interface IO Timing Characteristics

SYMBOL	DESCRIPTION	MIN	MAX	UNIT
t_{IORH}, t_{IOWH}	IOR/IOW falling to IOR/IOW rising	70	-	ns
t_{IORL}, t_{IOWL}	IOR/IOW rising to IOR/IOW falling	25	-	ns
t_{VAL}	HD valid to IOR/IOW falling	25	-	ns
t_{IOWS}	IOW data setup(write data valid to IOW rising)	20	-	ns
t_{IOWH}	IOW data hold(IOW rising to write data invalid)	10	-	ns
t_{IORS}	IOR data setup(read data valid to IOR rising)	20	-	ns
t_{IORH}	IOR data hold(IOW rising to read data invalid)	5	-	ns

```
IOR hhhhhhhhf l l l l l l l l l l r hhhhhf l l
              ~          t_IORH    ! ~    t_IORL    !
          ~ t_VAL !                ~ t_IORS! ~ t_IORH!
HD z z z z z n ddddddddddddddddo z z z z z
```

CPU read timing diagram

```
IOWhhhhhhhhhf l l l l l l l l l l r hhhhhf l l
              ~          t_IOWH    ! ~    t_IOWL    !
          ~ t_VAL !                ~ t_IOWS! ~ t_IOWH!
HD z z z z z n ddddddddddddddddo z z z z z
```

CPU write timing diagram

- SRAM interface Timing Characteristics

SYMBOL	DESCRIPTION	SETUP	HOLD	MIN	MAX	UNIT
t_{SA}	SA output delay			2	7	Ns
t_{SDS} , T_{SDH}	SD input	2.5	1.5			Ns
t_{SD}	SD output delay			2	7	Ns
t_{SADS}	SADS output delay			2	7	Ns
t_{SCS}	SCS0 output delay			2	7	Ns
t_{SWE}	SWE output delay			2	7	ns

```
SCLK l l r hhhhf l l l l r hhhhf l l l l r hhhf l l l r hhh
          + t_SADS ~
SADS z z z z z n d d d d d d d d d d o z z z z z z z z n d d d d d d d
          + t_SCS ~
SCS  z z z z z n d d d d d d d d d d o z z z z z z z z n d d d d d d d
          + t_SWE ~
SWE  z z z z z n d d d d d d d d d d o z z z z z z z z n d d d d d d d
          + t_SA ~
SA   z z z z n d d d d d d d d d d o z z z z z z z z n d d d d d d d
         V           input cycle       V    turn around cycle  V   output cycle    V
         + t_SDS ~ \ \                                          + t_SD ~
SD   n d d d d o z z z z z z z z z z z z z z z z z z z n d d d d d o
          +  t_SDH  ~
```

- DRAM interface Timing Characteristics

SYMBOL	DESCRIPTION	SETUP	HOLD	MIN	MAX
t_{MA}	MA output delay			2	6.5
t_{MDS} , t_{MDH}	MD input	1.5	2		
t_{MD}	MD output delay			2	6.5
t_{BA}	BA0, BA1 output delay			2	6.5
t_{RAS}	RAS0, RAS1 output delay			2	6.5
t_{CAS}	CAS0, CAS1 output delay			2	6.5
t_{DWE}	DWE0, DWE1 output delay			2	6.5
t_{DCS}	DCS3~0 output delay			2	6

```
DCLK  l l r h h h h f l l l l r h h h h f l l l l r h h h f l l l r h h
          +  t_BA  ~
BA    z z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d
          +  t_RAs  ~
RAS   z z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d d
          +  t_CAS ~
CAS   z z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d d
          +  t_DWE ~
DWE   z z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d
          +  t_DCS ~
DCS   z z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d d
          + t_MA ~
MA    z z z z n d d d d d d d d d o z z z z z z z z n d d d d d d d
         V          input cycle        V    turn around cycle  V   output cycle    V
      + t_MDS ~ \ \                                         + t_MD ~
MD    n d d d d o z z z z z z z z z z z z z z z z z n d d d d d d o
          +  t_NDH  ~
```

- RMII Interface Timing Characteristics

SYMBOL	DESCRIPTION	min	type	max	unit	condition
t_{RC}	RCLK50 cycle time			20	ns	
t_{RXDS}	RXD CRS_DV setup time	4	-	-	ns	to RCLK50 rising edge
t_{RXDH}	RXD CRS_DV hold time	2	-	-	ns	to RCLK50 rising edge
t_{TXD}	TXD TX_EN output delay	3	-	12	ns	to RCLK50 rising edge
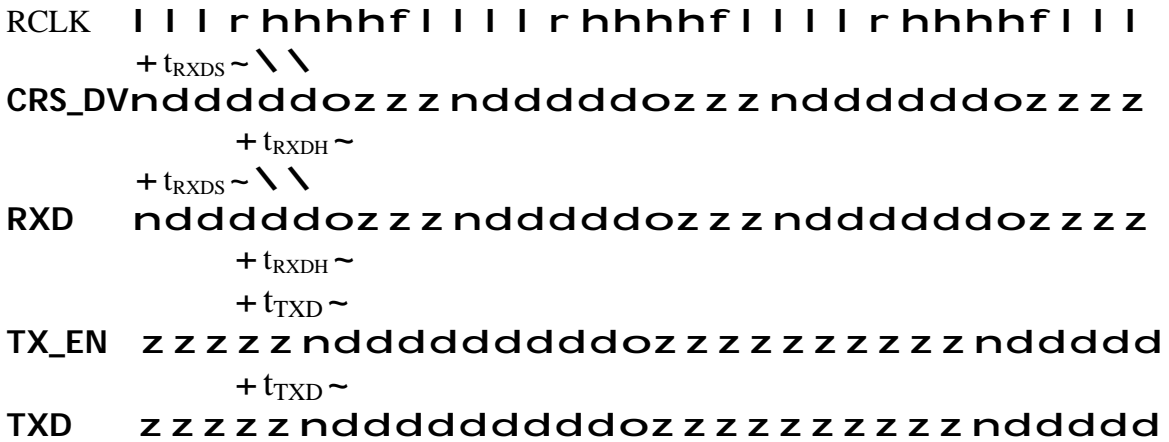
```
RCLK    l l l r hhhhf l l l l r hhhhf l l l l r hhhhf l l l
        + t_RXDS ~ \ \
CRS_DVnddddddoz z z nddddddoz z z nddddddoz z z z
           + t_RXDH ~
        + t_RXDS ~ \ \
RXD     nddddddoz z z nddddddoz z z nddddddoz z z z
           + t_RXDH ~
           + t_TXD ~
TX_EN   z z z z z nddddddddddoz z z z z z z z z ndddddd
           + t_TXD ~
TXD     z z z z z nddddddddddoz z z z z z z z z ndddddd
```

MII Interface Timing Characteristics

SYMBOL	DESCRIPTION	min	type	max	unit	condition
t_{RC}	RCLK cycle time	-	40	-	ns	
t_{RXDS}	RXD, RXDV setup time	5	-	-	ns	to RCLK50 rising edge
t_{RXDH}	RXD, RXDV hold time	5	-	-	ns	to RCLK50 rising edge
t_{TC}	TCLK cycle time	-	40	-	ns	
t_{TXD}	TXD, TX_EN output delay	4	-	20	ns	to RCLK50 rising edge

-

```
RCLK    l l l r hhhhf l l l l r hhhhf l l l l r hhhhf l l l
        + tRXDS ~ \ \
CRS_DVn dddddoz z z nddddddoz z z nddddddoz z z z
            + tRXDH ~
        + tRXDS ~ \ \
RXD     nddddddoz z z nddddddoz z z nddddddoz z z z
            + tRXDH ~


TCLK    l l l r hhhhf l l l l r hhhhf l l l l r hhhhf l l l
            + tTXD ~
TX_EN   z z z z z ndddddddddoz z z z z z z z z ndddddd
            + tTXD ~
TXD     z z z z z ndddddddddoz z z z z z z z z ndddddd
```

- Management Interface (MI) Timing Characteristics

Parameter	min	typ	max	unit	condition
MDC cycle time	-	400	-	ns	
MDC high time	180	200	220	ns	
MDC low time	180	200	220	ns	
MDIO setup time (source by PHY)	30	-	-	ns	to MDC rising edge
MDIO hold time (source by PHY)	0	-	-	ns	to MDC rising edge
MDIO output	200	-	300	ns	to MDC rising

delay (source by vt3061)					edge

- EEPROM Interface Timing Characteristics

Parameter	min	typ	max	unit	condition
EEC clock frequency		0	78.12	-	kHz
Clock high time	-	6.4	-	μs	
Clock low time	-	6.4	-	μs	
Start Condition setup time		6.4	-	-	μs
Start Condition hold time	6.4	-	-	μs	
Stop Condition setup time		6.4	-	-	μs
Stop Condition hold time	6.4	-	-	μs	
Read Data In setup time	0	-	-		to EEC rising edge
Read Data In hold time	0	-	-		to EEC falling edge
EEIO Data out delay	2.6	-	3.0	μs	to EEC falling edge
Write Cycle time	-	11.4	-	ms	

PACKAGE MECHANICAL SPECIFICATIONS

PIN #1 CORNER

Ø 1.00(3X) REF

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

A B C D E F G H J K L M N P R T U V W Y AA AB AC AD AE AF

VT 6516

YYWWRR   TAIWAN
LLLLLLLLL   C   M

30.00 REF

Y = Date Code Year
W = Date Code Week
R = Chip Revision
L = Lot Code

4.00*45º (4X)

30.00 REF

Ø 0.10 Ⓢ C
Ø 0.30 Ⓢ C A Ⓢ B Ⓢ
Ø 0.75±0.15(492X)

26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

A B C D E F G H J K L M N P R T U V W Y AA AB AC AD AE AF

476-Pin BGA

35x35x2.33 mm
JEDEC Spec MO-151

35.00 ±0.20

31.75

1.27

−B−

−A−

31.75

1.27

35.00 ±0.20

⌂ 0.20(4X)

1.17 REF

30º TYP

−C− SEATING PLANE

0.56 REF

0.15 C

0.60 ±0.10
2.33 ±0.13