Product Specification

ADM5106 / ADM5107

Home Gateway Controller

Admtek Incorporated
5F, No.2-1,Industry East 1 Road.
Science-based Industrial Park
Hsinchu, Taiwan, R.O.C.

Content

PRODUCT SPECIFICATION **1**

1. OVERVIEW 5
2. FEATURE 5
3. BLOCK DIAGRAM 7
4. PIN ASSIGNMENT 8
4.1 Pin Diagram *8*
4.2 Pin Description *10*
5. FUNCTION DESCRIPTION 15
5.0 System *15*
5.0.1 Frequency 15
5.0.2 Boot code data-width 15
5.0.3 MII port 15
5.0.4 SDRAM initialization 15
5.1 PHY *16*
5.1.1 PHY Overview 16
5.1.2 Link Detect 16
5.1.3 Auto-Negotiation 16
5.1.4 Digital Adaptive Equalizer 17
5.1.5 Clock Recovery 17
5.1.6 Stream Cipher Scrambler/ De-scrambler 17
5.1.7 Encoder/Decoder 17
5.2 Switch Engine *17*
5.2.1 Hashing Function 17
5.2.2 Learning Process 17
5.2.3 Routing 18
5.2.4 Forwarding 18
5.2.5 Buffer Management 18
5.2.6 Flow Control (Patent Pending) 18
5.2.6.1 Full Duplex 18
5.2.6.2 Half Duplex 18
5.2.7 Packet priority and CoS (class of service) 19
5.2.8 VLAN 19
5.2.9 The address table access 19
5.2.10 The address security 19
5.2.11 The forced routing function 19
5.2.12 The bandwidth control function 20
5.2.13 The DMA function between switch and ARM7 20
5.2.13.1 The from_switch descriptors content 20
5.2.13.2 The to_switch descriptors content 21
7. SYSTEM MEMORY MAP 22
7.1 system memory map table *22*
8. SYS AND INT 23
8.1 SYSC register map *23*
8.2 SYSC registers description *23*
Control, offset: 0x00 23
8.3 Interrupt map *23*
8.4 INTC register map *23*
8.5 INTC registers description *23*
IRQ_status, offset: 0x00 23
IRQ_raw_status, offset: 0x40 24
IRQ_enable, offset: 0x80 24
IRQ_enable_clear, offset: 0xc0 24
IRQ_soft, offset: 0x10 24
IRQ_test_source, offset: 0x14 24
IRQ_source_sel, offset: 0x18 24
9. MEMORY AND UART REGISTER MAP 25
9.1 SMC register map *25*

9.2 SMC registers description *25*
control register of bank0 and 1, offset: 0x00 25
9.3 SDC register map *26*
9.4 SDC registers description *26*
control register0, offset: 0x00........ 26
control register1, offset: 0x04........ 27
refresh timer register, offset: 0x04 27
write buffer time-out register, offset: 0x04 28
9.5 UART register map *28*
9.6 UART registers description *28*
UARTDR, offset: 0x00 28
UARTRSR/UARTECR, offset: 0x04 (write to clear)........ 28
UARTLCR_H, offset: 0x08........ 29
UARTLCR_M, offset: 0x0c 29
UARTLCR_L, offset: 0x10 30
UARTCR, offset: 0x14 30
UARTFR, offset: 0x18........ 30
UARTIIR/UARTICR, offset: 0x1c, write clear........ 31
10. SWITCH REGISTER MAP........ 32
10.1 The control registers and MIB registers map *32*
10.2 MIB counter registers maps *33*
10.3 Registers description *35*
Code, offset: 0x00........ 35
SftReset , offset: 0x04........ 35
MIBReset , offset: 0x08 35
Reserved, offset: 0x0c........ 35
Global_Status, offset: 0x10 35
PHY_Status, offset: 0x14........ 35
Port_Status, offset: 0x18 36
ROMsize, offset: 0x1c 36
SW_config, offset: 0x20 36
CPUp_config, offset: 0x24........ 37
Port_config0, offset: 0x28........ 37
Port_config1, offset: 0x2C........ 37
MII_force, offset: 0x30 37
Pkt_force0, offset: 0x34 38
Pkt_force1, offset: 0x38 38
VLAN_G0, offset: 0x40........ 38
VLAN_G1, offset: 0x44........ 39
Send_trig, offset: 0x48........ 39
Srch_cmd, offset: 0x4C........ 39
Addr_status0, offset: 0x50 39
Addr_status1, offset: 0x54 40
MAC_wt0, offset: 0x58 40
MAC_wt1, offset: 0x5c........ 40
BW_cntl0, offset: 0x60 40
BW_cntl1, offset: 0x64 40
PHY_cntl0, offset: 0x68........ 41
PHY_cntl1, offset: 0x6c 41
reserved, offset: 0x70........ 41
reserved, offset: 0x74........ 41
Port0123_block, offset: 0x78........ 41
Port456_block, offset: 0x7c........ 41
CPU_block, offset: 0x80 41
Priority_cntl, offset: 0x84........ 42
VLAN_priority, offset: 0x88 42
TOS_priority, offset: 0x8c........ 42
TOS_map0, offset: 0x90 42
TOS_map1, offset: 0x94 42
Custom_priority0, offset: 0x98........ 42
Custom_priority1, offset: 0x9c 42
MIB_err_cntl, offset: 0xa0 43
buffer_status, offset: 0xa4 43
reserved, offset: 0xa8 43
reserved, offset: 0xac 43

INT_status, offset: 0xb0 43
INT_mask, offset: 0xb4 44
GPIO_config0, offset: 0xb8 44
GPIO_config1, offset: 0xbc 44
GPIO_config2, offset: 0xc0 44
Watchdog, offset: 0xc4 44
Swap_in, offset: 0xc8 45
Swap_out, offset: 0xcc 45
ToSW_HBaddr, offset: 0xd0 45
ToSW_LBaddr, offset: 0xd4 45
FmSW_HBaddr, offset: 0xd8 45
FmSW_LBaddr, offset: 0xdc 45
ToSW_HWaddr, offset: 0xe0 45
ToSW_LWaddr, offset: 0xe4 45
FmSW_HWaddr, offset: 0xe8 45
FmSW_HWaddr, offset: 0xec 45
Timer_int, offset: 0xf0 46
Timer, offset: 0xf4 46
MIB counter 46
P0RxPkts, offset: 0x100 46
P0RxBytes, offset: 0x104 46
P0RxBCPkt, offset: 0x108 46
P0RxMCPkt, offset: 0x10c 46
P0RxErr, offset: 0x110 46
P0TxPkts, offset: 0x114 46
P0TxBytes, offset: 0x118 46
P0TxCol, offset: 0x11c 47
ADM5106 / 5107 PACKAGE 50

1. Overview

The ADM5106/5107 is an ARM7-based home gateway controller integrated with 7-port switch, 5-port 10/100BaseT/TX PHY, and peripheral interface such as USB, SDRAM and flash memory. It provides a low cost solution to build a home gateway product allowing the home users to connect to the internet via an Ethernet port (or USB) through any xDSL modem or cable modem.

All the configured LAN ports are functioned as the Layer2 switch --- address routing/learning, packet forwarding/filtering, packet prioritizing, flow control, and all IEEE802.3 MAC specification.

There are five embedded DSP based 10BASE-T/100BASE-TX transceivers. An auto-negotiation function determines the selection of either the 10BASE-T or 100BASE-TX mode. The device fully complies with the IEEE 802.3 specification that determines the requisite registers, coding and waveforms to ensure proper transmitter and receiver operation.

The patent architecture makes the application be flexible – configure the LAN or WAN ports in any port or even multiple ports and high efficiency DMA architecture provides easy programmed interface and high throughput.

The ADM5106/5107 is implemented in a .25u CMOS technology. This technology provides a lower power device. It is featured in a 208 PQFP package.

2. Feature

- ADM5106: 6 ports (5xTP + 1xMII) + USB
- ADM5107: 7 ports (5xTP + 2xMII)
- Embedded RISC CPU, ARM7TDMI
- Architecture (patent pending)
- Embedded switch engine
  - Store-and- forward operation
  - Non-blocking, full line speed capability
  - Full-duplex (IEEE802.3x) and half-duplex flow control (Back pressure)
  - Two priorities CoS, set by port, VLAN tag, TCPIP TOS, and customized type
  - 7 port grouping VLAN (overlap-able)
  - MIB counters
  - Sharing dynamic data buffer management, embedded SSRAM
  - MAC look-up table read/write-able
  - MAC address level security
  - Bandwidth control
  - Hashing scheme: direct-map or XOR
- Embedded 10/100BaseT/TX PHY
  - DSP based design
  - 10BASE-T/100BASE-TX IEEE 802.3u Compliant
  - Digital Decision Feedback Equalizer
  - Integrated Baseline Wander Correction
  - Per port 3 LEDs provided
- Flash interface (512Kx8 , 512Kx16 up to 1Mx16)
- SDRAM interface (1Mx16x2 up to 4Mx16x2)

- 2 UART interface, one for the modem interface
- Embedded USB 1.1 client interface (1.1 PHY embedded)
- GPIO
- LAN / WAN bandwidth prioritized
- Single 25MHz crystal input for network
- 2.5/3.3V, 0.25u CMOS process, 208-pin PQFP

3. Block Diagram

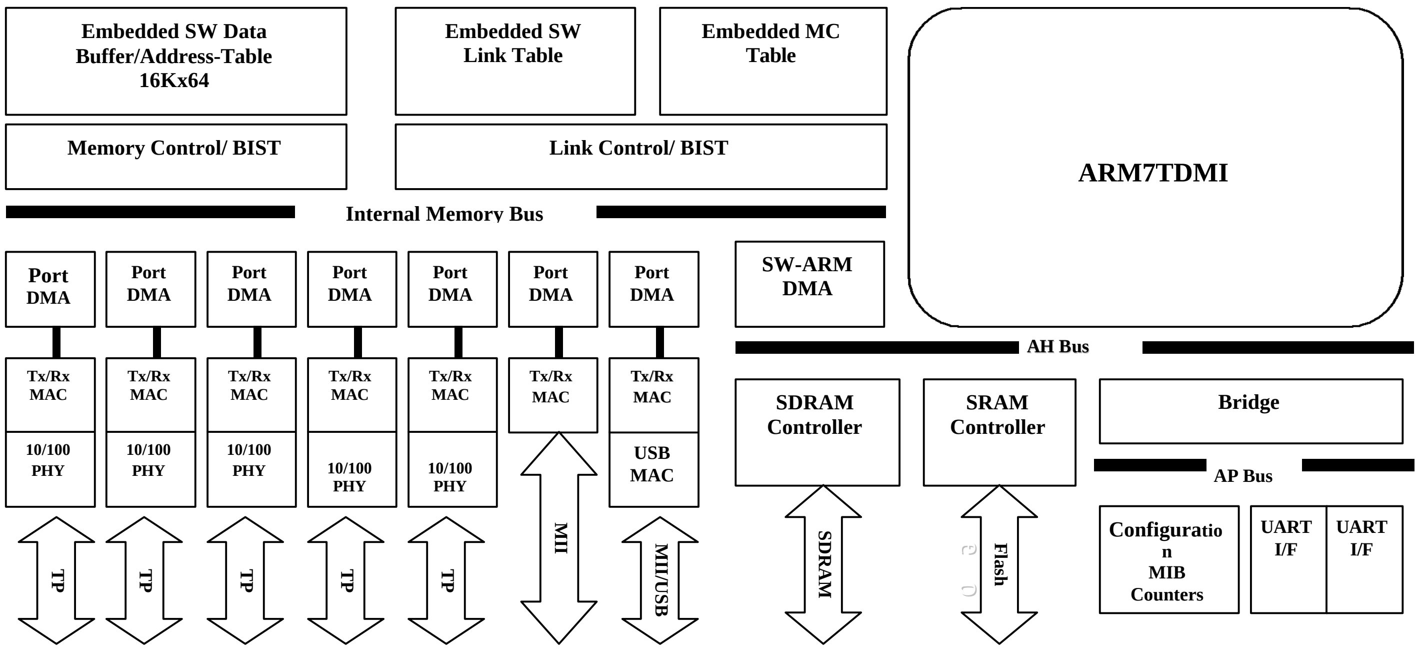
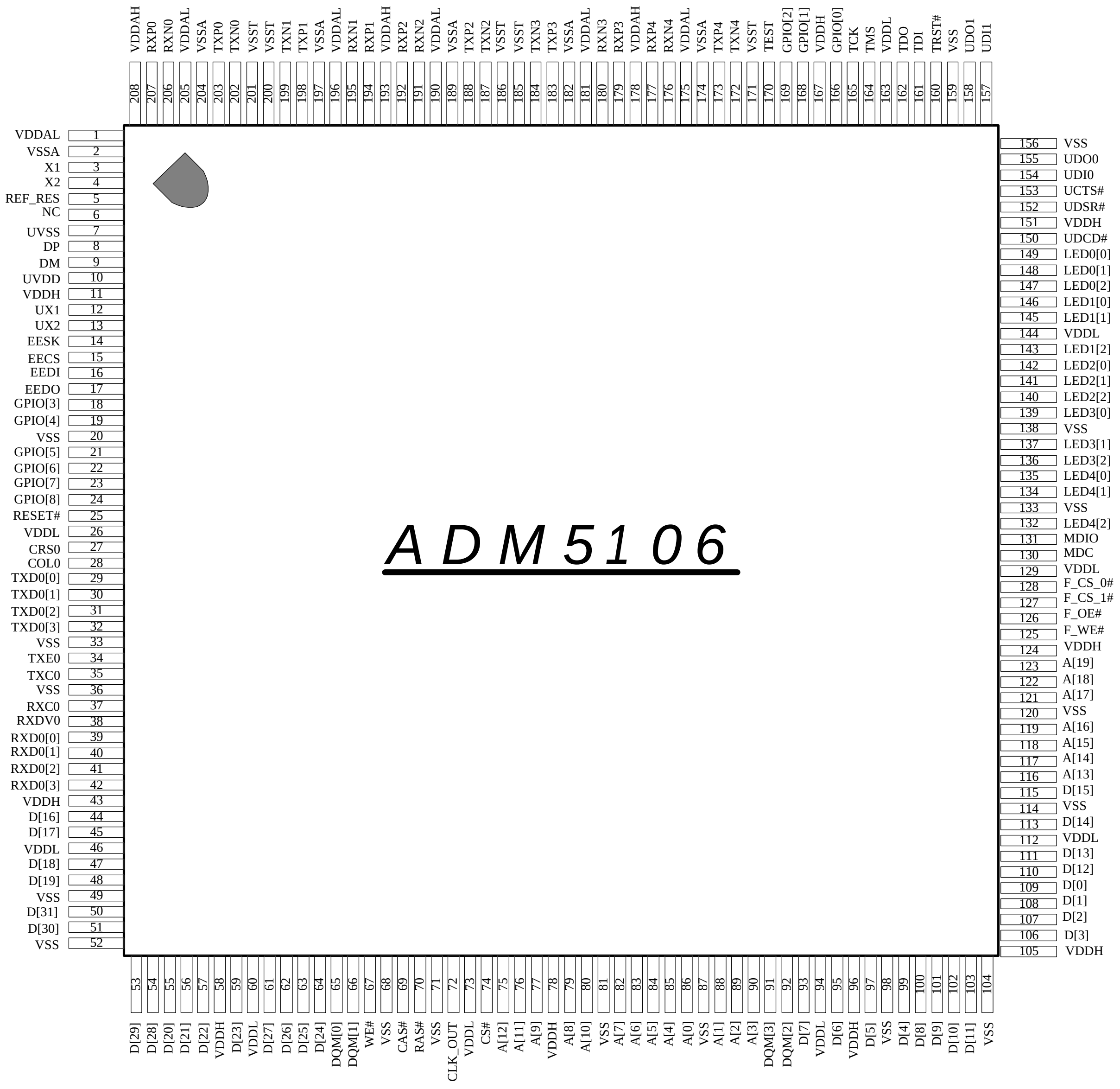


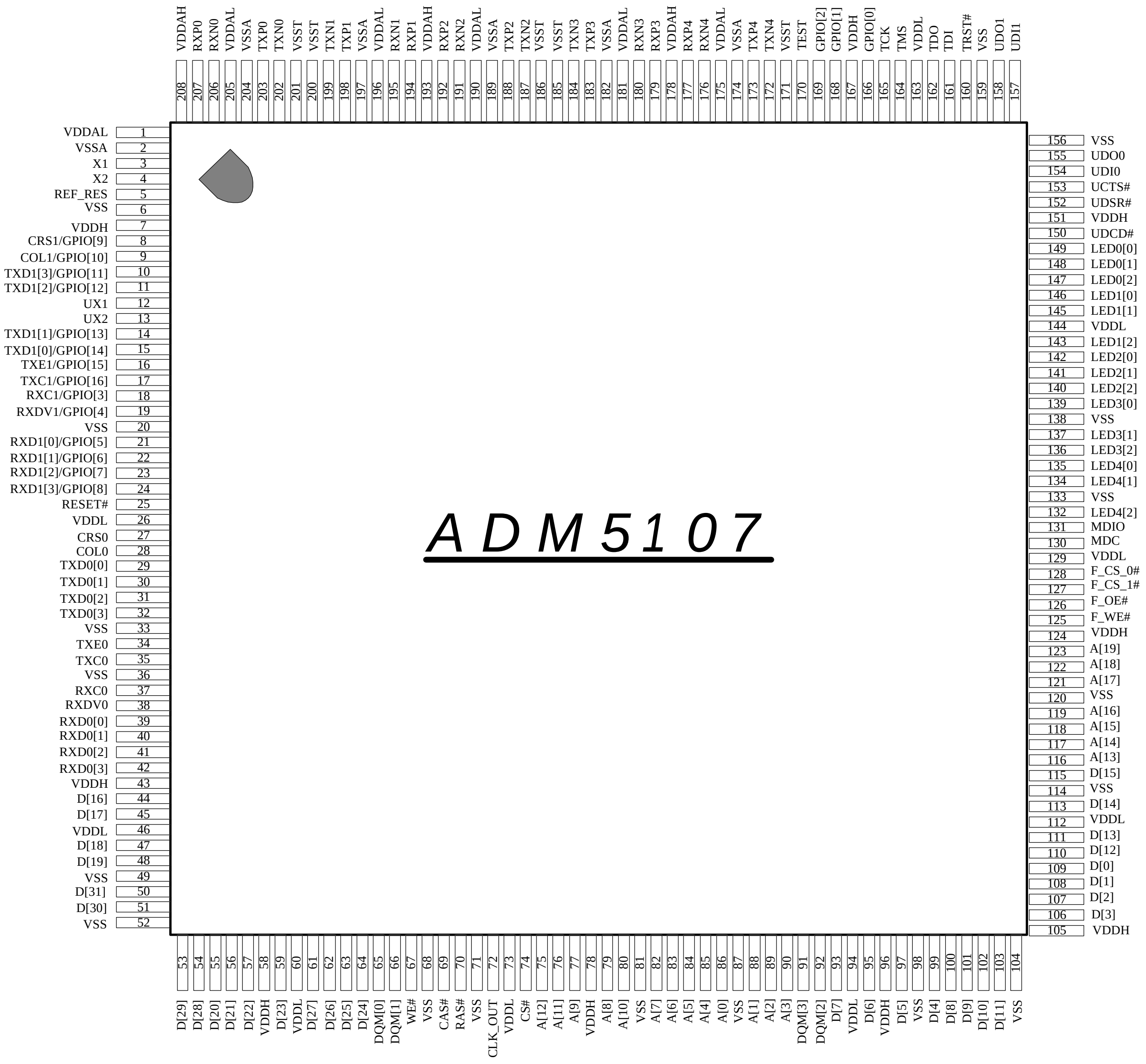
Figure 3.1 ADM5106/5107 Hardware Architecture

4. Pin Assignment

4.1 Pin Diagram



ADM5106 Pin Assignment



ADM5107 Pin Assignment

4.2 Pin Description

Model	Pin Name	Pin #	Type	Descriptions	
Network Media Connections port 4 to port0					
Both	RXP[4:0] RXN[4:0]	177, 179, 192,194, 207 176, 180, 191, 195, 206	I	Receive Pair. Differential data is received on this pin.	
Both	TXP[4:0] TXN[4:0]	173, 183, 188, 198, 203 172, 184, 187, 199, 202	O	Transmit Pair. Differential data is transmitted on this pin.	
Clock for Network					
Both	X1	3	I	25 MHz crystal, external clock input	
Both	X2	4	O	25 MHz crystal	
Both	REF_RES	5	I	Reference Voltage	
LED for port4 to port0 (TP ports only)					
Both	LED4[2:0]	132, 134, 135	O	Mode (A[10:9])	LED[2:0]
				00	LED[2]: 0 = Full_Duplex/Col(Steady/Flash). LED[1]: 0 = Speed_100M (Steady). LED[0]: 0 = Link/activity (Steady/Flash).
				01	LED[2]: 0 = Full_Duplex (Steady). LED[1]: 0 = Speed_100M (Steady). LED[0]: 0 = Link/activity (Steady/Flash).
Both	LED3[2:0]	136, 137, 139	O		
Both	LED2[2:0]	140, 141, 142	O		
				10	LED[2]: 0 = Full_Duplex/Col (Steady/Flash). LED[1]: 0 = Link/Activity (Steady/Flash) if in 100M mode. LED[0]: 0 = Link/Activity(Steady/Flash) if in the10M mode.
Both	LED1[2:0]	143, 145, 146	O		
				11	LED[2]: 0 = col (Flash). LED[1]: 0 = Speed_100M(Steady). LED[0]: 0 = Link/Activity (Steady/Flash).
Both	LED0[2:0]	147, 148, 149	O		
MII Management for MII_0, and MII_1 port. All can be floated if no MII need.					
Both	MDC	130	O	Clock input MDIO. It runs at a 1MHz frequency clock for MII port auto-negotiation result monitoring.	
Both	MDIO	131	I/O	Bi-directional serial pin used to write and read form the registers of the device.	
MII_0 Interface, if no need, let all pins be floated					
Both	CRS0	27	I	Carrier Sense	
Both	COL0	28	I	Collision	

Both	TXD0[3:0]	29, 30, 31, 32	O	Transmit Data, all internal pull down. (1)TXD0[0]: when auto-negotiation is off, and TXD0[0] be pulled down, It means force flow control in MII_0. (2) TXD0[1]: when auto-negotiation is off, and TXD0[1] be pulled down, It means force 100M speed in MII_0. (3) TXD0[2]: when auto-negotiation is off, and TXD0[2] be pulled down, It means force full duplex in MII_0. (4) TXD0[3]: when TXD0[2] be pulled down, It means Nway monitor is on in MII_0.
Both	TXE0	34	O	Transmit Enable, (1)If be pulled up, then MII_0 will be reversed MII interface. (2)If be pulled down, then MII_0 will be normal MII interface. (default)
Both	TXC0	35	I	Transmit Clock
Both	RXC0	37	I	Receive Clock
Both	RXDV0	38	I	Receive Data Valid
Both	RXD0[3:0]	42, 41, 40, 39	I	Receive Data
MII_1 Interface for (5107 and A[16] pull down) only				
5107	CRS1	8	I	Carrier Sense
5107	COL1	9	I	Collision
5107	TXD1[3:0]	10, 11, 14, 15	O	Transmit Data, all internal pull down. (1)TXD1[0]: when auto-negotiation is off, and TXD0[0] be pulled down, It means force flow control in MII_1. (2) TXD1[1]: when auto-negotiation is off, and TXD0[1] be pulled down, It means force 100M speed in MII_1. (3) TXD1[2]: when auto-negotiation is off, and TXD0[2] be pulled down, It means force full duplex in MII_1. (4) TXD1[3]: when TXD0[2] be pulled down, It means Nway monitor is on in MII_1.
5107	TXE1	16	O	Transmit Enable (1)If be pulled up, then MII_1 will be reversed MII interface. (2)If be pulled down, then MII_1 will be normal MII interface. (default)
5107	TXC1	17	I	Transmit Clock
5107	RXC1	18	I	Receive Clock
5107	RXDV1	19	I	Receive Data Valid
5107	RXD1[3:0]	24, 23, 22, 21	I	Receive Data
Memory Bus				
Both	D[31:0]	50, 51, 53, 54, 61, 62, 63, 64, 59, 57, 56, 55, 48, 47, 45, 44, 115, 113, 111, 110, 103, 102, 101, 100, 93, 95, 97, 99, 106, 107, 108, 109	BI	Data bus for SDRAM, flash memory, and external device

Both	A[19:0]	123, 122, 121, 119, 118, 117, 116, 75, 76, 80, 77, 79, 82, 83, 84, 85, 90, 89, 88, 86	O	Address bus for SDRAM, flash memory, and external device. (1) A[19]: Internal pull down. Pull down = Little Endian. (default) (2) A[18:17]: Internal pull down. Can be pulled up and down as following: 00 : boot in 8-bit (Flash memory) (default) 01 : boot in 16-bit 10 : boot in 32-bit. (3) A[16]: Internal pull down. Pull down = the signals of MII_1 will be as MII interface. (default) Pull up = the signals of MII_1 will be as GPIO. (ADM5107 only) (4) A[15]: Pull up = enable 802.3x flow control. (default) (5) A[14:13]: Can be pulled up and down as following: 00 : disable back-pressure. 10 : all jam back-pressure. (default) (6) A[12]: Internal pull down. Pull down = will not abort packet when meet 16 consecutive collisions. (default) (7) A[10:9]: Internal pull down. Can be pulled up and down as following: 00: LED[2:0] = duplex/col (Steady/Flash), speed (Steady), link/activity (Steady/Flash), (default). (8) A[6:3]: Can be pulled up and down as following: 1000: 62.5M 1001: 68.75M 1010: 75M (default) 1011: 81.25M. (9) Note: If "2Mx32" SDRAM is used, then A12, A13 will be the bank select (BS0, BS1).
SDRAM control signals				
Both	CLK_OUT	72	O	SDRAM clock, the frequency is set by A[6:3] A[6:3] = 1000: 62.5M 1001: 68.75M 1010: 75M (default) 1011: 81.25M Note: 1=pull up, 0=pull dwon
Both	CS#	74	O	SDRAM chip select
Both	RAS#	70	O	Raw address strobe, active low
Both	CAS#	69	O	Column address strobe, active low
Both	WE#	67	O	Write enable, active low
Both	DQM[3:0]	91, 92, 66, 65	O	Data mask output to SDRAM
SRAM control signals				
Both	F_CS0#	128	O	Chip select for external memory, like flash, bank0, active low
Both	F_CS1#	127	O	Chip select for external memory, like flash, bank1, active low
Both	F_OE#	126	O	Output enable for external memory banks, active low
Both	F_WE#	125	O	Write enable for external memory banks, active low

UART				
Both	UDI0	154	I	UART0 receive serial data input
Both	UDO0	155	O	UART0 transmit serial data output
Both	UDCD#	150	I	UART0 Data carrier detect (modem status input), active low
Both	UDSR#	152	I	UART0 Data set ready (modem status input), active low
Both	UCTS#	153	I	UART0 clear to send (modem status input), active low
Both	UDI1	157	I	UART1 receive serial data input
Both	UDO1	158	O	UART1 transmit serial data output
JTAG				
Both	TCK	165	I	JTAG test clock
Both	TMS	164	I	JTAG test mode select
Both	TDO	162	O	JTAG test data out
Both	TDI	161	I	JTAG test data in
Both	TRST#	160	I	JTAG asynchronous reset (active low)
USB				
5106	DM	9	BI	USB data minus pin
5106	DP	8	BI	USB data plus pin
5106	EESK	14	O	EEPROM clock
5106	EECS	15	O	EEPROM chip select
5106	EEDI	16	O	EEPROM data in
5106	EEDO	17	I	EEPROM data out
5106	UX1	12	I	USB 48MHx crystal input (for USB and UART)
5106	UX2	13	O	USB 48MHx crystal output (for USB and UART)
GPIO				
Both	GPIO[2:0]	169, 168, 166	BI	General purpose IO pin
5106 or 5107 & A[16] pullup	GPIO[8:3]	24, 23, 22, 21, 19, 18	BI	General purpose IO pin or active
5107 & A[16] pullup	GPIO[10:16]	9, 10,11, 14, 15, 16, 17	BI	General purpose IO pin or active
external CS/INT/wait				
Both	WAIT#	166	O	**WAIT#** is available in en_csx_intx and en_wait enable. register GPIO_config2, bit[4] and bit[6] when CSX active and SMC programmable wait_state time-out, then check the **WAIT#** if high, then complete the access if low, then wait until **WAIT#** go high
Both	CSX#	168	O	external chip select, active low, available if en_csx_intx enable in the .register GPIO_config2, bit[4]
Both	INTX	169	I	external interrupt input, active high, available if en_csx_intx enable in the register GPIO_config2, bit[4]

5106 or 5107 & A[16] pullup	CSX_1#	18	O	external chip select 1, active low, available if en_csx_intx_1 enable in the register in the GPIO_config2, bit[5]
	INTX_1	19	I	external interrupt input 1, active high, available if en_csx_intx_1 enable in the register GPIO_config2, bit[5]
Miscellaneous				
Both	NC	170		No connection
Both	RESET#	25	BI	System reset, active low
Power				
Both	VDDL	26, 46, 60, 73, 94, 112, 129, 144, 163		Positive Power for Digital Core,2.5V
Both	VDDH	43, 58, 78, 96, 105, 124, 151, 167		Positive Power for I/O, 3.3V
Both	VDDAL	1, 175, 181, 190, 196, 205		Positive Power for Analog circuitry, 2.5V
Both	VDDAH	178, 193, 208		Positive Power for Analog circuitry, 3.3V
Both	VSS	6, 20, 33, 36, 49, 52, 68, 71, 81, 87, 98, 104, 114, 120, 133, 138, 156, 159,		GND for Digital circuit, core and I/O
Both	VSSA	2, 174, 182, 189, 197, 204		GND for Analog circuitry
Both	VSST	171, 185, 186, 200, 201		GND for Transmit Analog circuitry
5107	**VDDH**	**7**		**Positive Power for I/O, 3.3V**
5106	**VDDH**	**11**		**Positive Power for I/O, 3.3V**
5106	**VSSA**	**7**		**GND for USB Analog circuitry**
5106	**VDDAL**	**10**		**Positive Power for USB Analog circuitry, 2.5V**

5. Function Description

5.0 System

5.0.1 Frequency

The system clock frequency of ADM5106/5107 is programmable as follows

1000: 62.5M
1001: 68.75M
<u>1010: 75M (default)</u>
1011: 81.25M

The number is set by the pull up or pull down of A[6:3].
This clock is for the ARM and the major switch core, and SDRAM access.

5.0.2 Boot code data-width

The data width of flash or ROM is set by A[18:17] pull up or down

00: boot in 8-bit
01: boot in 16-bit
10: boot in 32-bit

5.0.3 MII port

The MII port can be programmed, AN monitor on/off, force speed/duplex/flow-control, set by the TXD[3:0] pull up or down (default is pull down).
1xxx: AN monitor OFF
11xx: AN OFF and at the half duplex
1x1x: AN OFF and at 10M speed
1xx1: AN OFF and flow-control off
The MII direction is also programmable – connect to PHY or MAC.
The default is MII mode, connected to PHY. If the TXE is pulled up, then is programmed to connect to MAC.
The signals direction will changed, and suggest the connection as below

5106/7	RXC → RXC (MAC side)
	TXC → TXC
	TXD → RXD
	TXE → RXDV
	RXD ← TXD
	RXDV ← TXE
	COL → COL
	CRS → CRS

5.0.4 SDRAM initialization

After power on & reset, software must initialize the SDRAM controller.
An example sequence is given below:

1. Wait 100us to allow SDRAMs power and clocks to stabilize.
2. Set the I and M bits. This automatically issues a NOP to the SDRAMs.
   Note:
   The M and I bits are in control register1.
3. Waits 200us
4. Reset the M bit (I=1, M=0). This automatically issues a Pre-Charge-All to the SDRAMs.
5. Write 10 to the refresh timer register. This provides a refresh cycle every 10 clock cycles
6. Wait for a time period equivalent to 80 clock cycles (8 refresh cycles).

7. Program the operational value to the refresh timer.
8. Select command write mode (I=0, M=1) and perform a read operation at address
   " SDRAM_base_address + X " for SDRAM,
   which "X" is the "mode register data" shifted left by 12 bits.
9. Program configuration register0.
   Note:
   Parameters programmed into the SDRAMs, such as burst length, RAS and CAS delays, must be consistent with the values written to control register0.
10. Clear the M and I bits and set the other bits in control register1 to their normal operational values.
11. The SDRAM is now ready for normal operation.

5.1 PHY

5.1.1 PHY Overview

The ADM5106/5107 is an embedded 5 ports Ethernet PHY (physical layer) device. It is capable of operating at either 10Mbps or 100Mbps. The PHY is associated with the Physical Layer of the OSI model. The PHY performs functions between the MDI (Medium dependent interface) and the internal MAC of switch. According to the IEEE 802.3u, the PHY contains the PCS (physical coding sublayer), PMA(physical medium attachment), PMD(physical medium attachment), and the optional Auto- Negotiation functions.

In the 100BASE-TX mode, the ADM6509 takes the data from the MACs and performs the physical layer 4B/5B encoding, scrambling, parallel to serial conversion, NRZ to NRZI conversion and NRZI to MLT-3 conversion for transmission over the UTP CAT 5 cable.

The receive path uses an adaptive equalizer to take MLT-3 signals from the cable, performs clock recovery, perform a MLT-3 to NRZI conversion, a NRZI to NRZ conversion, de-scrambling the data, 4B/5B decoding, performs data alignment, stores the data in an elasticity buffer and then converts to nibble data that is applied to the MACs. Auto-Negotiation, carrier sense, collision detection is implemented in the device logic.

Similarly, in the 10BASE-T mode, Manchester encoding and decoding is used with two level transmitted and received data on CAT 3 cable.

5.1.2 Link Detect

The 10Base-T and the 100Base-TX has different means of signaling link integrity. The link detect function uses a counter to count the number of edges in the receive signal over a given period. When enough edges are counted on three consecutive periods, link detect is established. The number of edges will determine whether the link is 10Base-T or 100Base-TX.

5.1.3 Auto-Negotiation

An Auto-Negotiation(AN) mechanism as defined in the IEEE 802.3u is implemented. When AN function is enabled, the mode of operation will be automatically chosen by advertising its abilities and comparing them with those received from its link partner. Each transceiver port can advertise 100Base-Tx full duplex, 100Base-T half duplex, and 10Base-T full or half duplex. Each transceiver will negotiate independently with its link partner and choose the highest level of operation available for its own link.
The enable/disable/force can be set by CPU via the registers.

5.1.4 Digital Adaptive Equalizer

100Base-TX transmission through the transmission media causes signal distortion characterized as wideband loss, baseline wander, jitter, frequency errors and intersymbol interference. The adaptive equalizer function conditions the incoming 100Base-T receive signal to compensate for this distortion. Transceivers that utilize an analog methodology to equalize are subject to system noise degrading their performance while a digital methodology provides better noise immunity but with the tradeoff of high power consumption. This design uses an optimal combination of analog and digital techniques resulting in superior performance in signal recovery and with low power consumption. This equalizer uses forward analog equalization and digital decision feedback equalization(DFE). Gain, offset, baseline wander, jitter and frequency error are addressed and are compensated. This technique enables outstanding performance up to 100meters on CAT 5 twisted pair even in noisy environments.
10Base-T transmission utilizes a pre-equalization technique, the adaptive equalizer is bypassed.

5.1.5 Clock Recovery

The clock recovery function is based on the PLL The equalized signal is sampled with a 125 MHz clock with a feedback mechanism to shift the sampling point to the optimum position. This recovered clock is used to synchronize all other functions of the “Receive” section especially data recovery and data transmittal.

5.1.6 Stream Cipher Scrambler/ De-scrambler

To reduce the radiated emissions on the twisted pair cable, a scrambling function is implemented to randomize the data. The TX data is combined with a 11bit wide linear shift register producing a 2047 bit non repeating sequence.
The de-scrambling function requires that the data is synchronized and aligned. The de-scrambler monitors the recovered data for a sequence representing idle codes to achieve synchronization with the transmit shift register. Once the pattern is deciphered, the data can then be retrieved for further processing.

5.1.7 Encoder/Decoder

To reduce the radiated emissions, to ensure bandwidth spread, to ensure sufficient signal transitions for clock recovery, and to provide framing for the 100Base-TX mode, the IEEE 802.3 provides for a 4B(4bit data nibble) to 5B (5 bit symbol) conversion. As well, the MLT-3 line coding was introduced to limit signal transitions to half peak value per cycle. Thus significantly reducing the FCC emissions above 30 MHz. Thus the signal is encoded for transmission along the line and decoded upon signal recovery.

5.2 Switch Engine

5.2.1 Hashing Function

ADM5106 provides an embedded 1K MAC address look-up table to implement the address recognition. The entry of hashing table is calculated by direct mapping or XOR function to produce a 10-bit hashing address entry.

5.2.2 Learning Process

Address learning process is composed of SA of packets and hashing function. ADM5106/5107 will compare the source address (SA) of each incoming packet:

a. If the source address of incoming packet is the same as the source MAC address table, then the aging status and port number will be updated.
b. If the source address is different from the source MAC address table (mean address collision), then no learning process will occur.

Exception case of address learning

1. the packets have error
2. the port learning is been disable
3. address collision
4. the source address is multicase
5. the packets from CPU

5.2.3 Routing

When a packet comes from portA, ADM5106/5107 will compare its destination MAC address with the MAC address in the MAC address lookup table. If the address is the same and port number is portA means that the packet is a local packet, then discarded. If the address is the same but port numbers are different, the packet is an unicast packet, and will be forwarded to the assigned port. If the incoming packet is a broadcasted one, a multicast one, or an unknown one (i.e. the destination address cannot be found in the MAC address lookup table), then the routing scheme will broadcast it to all ports.

If the MAC address is VLAN address, then the packet will be routed to CPU port. The VLAN address is programmed by CPU, but not from the address learning.

5.2.4 Forwarding

ADM5106/5107 provides store-and-forward method as forwarding scheme. Each outgoing packet, including to-CPU packets, will be stored in the buffer first, and then directly sent to the assigned port or CPU via DMA. However, only the good and non-local packets will be sent.

5.2.5 Buffer Management

The buffer memory (SSRAM) is embedded in ADM5106/5107 for the switch operations, which are designed based on output queuing and dynamic shared memory management architecture. It will assign buffer resources based on traffic status. In addition, this efficiency method can avoid the problem of HOL (Head-on-Line) blocking and cause better transmitting performance.

5.2.6 Flow Control (Patent Pending)

The on/off status for flow control depends on the global empty buffer count and per-port waiting-transmit count. Based on this intelligent scheme, if the packet transmits to the destination port which is full , then the flow control is turned on. And if the situation, the full condition, is released, including packets transmitted out or disable, then the flow control is turned off.

ADM5106/5107 does not do the flow control to CPU, mean – never send the flow control packets to CPU port, so the firmware need monitor the buffer status to prevent the packet lost.

5.2.6.1 Full Duplex

In full duplex flow control, ADM5106/5107 follows IEEE 802.3x standards. If a PAUSE frame is received from a certain port, it will stop the port transmission of packets until the timer is timeout or another PAUSE frame with zero time is received. If the buffer is full and is in full-duplex mode, ADM5106/5107 will send a PAUSE frame with the maximum value, to defer receiving packet. When enough buffer space is released, PAUSE frame with zero delay is sent.

5.2.6.2 Half Duplex

In half duplex operation, ADM5106/5107 supports backpressure features. If free blocks in the buffer memory are below the threshold, a jam packet (jam mode) is sent to the connected segment, regardless of routing decisions. In jam mode, the jam number can be programmed by EEPROM.

5.2.7 Packet priority and CoS (class of service)

ADM5106/5107 can set the packets as high priority as follows via registers:

Port Number
VLAN tag
TCP/IP TOS/DS
Customer defined type

The priority setting by port means that all the packets received by the port will be priority frames; ADM5106/5107 can also judge the priority of frames by checking the specific bits of VLAN tag or TCP/IP TOS/DS in the frame or the customer defined type.
It will determine the packet priority. First it will check if the packet type meets VLAN or TCP/IP. Then, it will check whether the value of VLAN tag or TCP/IP TOS/DS field meets the registers setting. Depending on these two conditions, the scheme of weighted round robin can determine the high and low priority of frames, and thus set the transmitting order.
ADM5106/5107 provides a function to improve the delay-time sensitive traffic in the flow-control condition. When the port receives a priority frame, the back pressure & 802.3x flow control can be turned off until no priority frame occurs within 1 or 2 seconds, then turned back on again (Patent Pending). So it can prevent the jitter caused by the flow control and the better timing-variation result for the priority traffic. This is a register programmable function.
All the packets from CPU port will be treated as the high priority for the switch ports, then it can provide the best effort result for the CPU traffic (mostly they are this bandwidth is charged.)

5.2.8 VLAN

ADM5106/5107 supports 7 port-grouping VLAN. Each of VLAN will be treated as the isolated ports.
ADM5106/5107 provides the VLAN MAC address function, if the packet with the assigned VLAN address as its destination MAC address, then this packet will be forwarded to CPU via DMA.
For example, port0 is the WAN port, and the others are the LAN port, then WAN port set as VLAN1 and others set as VLAN2. And program the different MAC address for the VLAN1/2 into address table. Then if the LAN ports receive packets with VLAN2 address, the packets will be forward to CPU via DMA. After processing the packets (like NAT), CPU can forward the packets to VLAN1.

5.2.9 The address table access

ADM5106/5107 provides the access of the embedded MAC address.

Read – register B+48, 4C, and 50

Issue the search-start command, and ADM5106/5107 will automatically search the embedded address table and report the valid one only.
If end of table, it also report the status.

Write -- register B+58 and 5C

Fill the write address and other information, like port number (or VLAN number), age-time (or static), then issue the write command, and wait for the write done bit.

5.2.10 The address security

ADM5106/5107 support the source MAC address security function, register B+2C.
It can check the all incoming packets in the enable ports – compare the source MAC is existed in the MAC address table or not, if not, discard the packets and report the status, register B+18.

5.2.11 The forced routing function

This is the feature good for the protocol which need CPU help to transfer, like HPNA2.0. the operation is -

---

Program B+34 or B+38

For example, port5 is a HPNA2.0 port. For port0 to 4, all the received packets and destination port is port5 need forward to CPU instead of directly forwarding to port5. After CPU convert the protocol, then forward to port5.

Then B34 bit[6:0] = 0100000, bit[14:8] = 0100000 …

5.2.12 The bandwidth control function

This is the patent feature of ADMtek.
ADM5106/5107 can provide the RX/TX separated bandwidth control (or traffic shapping) function which can be programmed to 64K-bit/128K/256K/512K/1M/4M/10M.
In a fixed period, ADM5106/5107 will count the per port RX and TX byte number, and compare with the bandwidth control threshold. If over this threshold, ADM5106/5107 will turn on the proprietary scheme to control the RX/TX behavior.

5.2.13 The DMA function between switch and ARM7

ADM5106/5107 use the DMA scheme to handle all the traffic between LAN and WAN ports. Basically it use VLAN to identify the LAN and WAN.
For example ➔

LAN ports are VLAN1, and WAN port is VLAN2. (refer to registers B+40, 44)
Setting VLAN MAC address into address table
Setting the BC or MC receiving ability for CPU

The DMA is controlled by the descriptions, to_switch_low, to_switch_high, from_switch_low, and from_switch_high (refer to register B+ D0, D4, D8, DC)

to_switch_low mean the normal priority packets from switch to CPU
to_switch_high mean the high priority packets from switch to CPU
from_switch_low mean the normal priority packets from CPU to switch
from_switch_low mean the high priority packets from CPU to switch

In the initialization, the base address of the 4 descriptors must be set.

The descriptor structure is link and ring together, and the mapped buffer size is fixed in 2K-bytes.
So each packet occupies one descriptor only.
The priority of packet is programmed by the same scheme as switch priority handling.
CPU can decide how to handle priority for the low and high priority from_switch descriptors.
For the to_switch descriptors, the switch will process high priority descriptors until empty.
The interrupt status will report all the condition. (refer to register B+B0)

5.2.13.1 The from_switch descriptors content

Bit	Bit[31]	Bit[30:25]	Bit[24]	Bit[23:0]	remark
Type			control		Controlled by CPU, except bit[31]
function	Own bit	reserved	ring end flag	buffer_address[23:0]	

Bit	Bit[26:16]	Bit[15:12]	Bit[11]	Bit[10]	Bit[9]	Bit[8]			remark
Type	status								Controlled by SW
function	pkt-length[10:0]	Source port	**Forced**	Reserve	Reserve	Reserve			

The own-bit ➔ if 1, mean switch can store the packet into the assigned buffer by buffer address field, bit[23:0]. If 0, mean CPU need process this packet, and return to 1 after process completed.

The ring_end_flag ➔ indicate this descriptor is the last one, the next one will be the base.

The buffer_address ➔ the assigned buffer address(by byte) which can be any bytes allocated, but must be less than 512 bytes. It is pre-programmed by CPU before own-bit change to 1.

The packet_length ➔ the packet length (by byte)

The source_port ➔ the source port of the packet. For example, if 0, mean the packet is from port0.

The forced ➔ the packet is through forced_rout function to CPU.

5.2.13.2 The to_switch descriptors content

Bit	Bit[31]	Bit[28]	Bit[24]	Bit[23:0]	Remark
Type			Control		Controlled by CPU except Own-bit
Function	Own bit	reserved	ring end flag	buffer_address[23:0]	

Bit	Bit[26:16]	**Bit[14:8]**			**Bit[6:0]**	Remark
Type	Control					Controlled by CPU
Function	pkt-length[10:0]	**force desti-port[6:0]**			**To_VLAN[6:0]**	

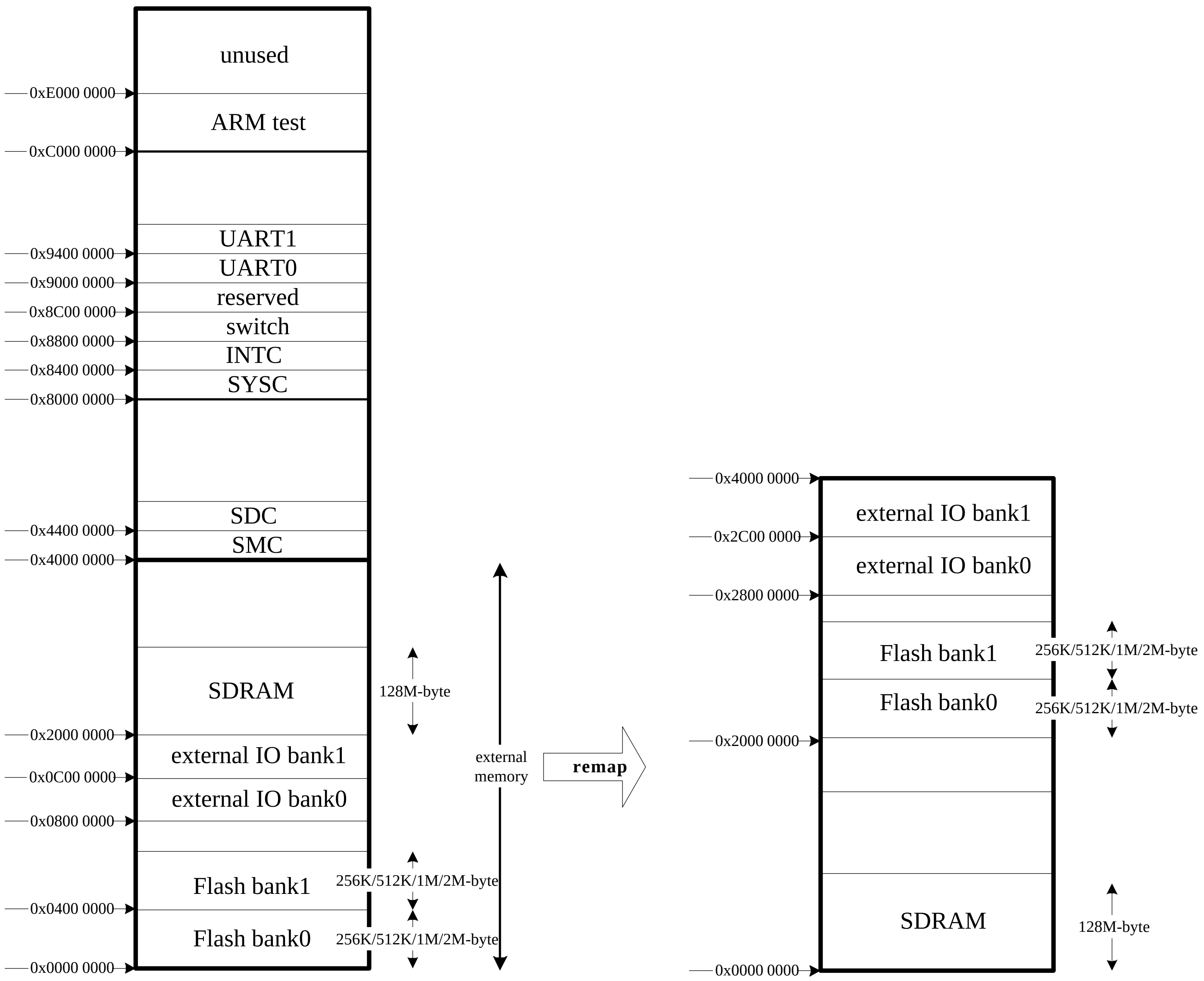
Own_bit, ring_end_flag, buffer_address , and packet length are the same as the to_switch descriptors.

Force_desti-port ➔ CPU wants to force the packet to the assigned port, instead of VLAN and DA routing information.

To_VLAN ➔ CPU indicate the destination VLAN of the packet, then the switch do the routing to decide the port number based on the destination address.

7. System Memory map

7.1 system memory map table



SDC: SDRAM control registers
SMC: Flash control registers
SYSC: System control registers
INTC: interrupt control registers

After power on, the two bank of flash memory are located at the address zero (0x0000_0000), and the SDRAM is located at the address 512M (0x2000_0000). The re-map function is enabled by the SYSC.

8. SYS and INT

8.1 SYSC register map

address	**name**
Base + 00	**Control**

8.2 SYSC registers description

Control, offset: 0x00

Bits	Type	Name	Description	Initial value
4	RW	re-map	1: re-map flash and SDRAM address	0
others		reserved		

8.3 Interrupt map

interrupt	**name**
0	**Timer Interrupt,**
1	**Reserved**
2	**Reserved**
3	**External I/O Bank1 Interrupt**
4	**UART0 Interrupt**
5	**UART1 Interrupt**
6	**External I/O Bank0 Interrupt**
7	**Switch Interrupt**

8.4 INTC register map

address	**name**
Base + 00	**IRQ_status**
Base + 04	**IRQ_raw_status**
Base + 08	**IRQ_enable**
Base + 0c	**IRQ_enable_clear**
Base + 10	**IRQ_soft**
Base + 14	**IRQ_test_source**
Base + 18	**IRQ_source_sel**

8.5 INTC registers description

IRQ_status, offset: 0x00

Bits	Type	Name		Initial value
31:8	RO	reserved		0
7:0	RO	IRQ_status[7:0]	The status of the interrupt sources **after** masking. 1: the corresponding IRQ is active, and generate the interrupt to ARM	0

IRQ_raw_status, offset: 0x04

Bits	Type	Name		Initial value
31:8	RO	reserved		0
7:0	RO	IRQ_raw_status[7:0]	The status of the interrupt sources **before** masking. 1: the corresponding IRQ is active	0

IRQ_enable, offset: 0x08

Bits	Type	Name		Initial value
31:8	RO	reserved		0
7:0	RW	IRQ_enable[7:0]	The enable register is used to mask the interrupt source. 1: enable the interrupt and allow the interrupt request to ARM. Writing “0” has no effect.	0

IRQ_enable_clear, offset: 0x0c

Bits	Type	Name		Initial value
31:8		reserved		
7:0	RW	IRQ_enable_clear[7:0]	The clear bits of the IRQ_enable. Writing “1” clear the corresponding bit of IRQ_enable. Writing “0” has no effect.	0

IRQ_soft, offset: 0x10

Bits	Type	Name		Initial value
31:2		reserved		
1	WO	IRQ_soft[7:0]	software interrupt 1: generate the software interrupt 0: clear software interrupt	0
0	RO	reserved		0

IRQ_test_source, offset: 0x14

Bits	Type	Name		Initial value
31:8	RO	reserved		
7:0	RW	IRQ_test_source[7:0]	the test data for the IRQ_raw_status	0

IRQ_source_sel, offset: 0x18

Bits	Type	Name		Initial value
31:1	RO	reserved		
0	RW	IRQ_source_selection	1: load the IRQ_test_source into IRQ_raw_status	0

9. Memory and UART register map

9.1 SMC register map

address	name
Base + 00	**control register of Flash bank0**
Base + 04	**control register of Flash bank1**
Base + 08	**control register of I/O bank0**
Base + 0C	**control register of I/O bank1**

9.2 SMC registers description

control register of Flash bank0, offset: 0x00,
(also Flash bank1,offset=0x04; I/O bank0,offset=0x08; I/O bank1,offset=0x0C)

Bits	Type	Name	Description	Initial value
31:30	RW	AT	access type 00: no retry 01: reserved 10: retry after every access 11: retry after every four memory access	00
29:28	RW	MW	Flash memory width 00: 8-bit 01: 16-bit 10: 32-bit 11: reserved	00
27	RW	BM	burst mode 0: non burst device 1: burst flash	0
26	RW	WP	write protect 0: SRAM not write protected 1: ROM, burst ROM and write protected SRAM	0
25	RW	WPERR	write protect error status 0: no error 1: write protect error write “1” clear	0
24	RW	BUSERR	bus transfer error status 0: no error 1: bus transfer error write “1” clear	0
23:16	RO	reserved		
15:11	RW	WST2	wait state 2, WST2 is the write access time in the case of SRAM, the burst access time in the case of burst ROM, and do not apply to ROM device. This wait state time is (WST2+1) x t_{HCLK} in the case of SRAM. This wait state time is (WST2) x t_{HCLK} in the case of burst ROM.	11111

10	RW	RBLE	It should be written to 1. (default is 0 after reset). 1: all byte lane strobes nXBLS[3:0] held **low** during system reads from memory	0
9:5	RW	WST1	wait state 1, WST1 is the read access time in the case of SRAM and ROM, ot the initial access time in the case of burst ROM. This wait state time is (WST2+1) x t_{CLK}.	11111
4		reserved		
3:0	RW	IDCY	Idle cycle memory data bus turn around time. This turn around time is (IDCY+1) x t_{CLK}.	1111

t_{CLK}= the clock period of setting, like 75MHz in the default.

9.3 SDC register map

address	**name**
Base + 00	**control register0**
Base + 04	**control register1**
Base + 08	**refresh timer register**
Base + 0c	**write buffer time-out register**

9.4 SDC registers description

control register0, offset: 0x00

Bits	Type	Name	Description	Initial value
31:25		reserved		
24	RW	A	auto pre-charge control for SDRAM access. 1: auto pre-charge	1
23:22	RW	R[1:0]	RAS to CAS latency SDRAM mode 00: reserved 01: RAS to CAS latency=1 10: RAS to CAS latency=2 11: RAS to CAS latency=3	11
21:20	RW	C[1:0]	CAS latency 00: reserved 01: CAS latency=1 01: CAS latency=2 01: CAS latency=3	11
19	RW	X	external bus width 0: external bus width=32 1: external bus width=16	0
18	RW	C	must 1	0
17	RW	E	must 0	0

3	RW	B[0]	Bank select	0
2	RW	T[0]	address multiplexing 1: x8 memory device 0: x16 or x32 memory device	0
1	R/W	F[0]	1 = 256 MBit device. 0 = Not 256 MBit device.	0

control register1, offset: 0x04

Bits	Type	Name	Description	Initial value
31:6, 4		reserved		
5	RO	B	SDRAM engine status 1: busy, **should not write control register0 when SDARM engine is busy**	
3	RW	W	write buffer enable 0: merging write buffer disable 1: merging write buffer enable disabling the write buffer will flush any stored value to the external memory	0
2	RW	R	readt buffer enable 0: read buffer disable 1: read buffer enable	0
1	RW	M	control bit for the memory device initialization	0
0	RW	I	control bit for the memory device initialization M, I = 11, automatically issue NOP to the SDRAM M, I = 01, automatically issue a PALL to the SDRAM M, I = 10, enable SDRAM MODE command M, I = 00, Normal operation	0

refresh timer register, offset: 0x08

Bits	Type	Name	Description	Initial value
31:16		reserved		
15:0	RW	timer	program the number of CLK_OUT that should be counted between SDRAM refresh cycle.	

For example, for the common refresh period of 16us, and the CLK_OUT frequency is 75MHz, the following value should be programmed into it:

$16 \times 10^{-6} \times 75 \times 10^{6} = 1200$

write buffer time-out register, offset: 0x0c

Bits	Type	Name	Description	Initial value
31:16		reserved		
15:0	RW	WB_time_out	this register works with the write buffer enable and set the delay time for a flush of a write buffer. it is clocked by CLK_OUT.	

9.5 UART register map

address	name
Base + 00	**UARTDR, data read or written from the interface**
Base + 04	**UARTRSR/ UARTECR, receive status register / error clear register**
Base + 08	**UARTLCR_H, line control register, high byte**
Base + 0c	**UARTLCR_M, line control register, middle byte**
Base + 10	**UARTLCR_L, line control register, low byte**
Base + 14	**UARTCR, control register**
Base + 18	**UARTFR, flag register**
Base + 1c	**UARTIIR, UARTICR, interrupt identification register, and clear register**
Base + 20	**reserved**
Base + 24	**reserved**
Base + 28	**reserved**
Base + 2c	**reserved**

9.6 UART registers description

UARTDR, offset: 0x00

Bits	Type	Name	Description	Initial value
31:8		reserved		
7:0	RW	data	receive data, If FIFO enable, the data is extracted, the 3-bit status is pushed onto the 11-bit FIFO transmit data, If FIFO enable, the data is pushed into the transmit FIFO	

UARTRSR/UARTECR, offset: 0x04 (write to clear)

Bits	Type	Name	Description	Initial value
31:8		reserved		
3	RW	OE	overrun error. 1: the data is received and FIFO is already full The FIFO contents remain valid since no further data is written whne FIFO is full.	
2	RW	BE	break error 1: a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop)	

			In FIFO mode, the error is associated with the character at the top of FIFO.	
1	RW	PE	parity error 1: it indicates the parity of receive data does not match the parity selected in UARTLCR_H (bit2). In FIFO mode, the error is associated with the character at the top of FIFO.	
0	RW	FE	framing error 1: it indicates that the receive character did not have a valid stop bit. In FIFO mode, the error is associated with the character at the top of FIFO.	

UARTLCR_H, offset: 0x08

Bits	Type	Name	Description	Initial value
31:7		reserved		
6:5	RW	WLEN[1:0]	word length, indicates the number of data bits transmitted or received in a frame 11: 8-bit 10: 7-bit 01: 6-bit 00: 5-bit	00
4	RW	FEN	enable FIFOs 1: transmit and receive FIFO buffer are enabled (FIFO mode). 0: the FIFOs are disabled (character mode), the FIFO become one-byte-deep holding register.	
3	RW	STP2	two stop bits select 1: two stop bits are transmitted at the end of frame. The receive logic does not check for two stop bits being received.	
2	RW	EPS	even parity select 1: even parity generation and checking is performed during the transmission and reception, which check the even number of 1s in data and parity bits. 0: odd parity	
1	RW	PEN	parity enable 1: the parity generation and checking is enabled. 0: disable, and no parity bit added to the data frame.	
0	RW	BRK	send break 1: a low level is continually out put on the UARTTXD output, after completing transmission of the current character. this bit must be asserted for at least one complete frame transmission time in order to generate a break condition.	0

UARTLCR_M, offset: 0x0c

Bits	Type	Name	Description	Initial value

31:8		reserved		
7:0	RW	baud_rate[15:8]	most significant byte of baud rate divisor.	00

UARTLCR_L, offset: 0x10

Bits	Type	Name	Description	Initial value
31:8		reserved		
7:0	RW	baud_rate[7:0]	least significant byte of baud rate divisor.	00

The baud rate divisor is calculated as follows:
Baud rate divisor BAUDDIV = (F_{UX} / (16*baud rate))-1
where F_{UX} is the clock of UX1 and UX2 clock frequency.

UARTCR, offset: 0x14

Bits	Type	Name	Description	Initial value
31:7		reserved		0
6	RW	RTIE	1: the receive timeout interrupt is enabled	0
5	RW	TIE	1: the transmit interrupt is enabled	0
4	RW	RIE	1: the receive interrupt is enabled	0
3	RW	MSIE	1: the modem status interrupt is enabled	0
2:1		reserved		0
0	RW	UARTEN	1: UART is enabled	0

UARTFR, offset: 0x18

Bits	Type	Name	Description	Initial value
31:8		reserved		0
7	RO	TXFE	transmit FIFO empty If FIFO is disabled, the bit is set when the transmit holding register is empty. If FIFO is enabled, the bit is set when the transmit FIFO is empty.	
6	RO	RXFF	receive FIFO full If FIFO is disabled, the bit is set when the receive holding register is full. If FIFO is enabled, the bit is set when the receive FIFO is full.	
5	RO	TXFF	transmit FIFO empty If FIFO is disabled, the bit is set when the transmit holding register is full. If FIFO is enabled, the bit is set when the transmit FIFO is full.	
4	RO	RXFE	receive FIFO full If FIFO is disabled, the bit is set when the receive holding register is empty. If FIFO is enabled, the bit is set when the receive FIFO is empty.	
3	RO	BUSY	1: the UART is bust transmitting data. this bit remains set until the complete byte, including stop-bit.	

2	RO	DCD	data carrier detect 1: when the modem status input is 0.	
1	RO	DSR	data set ready 1: when the modem status input is 0.	
0	RO	CTS	clear to send 1: when the modem status input is 0.	

UARTIIR/UARTICR, offset: 0x1c, write clear

Bits	Type	Name	Description	Initial value
31:4		reserved		0
3	RW	RTIS	receive timeout interrupt status	
2	RW	TIS	transmit interrupt status	
1	RW	RIS	receive interrupt status	
0	RW	MIS	modem interrupt status	

10. Switch register map

10.1 The control registers and MIB registers map

address	name
Base + 00	**Code**
Base + 04	**SftReset**
Base + 08	**MIBReset**
Base + 0C	**reserved**
Base + 10	**Global_status**
Base + 14	**PHY_status**
Base + 18	**Port_status**
Base + 1C	**ROMsize**
Base + 20	**SW_config**
Base + 24	**CPUp_config**
Base + 28	**Port_config0**
Base + 2C	**Port_config1**
Base + 30	**MII_force**
Base + 34	**pkt_force0**
Base + 38	**pkt_force1**
Base + 40	**VLAN_G0**
Base + 44	**VLAN_G1**
Base + 48	**Send_trig**
Base + 4C	**Srch_cmd**
Base + 50	**Addr_status0**
Base + 54	**Addr_status1**
Base + 58	**MAC_wt0**
Base + 5C	**MAC_wt1**
Base + 60	**BW_cntl0**
Base + 64	**BW_cntl1**
Base + 68	**PHY_cntl0**
Base + 6C	**PHY_cntl1**
Base + 70	**Reserved**
Base + 74	**Reserved**
Base + 78	**Port0123_block**
Base + 7C	**Port456_block**
Base + 80	**CPU_block**
Base + 84	**Priority_cntl**
Base + 88	**VLAN_priority**
Base + 8C	**TOS_priority**
Base + 90	**TOS_map0**
Base + 94	**TOS_map1**
Base + 98	**Custom_priority0**
Base + 9C	**Custom_priority1**

Base + A0	**MIB_err_cntl**
Base + A4	**Buffer_status**
Base + A8	**Reserved**
Base + AC	**Reserved**
Base + B0	**INT_status**
Base + B4	**INT_mask**
Base +B8	**GPIO_config0**
Base +BC	**GPIO_config1**
Base + C0	**GPIO_config2**
Base + C4	**Watchdog**
Base + C8	**Swap_in**
Base + CC	**Swap_out**
Base + D0	**ToSW_HBaddr**
Base + D4	**ToSW_LBaddr**
Base + D8	**FmSW_HBaddr**
Base + DC	**FmSW_LBaddr**
Base + E0	**ToSW_HWaddr**
Base + E4	**ToSW_LWaddr**
Base + E8	**FmSW_HWaddr**
Base +EC	**FmSW_LWaddr**
Base +F0	**Timer_int**
Base +F4	**Timer**

10.2 MIB counter registers maps

Base + 100	**P0RxPkts**
Base + 104	**P0RxBytes**
Base + 108	**P0RxBCPkt**
Base + 10C	**P0RxMCPkt**
Base + 110	**P0RxErr**
Base + 114	**P0TxPkt**
Base + 118	**P0TxBytes**
Base + 11C	**P0TxCol**
Base + 120	**P1RxPkts**
Base + 124	**P1RxBytes**
Base + 128	**P1RxBCPkt**
Base + 12C	**P1RxMCPkt**
Base + 130	**P1RxErr**
Base + 134	**P1TxPkt**
Base + 138	**P1TxBytes**
Base + 13C	**P1TxCol**
Base + 140	**P2RxPkts**

Base + 144	**P2RxBytes**
Base + 148	**P2RxBCPkt**
Base + 14C	**P2RxMCPkt**
Base + 150	**P2RxErr**
Base + 154	**P2TxPkt**
Base+ 158	**P2TxBytes**
Base + 15C	**P2TxCol**
Base + 160	**P3RxPkts**
Base + 164	**P3RxBytes**
Base + 168	**P3RxBCPkt**
Base + 16C	**P3RxMCPkt**
Base + 170	**P3RxErr**
Base + 174	**P3TxPkt**
Base + 178	**P3TxBytes**
Base + 17C	**P3TxCol**
Base + 180	**P4RxPkts**
Base + 184	**P4RxBytes**
Base + 188	**P4RxBCPkt**
Base + 18C	**P4RxMCPkt**
Base + 190	**P4RxErr**
Base + 194	**P4TxPkt**
Base + 198	**P4TxBytes**
Base + 19C	**P4TxCol**
Base + 1A0	**P5RxPkts**
Base + 1A4	**P5RxBytes**
Base + 1A8	**P5RxBCPkt**
Base + 1AC	**P5RxMCPkt**
Base + 1B0	**P5RxErr**
Base + 1B4	**P5TxPkt**
Base + 1B8	**P5TxBytes**
Base + 1BC	**P5TxCol**
Base + 1C0	**P6RxPkts**
Base + 1C4	**P6RxBytes**
Base + 1C8	**P6RxBCPkt**
Base + 1CC	**P6RxMCPkt**
Base + 1D0	**P6RxErr**
Base + 1D4	**P6TxPkt**
Base + 1D8	**P6TxBytes**
Base + 1DC	**P6TxCol**

10.3 Registers description

Code, offset: 0x00

Bits	Type	Name	Description	Initial value
15:0	RO	Product code	Product code	If ADM5106 = 0x5106 If ADM5107 = 0x5107
19:16	RO	Revision	Revision code	0

SftReset , offset: 0x04

Bits	Type	Name	Description	Initial value
	RO	**SftReset**	Software reset Whole chip reset after read, and please wait 100ms for PHY initial, and memory BIST	

MIBReset , offset: 0x08

Bits	Type	Name	Description	Initial value
	RO	**MIBReset**	MIB counters reset Reset all MIB counters after read	

Reserved, offset: 0x0c

Global_Status, offset: 0x10

Bits	Type	Name	Description	Initial value
0	RO	D_bist_fail	Embedded buffer BIST result (1M-bit SSRAM) 0: good	
1	RO	L_bist_fail	Embedded link list table memory BIST result 0: good	
2	RO	MC_bist_fail	Embedded multicast table memory BIST result 0: good	
31:3			Reserved	

Note: BIST = Build-In Self Test

PHY_Status, offset: 0x14

Bits	Type	Name	Description	Initial value
4:0	RO	Link	Port4 to port0 (embedded PHY ports) link status 1: link up, 0: link down	
6	RO	MII_fail	MII_1 port fail (ADM5107 only, link down) 1: port fail	
5	RO	MII_fail	MII_0 port fail (link down) 1: port fail	
14:8	RO	Speed	The speed status for all MII_1, MII_0, and PHY-ports. 1: 100M	
22:16	RO	Duplex	The duplex status for all MII_1, MII_0, and PHY-ports. 1: full duplex	
30:24	RO	FC_status	Flow control, 802.3x, status after auto-negotiation	

			1: enable for both, self and link-partner	

Port_Status, offset: 0x18

Bits	Type	Name	Description	Initial value
6:0	RO	Secured_st	Security status, the port has intruder if SA_security is ON 1: intruder recoder	
8:7	RO	TXC_st	Monitor the TXC clock of MII_1 and MII_0 1: too slow TXC	

ROMsize, offset: 0x1c

Bits	Type	Name	Description	Initial value
1:0	RW	ROMsize	The external ROM (Flash memory) size option 00: 256K-byte, 01: 512K-byte, 10: 1M-byte, 11: 2M-byte	00

SW_config, offset: 0x20

Bits	Type	Name	Description	Initial value
3:0	RW	HD_IPG	Half duplex IPG setting Bit3: sign bit Bit[2:0]: clock number (by 4-bit clocks)	0000
7:4	RW	age_tmr	Aging time setting 0000: disable 1xxx: fast age 0001: 300 sec, 0010: 600 sec …. 0111: 38400 sec	0001
9:8	RW	BC_prev	Broadcast prevention 00: disable 01: BC discarded, if 64 BC blocks in queue 10: BC discarded, if 48 BC blocks in queue 11: BC discarded, if 32 BC blocks in queue	00
11:10	RW	Max_len	The maximum packet length 00: 1536, 01: 1518, 10: 1522, 11: reserved	00
12	RW	Dis_colabt	Disable collision-16 packet abort	By pin
14:13	RW	Hash_alg	MAC address hashing algorithm 00: direct mode, using last 10-bit as hask address 01: XOR48 mode 10: XOR32 mode 11:reserved example: MAC address = 00 12 34 56 78 9A	
15	RW	Reserved	Must 1	
19:16	RW	Reserved	Must 1010	
21:20	RW	Reserved	Must 10	
22	RW	Rsrv_MC_filter	1: filter reserved MC packets 0: forward reserved MC packets Notes: reserved MC packet ➔ DA = 01-80-c2-00-00-00 (BPDU) and 01-80-c2-00-00-02 to 01-80-c2-00-00-0f	
27:23		Reserved		

31:28	RW	Req_lat	AHB request latency ➔ the switch AHB-DMA bus request latency, 4: 4 clocks latency between requests	

CPUp_config, offset: 0x24

Bits	Type	Name	Description	Initial value
0	RW	DisCPUport	Disable CPU port 1: disable the switch to CPU port, stop sending packets to CPU and clear all the packets in the switch buffer	1
1	RW	CRC_padding	CRC padding from CPU 0: the packet from CPU with CRC padding 1: switch append the CRC all packets from CPU	0
22:16	RW	DisMC_port	Disable the port forward multicast packets to CPU 1: no send MC from the port0 to port6 to CPU	1
30:24	RW	DisBC_port	Disable the port forward broadcast packets to CPU 1: no send BC from the port0 to port6 to CPU	1

Port_config0, offset: 0x28

Bits	Type	Name	Description	Initial value
6:0	RW	Dis_port	Disable port[6:0] 1: Port[6:0] disable	1
22:16	RW	En_BP	Enable back pressure 1: enable port[6:0] back pressure for the half duplex	111111
30:24	RW	En_FC	Enable pause flow control 0: disable port[6:0] 802.3x flow control for the full duplex mode	By pin

Port_config1, offset: 0x2C

Bits	Type	Name	Description	Initial value
6:0	RW	Dis_learn	Disable source address learning 0: enable port[6:0] source address learning	000000
14:8	RW	Learn_secured	Secured the learned source address of port[6:0] 1: the ports learned addresses are secured that can not be used by other ports	000000
22:16	RW	Port__age_disable	Ports' address aging 1: disable the aging if the address is belonged to the ports	000000
30:24	RW	Address_secured	Source address secured 1: the source address of packets need match with the existed one in the table, otherwise the packets will be discarded	000000

MII_force, offset: 0x30

Bits	Type	Name	Description	Initial value
1:0	W	MII_AN	MII_0 and MII_1 auto-negotiation monitor enable 1: enable	By pins
5:4	W	Force_MII_spd	Force MII_0 and MII_1 speed if AN monitor off	By pins

			0: force in the 10M mode	
7:6	W	Force_MII_dpx	Force MII_0 and MII_1 duplex if AN monitor off 0: force in the half duplex mode	By pins
14:8	W	Force_FC	Force 802.3x pause flow control for port[6:0] 1: force flow control on if the port in the full duplex mode	By pins for MII_0 and MII_1

Note: the bit map of read is not the same as write ➔

Bits	Type	Name
1:0	R	MII_AN
3:2	R	Force_MII_spd
5:4	R	Force_MII_dpx
11:6	R	Force_FC

Pkt_force0, offset: 0x34

Bits	Type	Name	Description	Initial value
6:0	RW	P0_force	Port0 forward the packets to CPU instead of sending to the programmed ports Example: 0000010 → if based on the DA of packets the packet need rout to port1, then it will be forced to CPU instead of to port1 1100000 → if based on the DA of packets the packet need rout to port6 or port5, then it will be forced to CPU instead of to port6 or port5	0000000
14:8	RW	P1_force	Port1 forward the packets to CPU instead of sending to the programmed ports	0000000
22:16	RW	P2_force	Port2 forward the packets to CPU instead of sending to the programmed ports	0000000
30:24	RW	P3_force	Port3 forward the packets to CPU instead of sending to the programmed ports	0000000

Pkt_force1, offset: 0x38

Bits	Type	Name	Description	Initial value
6:0	RW	P4_force	Port4 forward the packets to CPU instead of sending to the programmed ports	0000000
14:8	RW	P5_force	Port5 forward the packets to CPU instead of sending to the programmed ports Port5 = MII_0	0000000
22:16	RW	P6_force	Port6 forward the packets to CPU instead of sending to the programmed ports Port6 = MII_1	0000000

VLAN_G0, offset: 0x40

Bits	Type	Name	Description	Initial value
7:0	RW	VLAN0	The port map in the VLAN0 The 7th bit is CPU port, example : 11111111: all port are in the VLAN0	11111111

15:8	RW	VLAN1	The port map in the VLAN1 The 7th bit is CPU port	00000000
23:16	RW	VLAN2	The port map in the VLAN2 The 7th bit is CPU port	00000000
31:24	RW	VLAN3	The port map in the VLAN3 The 7th bit is CPU port	00000000

VLAN_G1, offset: 0x44

Bits	Type	Name	Description	Initial value
7:0	RW	VLAN4	The port map in the VLAN4 The 7th bit is CPU port	00000000
15:8	RW	VLAN5	The port map in the VLAN5 The 7th bit is CPU port	00000000
23:16	RW	VLAN6	The port map in the VLAN6 The 7th bit is CPU port	00000000

Send_trig, offset: 0x48

Bits	Type	Name	Description	Initial value
0	RW	Send_trig_L	Trigger switch to read the normal priority packets through DMA When the packet is ready in the memory and the descriptor is updated, then CPU write this bit to trigger switch access the data/descriptor through DMA.	0
1	RW	Send_trig_H	Trigger switch to read the high priority packets through DMA	0

Srch_cmd, offset: 0x4C

Bits	Type	Name	Description	Initial value
0	RW	Srch_start	Trigger switch to search address table from the begin address	0, self clear
1	RW	Srch_again	Trigger switch to search address table again from the address of current available one. Must be program after the data_rdy bit is set	0, self clear

Addr_status0, offset: 0x50

Bits	Type	Name	Description	Initial value
0	RO	Data_rdy	The search data is ready in registers, including ADDR_st, ADDR_srch0 and ADDR_srch1	0, read clear
1	RO	Table_end	1: All the valid addresses were reported	0, read clear
7:4	RO	Port_num	The port number, refer to bit[7:4] of ADDR_st1	
10:8	RO	Age_field	000: invalid address 001 to 110: valid address, 001 is the youngest , 110: is oldest and near to be age out 111: static address, never age out, only over-written by CPU	
31:16	RO	MAC_addr0	The searched MAC address[15:0]	

Addr_status1, offset: 0x54

Bits	Type	Name	Description	Initial value
31:0	RO	MAC_addr1	The searched MAC address[47:16]	

MAC_wt0, offset: 0x58

Bits	Type	Name	Description	Initial value
0	RW	WtMAC_cmd	The write command of MAC address table 1: the data is ready to write into table	0, self clear
1	RO	WtMAC_done	1: The process of write is completed	0, read clear
7:4	RW	Wt_port_num	The port number need be written into table The bit[7] is indicated VLAN MAC address Ex.: 1000 → VLAN0 MAC address 1001→ VLAN1 MAC address	
10:8	RW	Wt_age_field	The age field need be written into table. 000: disable the MAC address 001 to 110: valid, but will be age out 111: static address	
31:16	RW	Wt_MAC_addr0	The written MAC address[15:0]	

MAC_wt1, offset: 0x5c

Bits	Type	Name	Description	Initial value
31:0	RW	Wt_MAC_addr1	The written MAC address[47:16]	

BW_cntl0, offset: 0x60

Bits	Type	Name	Description	Initial value
2:0	RW	P0tx_bwcntl	The transmit bandwidth control of port0 000: disable, no bandwidth control 001: 64K-bit per second, limit the transmit bandwidth in 64K-bit per second (count the packet excluding preamble and SFD) 010 to 111: 128K/256K/512K/1M/4M/10M bit per second	000
6:4	RW	P0rx_bwcntl	The receive bandwidth control of port0 000: disable, no bandwidth control 001: 64K-bit per second, limit the transmit bandwidth in 64K-bit per second (count the packet excluding preamble and SFD) 010 to 111: 128K/256K/512K/1M/4M/10M bit per second	000
10:8	RW	P1tx_bwcntl	The transmit bandwidth control of port1	000
14:12	RW	P1rx_bwcntl	The receive bandwidth control of port1	000
18:16	RW	P2tx_bwcntl	The transmit bandwidth control of port2	000
22:20	RW	P2rx_bwcntl	The receive bandwidth control of port2	000
26:24	RW	P3tx_bwcntl	The transmit bandwidth control of port3	000
30:28	RW	P3rx_bwcntl	The receive bandwidth control of port3	000

BW_cntl1, offset: 0x64

Bits	Type	Name	Description	Initial value

2:0	RW	P4tx_bwcntl	The transmit bandwidth control of port4	000
6:4	RW	P4rx_bwcntl	The receive bandwidth control of port4	000
10:8	RW	P5tx_bwcntl	The transmit bandwidth control of port5	000
14:12	RW	P5rx_bwcntl	The receive bandwidth control of port5	000
18:16	RW	P6tx_bwcntl	The transmit bandwidth control of port6	000
22:20	RW	P6rx_bwcntl	The receive bandwidth control of port6	000

PHY_cntl0, offset: 0x68

Bits	Type	Name	Description	Initial value
4:0	RW	PHY_addr	The accessed PHY address	
12:8	RW	PHY_register_addr	The accessed register of the assigned PHY	
13	RW	Wt_cmd	1: the write command	0, self clear
14	RW	Rd_cmd	1: the read command	0, self clear
31:16	RW	Wt_data	The written data to the assigned PHY	

Note: the PHY include the embedded and external PHY, the PHY address of embedded from 0 to 4

PHY_cntl1, offset: 0x6c

Bits	Type	Name	Description	Initial value
0	RO	Wt_done	1: the write process is completed	0, read clear
1	RO	Rd_rdy	1: the read process is completed and data is ready in the bit[31:16], Rd_data	
31:16	RO	Rd_data	The read data of read process	

reserved, offset: 0x70

Bits	Type	Name	Description	Initial value
31:0	RW	reserved	Must be 0x69599989	0x69599989

reserved, offset: 0x74

Bits	Type	Name	Description	Initial value
31:0	RW	reserved	Must be 0x7464b8a8	0x7464b8a8

Port0123_block, offset: 0x78

Bits	Type	Name	Description	Initial value
7:0	RW	Port0_th	The port0 protected output block count	0x0c
15:8	RW	Port1_th	The port1 protected output block count	0x0c
23:16	RW	Port2_th	The port2 protected output block count	0x0c
31:24	RW	Port3_th	The port3 protected output block count	0x0c

Port456_block, offset: 0x7c

Bits	Type	Name	Description	Initial value
7:0	RW	Port4_th	The port4 protected output block count	0x0c
15:8	RW	Port5_th	The port5 protected output block count	0x0c
23:16	RW	Port6_th	The port6 protected output block count	0x0c

CPU_block, offset: 0x80

Bits	Type	Name	Description	Initial value
23:16	RW	CPU_th	The CPU (DMA port) port protected output block	0x18

			count	
31:24	RW	USB_th	The USB port protected output block count	0x0c

Priority_cntl, offset: 0x84

Bits	Type	Name	Description	Initial value
6:0	RW	Port_pri	1: Force all forwarded packet are the high priority of the assigned port[6:0]	0000000
14:8	RW	Off_FC_en	1: Auto-turn-off flow control or back pressure when the assigned port[6:0] receive priority packet	0000000
19:16	RW	HL_robin	The transmit proportion of normal and high-priority packet 0000: unlimited, send priority packet until empty 0001: priority:normal=8:1 0010: priority:normal=16:1 ……	0000
31:20	RW	reserved	must be 0x330	0x330

VLAN_priority, offset: 0x88

Bits	Type	Name	Description	Initial value
6:0	RW	Port_vlan_pri_en	The VLAN priority enable for assigned port[6:0]	000000
10:8	RW	VLAN_th	The threshold of VLAN tag for priority If the tag ¡ Ùthis threshold, then the packet is high priority	100

TOS_priority, offset: 0x8c

Bits	Type	Name	Description	Initial value
6:0	RW	Port_tos_pri_en	The TCP/IP TOS, type of service, priority enable for assigned port[6:0]	000000

TOS_map0, offset: 0x90

Bits	Type	Name	Description	Initial value
31:0	RW	TOS_map0	TOS bit_map[31:0] The TOS bit[5:0] is decoded into the 64-bit map. If the map bit =1, then this packet is priority one.	0

TOS_map1, offset: 0x94

Bits	Type	Name	Description	Initial value
31:0	RW	TOS_map1	TOS bit_map[63:32]	0

Custom_priority0, offset: 0x98

Bits	Type	Name	Description	Initial value
6:0	RW	custom_pri_en	The custom priority enable for assigned port[6:0]	000000
31:16	RW	Custom_type	The type filed of priority packet must match this 16-bit.	0

Custom_priority1, offset: 0x9c

Bits	Type	Name	Description	Initial value
7:0	RW	Custom_define	The custom defined pattern.	0000

15:8	RW	Custom_mask	The mask of custom defined pattern. 1: do not care bit.	0000
19:16	RW	Custom_offset	The pattern offset from SA If VLAN packet, the offset will increase 4 automatically.	0000

MIB_err_cntl, offset: 0xa0

Bits	Type	Name	Description	Initial value
7:0	RW	MIB_rx_err_cnd	MII receive error condition control [7]: preamble + SFD only [6]: preamble only [5]: runt (length <64B & CRC error), excluding preamble + SFD only [4]: short (length <64 & CRC ok) [3]: jabber (length > defined length, & CRC error) [2]: long (length > defined length, & CRC error) [1]: alignment error (64≤length ≥ defined length, and CRC error and with dribble bit) [0]: CRC error (64≤length ≥ defined length, and CRC error and without dribble bit)	0xff

buffer_status, offset: 0xa4

Bits	Type	Name	Description	Initial value
8:0	RO	Empty_cnt	The count of empty block	
23:16	RO	H_droping_port	The (CPU-port, port[6:0]) priority packet queue is full, so the incoming ones will be rejected.	
31:24	RO	L_droping_port	The (CPU-port, port[6:0]) normal packet queue is full, so the incoming ones will be rejected.	

reserved, offset: 0xa8
reserved, offset: 0xac

INT_status, offset: 0xb0

Bits	Type	Name	Description	Initial value
0	RO*	To_SW_H_done	DMA move one high priority packet to switch	
1	RO*	To_SW_L_done	DMA move one normal priority packet to switch	
2	RO*	From_SW_H_done	DMA move one high priority packet to CPU	
3	RO*	From_SW_L_done	DMA move one normal priority packet to CPU	
4	RO*	H_Descriptor_full	the descriptor, “high priority from SW to CPU”, full	
5	RO*	L_Descriptor_ful	the descriptor, “normal priority from SW to CPU”, full	
6	RO*	Port0_que_full	the port0 output queue is full	
7	RO*	Port1_que_full	the port1 output queue is full	
8	RO*	Port2_que_full	the port2 output queue is full	
9	RO*	Port3_que_full	the port3 output queue is full	
10	RO*	Port4_que_full	the port4 output queue is full	
11	RO*	Port5_que_full	the port5 output queue is full	
12	RO*	Port6_que_full	the port6 output queue is full	
13	RO*	CPU_port_que_full	the CPU_port, DMA port, output queue is full	

14	RO*	global_que_full	the global resource is full	
15	RO*	Must_drop	The resource almost empty in the dangerous condition	
16	RO*	BC_storm	accumulated broadcast count over BC_storm setting	
17	RO*	MIB_overflow	the MIB counter overflow	
18	RO*	Port_status_chg	any port change the link status (link-up to/from link-down)	
19	RO*	Intruder	any secured port has intruder packets	
20	RO*	Watchdog_tmr_expired	watch dog timer expired	

Note: "*" = means "Write 1 clear", and the interrupt is deasserted.

INT_mask, offset: 0xb4

Bits	Type	Name	Description	Initial value
20:0	RE	Int_mask	1: mask the interrupt	0

GPIO_config0, offset: 0xb8

Bits	Type	Name	Description	Initial value
7:0	RW	In_out0	1: program GPIO[7:0] as input	11111111
15:8	RW	In_vlaue0	The input value of GPIO[7:0] if they are in the input mode	
23:16	RW	out_en0	GPIO[7:0] output enable if in the output mode	
31:24	RW	out_value0	GPIO[7:0] output value if in the output mode and enable	

GPIO_config1, offset: 0xbc

Bits	Type	Name	Description	Initial value
7:0	RW	In_out1	1: program GPIO[15:10] & GPIO[8] as input	11111111
15:8	RW	In_vlaue1	The input value of GPIO[15:10] & GPIO[8] if they are in the input mode	
23:16	RW	out_en1	GPIO[15:10]& GPIO[8]output enable if in the output mode	
31:24	RW	out_value1	GPIO[15:10] & GPIO[8] output value if in the output mode and enable	

GPIO_config2, offset: 0xc0

Bits	Type	Name	Description	Initial value
0	RW	In_out_16	1: program GPIO[16] as input	11111111
1	RW	In_value_16	The input value of GPIO[16] if they are in the input mode	
2	RW	out_en_16	GPIO[16] output enable if in the output mode	
3	RW	out_value_16	GPIO[16] output value if in the output mode and enable	
4	RW	en_csx_intx	1: Enable CSX, INTX in GPIO[1:2]	0
5	RW	en_csx_intx1	1: Enable CSX1, INTX1 in GPIO[3:4]	0
6	RW	en_wait	1: enable wait control, GPIO[0], for the CSX interface	0
8	RW	MII_1_en	1: Enable MII_1 port	By pin

Watchdog, offset: 0xc4

Bits	Type	Name	Description	Initial value

15:0	RW	Watchdog_tmr	Watchdog timer Around 10sec, mask-able, unit 10ms	0, read clear
31	RW	Watchdog_tmr	Watchdog timer trigger reset 1: reset the whole chip, if watchdog timer expired	

Swap_in, offset: 0xc8

Bits	Type	Name	Description	Initial value
31:0	RW	Swap_in	Swap data input Hardware swap the data	0

Swap_out, offset: 0xcc

Bits	Type	Name	Description	Initial value
31:0	RO	Swap_out	Swap data output Swap_dout[31:0] = { Swap_din[7:0], Swap_din[15:8], Swap_din[23:16], Swap_din[31:24]}	0

ToSW_HBaddr, offset: 0xd0

Bits	Type	Name	Description	Initial value
31:0	RW	ToSW_HBaddr	The descriptor base address of to_SW (priority)	0

ToSW_LBaddr, offset: 0xd4

Bits	Type	Name	Description	Initial value
31:0	RW	ToSW_HBaddr	The descriptor base address of to_SW (normal)	0

FmSW_HBaddr, offset: 0xd8

Bits	Type	Name	Description	Initial value
31:0	RW	FromSW_HBaddr	The descriptor base address of from_SW (priority)	0

FmSW_LBaddr, offset: 0xdc

Bits	Type	Name	Description	Initial value
31:0	RW	FromSW_LBaddr	The descriptor base address of from_SW (normal)	0

Note: the loading base addresses need all four address registers (d0, d4,d8,dc) been written.

ToSW_HWaddr, offset: 0xe0

Bits	Type	Name	Description	Initial value
31:0	RO	ToSW_HBaddr	The descriptor working address of to_SW (priority)	0

ToSW_LWaddr, offset: 0xe4

Bits	Type	Name	Description	Initial value
31:0	RO	ToSW_HBaddr	The descriptor working address of to_SW (normal)	0

FmSW_HWaddr, offset: 0xe8

Bits	Type	Name	Description	Initial value
31:0	RO	FromSW_HBaddr	The descriptor working address of from_SW (priority)	0

FmSW_HWaddr, offset: 0xec

Bits	Type	Name	Description	Initial value

31:0	RO	FromSW_LBaddr	The descriptor working address of from_SW (normal)	0

Timer_int, offset: 0xf0

Bits	Type	Name	Description	Initial value
0	RO*	time_out sataus	Time-out interrupt status	0, write 1 clear
16	RW	time_out_mask	1: mask time-out interrupt	0

Note: "*" = means "Write 1 clear", and the interrupt is deasserted.

Timer, offset: 0xf4

Bits	Type	Name	Description	Initial value
15:0	RW	Timer	Time-out interrupt status	0, write 1 clear
16	RW	Timer_en	Count–down timer, time unit 640ns, auto-reload when count down to 0	0xffff

MIB counter

All counter is 32-bit and wrapped around if counter full.

P0RxPkts, offset: 0x100

Bits	Type	Name	Description	Initial value
31:0	RO	P0RxPkts	The port0 receive packet count	0

P0RxBytes, offset: 0x104

Bits	Type	Name	Description	Initial value
31:0	RO	P0RxBytes	The port0 receive bytes count	0

P0RxBCPkt, offset: 0x108

Bits	Type	Name	Description	Initial value
31:0	RO	P0RxPkts	The port0 receive broadcast packet count	0

P0RxMCPkt, offset: 0x10c

Bits	Type	Name	Description	Initial value
31:0	RO	P0RxPkts	The port0 receive multicast packet count	0

P0RxErr, offset: 0x110

Bits	Type	Name	Description	Initial value
31:0	RO	P0RxPkts	The port0 receive error count	0

Note: the error condition can be programmable, register B+A0.

P0TxPkts, offset: 0x114

Bits	Type	Name	Description	Initial value
31:0	RO	P0TxPkts	The port0 transmit packet count	0

P0TxBytes, offset: 0x118

Bits	Type	Name	Description	Initial value
31:0	RO	P0TxBytes	The port0 transmit bytes count	0

P0TxCol, offset: 0x11c

Bits	Type	Name	Description	Initial value
31:0	RO	P0TxCol	The port0 transmit collision count	0

Absolute Maximum Ratings

Supply Voltage(Vcc)	-0.5 V to 2.7 V
Input Voltage	-0.5 V to VCC + 0.3 V
Output Voltage	-0.5 V to VCC + 0.3 V
Storage Temperature	-65 °C to 150 °C(-85°F to 302°F)
Ambient Temperature	0°C to 70°C(32°F to 158°F)
ESD Protection	2000V

DC Specifications

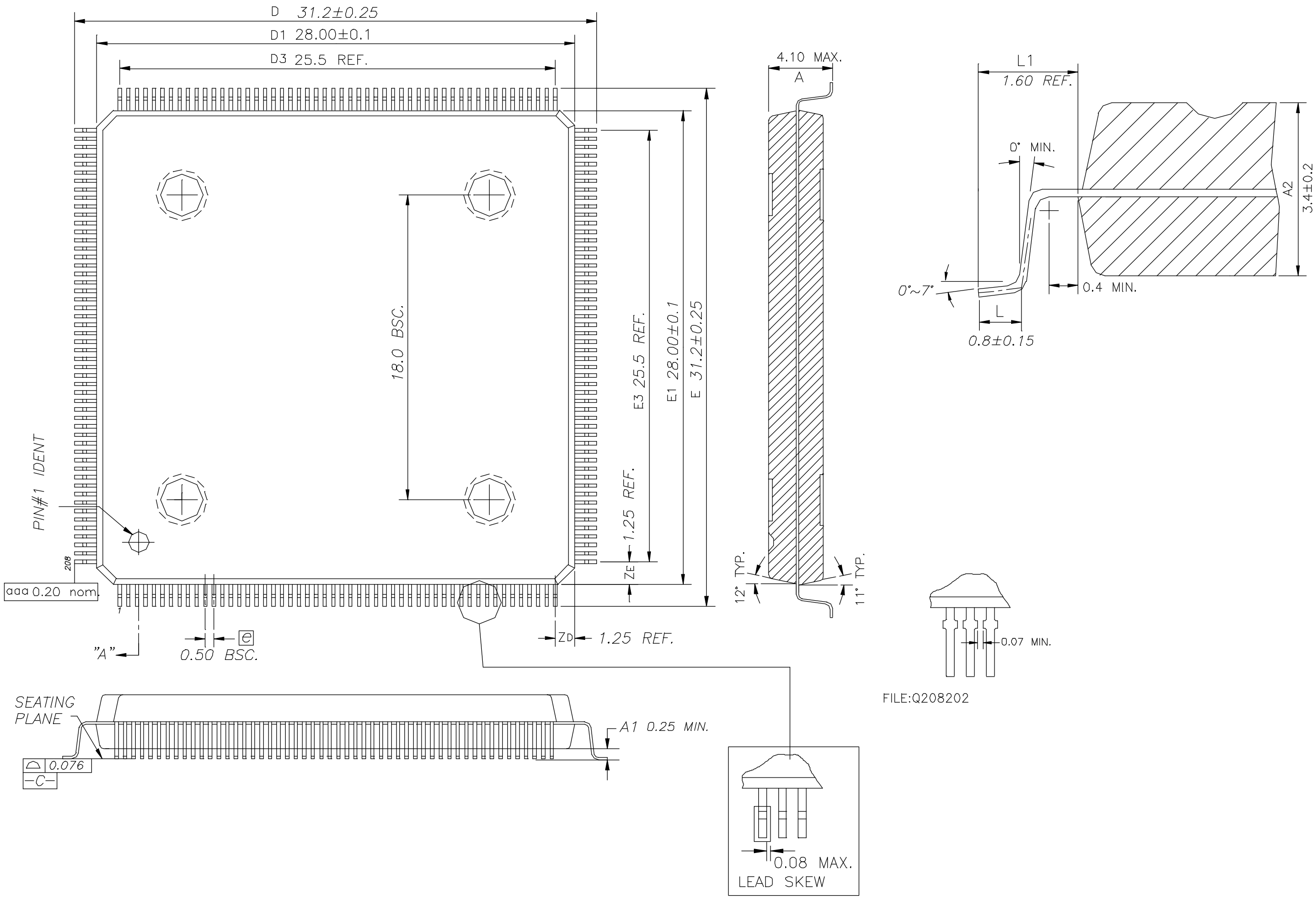
Parameter	**Description**	**Condition**	**Min**	**Typical**	**Max**	**Units**
Vcc	Supply Voltage		2.3	2.5	2.7	V
Vcc	Supply Voltage (I / O)		3.1	3.3	3.5	V
Icc	Power Supply	Vcc = 2.5V			675	mA
Icc	Power Supply (I / O)	Vcc = 3.3V			165	mA
Vil	Input LOW Voltage		-0.5		0.8	V
Vih	Input HIGH Voltage		2.0		3.8	V
Iil	Input LOW Leakage Current	Vin = 0.8V	-10		10	uA
Iih	Input HIGH Leakage Current	Vin = 2.0V	-10		10	uA
Vol	Output LOW Voltage	Iout =2~8mA	.		0.4	V
Voh	Output HIGH Voltage	Iout =-2~-8mA	2.4			V
Cinp	Input Pin Capacitance		5		8	pF
Lpinp	Pin Inductance		N/A			nH

Revision History

Revision	Date	Description
V1.31	Nov09	Original released file.
V1.34	Dec11	Page10: Modify LED description Page11: Modify address description.
V1.35	Dec26	Page12: Remove feature for big endian. Page18: Remove feature for carrier mode back pressure. Page20: Modify From_switch descriptors content. Page22: Modify system memory map.
V1.36	Dec31	Page23: Modify the address of register Page13&43: Remove GPIO[9]. Page15: Add 5.0.4, SDRAM initialization
V1.37	Jan14, (2002)	Page43: Modify INT_status.

ADM5106 / 5107 Package

Note: Scale = mm.


D 31.2±0.25
D1 28.00±0.1
D3 25.5 REF.
4.10 MAX.
A
L1
1.60 REF.
0° MIN.
A2
3.4±0.2
0°~7°
0.4 MIN.
L
0.8±0.15
18.0 BSC.
E3 25.5 REF.
E1 28.00±0.1
E 31.2±0.25
1.25 REF.
ZE
PIN#1 IDENT
208
aaa 0.20 nom.
"A"
0.50 BSC.
e
ZD
1.25 REF.
12° TYP.
11° TYP.
0.07 MIN.
FILE:Q208202
SEATING PLANE
A1 0.25 MIN.
0.076
-C-
0.08 MAX.
LEAD SKEW