Web Server Controller
(Embedded 8051, Offload engine and SRAM)

Features

- Built in an 8051
- Built in 32k byte SSRAM Macro
- Built in a 10/100 Ethernet MAC
- Built in a 4 channel DMA:
- Built in PLL to generate system clock
- Program starts at bank 0 0xFFFD
- Support external Flash I/F
- Support one MII/reverse MII or RMII
- Support one SMI (MDC/MDIO)
- Support EEPROM I/F
- Support 3 timers
   two 8-bit h/w auto load timers (13-bit/16-bit non-auto load timer)
   one 16-bit h/w auto load timer
- Support a high speed (up to 1Mbps) UART I/F
- Support 2 RS232
- Software Protocol Support ARP, ICMP, TCP/IP, UDP, HTTP server, TFTP server, DHCP client, Telnet server/client and SMTP
- Support Telnet, VCOM
- 0.25um CMOS technology
- 2.5V core and 3.3V (2.5V) IO power
- 128 pin package

General Description

The 128 pins IP210W is a very cost effective product and a highly integrated, 32K bytes SRAM, 10/100Mbps Ethernet MAC, offload engine and a 8051 for Web Controller applications.

The IP210W has a 10/100Mbps MAC interface and a external flash interface for Web controller application.

Table of Contents

Revision History

Revision #	Change Description
IP210W-DS-R01	Initial release.
IP210W-DS-R02	Modify MII signal naming on page 64, 65, 66
IP210W-DS-R03	Add flash AC timing on page 68
IP210W-DS-R04	Modify power consumption on page 62
IP210W-DS-R05	Add Junction Temperature for Absolute Maximum Rating on page 62
IP210W-DS-R06	Modify SMI Timing on page 67
IP210W-DS-R07	Modify RX_Filter_registers_0(0x8135)[1] description on page 45

1 PIN Diagram

1.1 IP210W 128 PIN
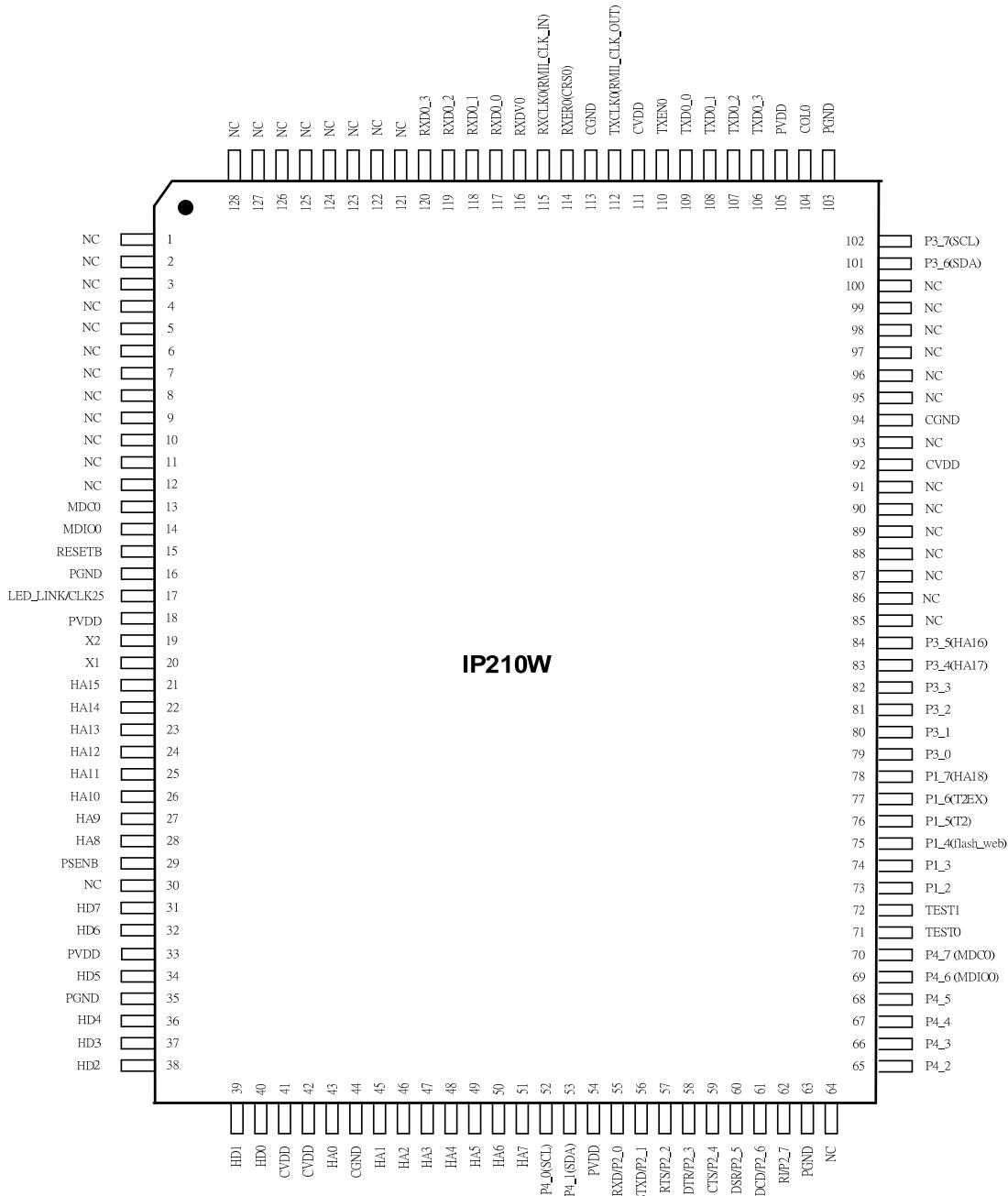
2 Block Diagram

IP210W block diagram
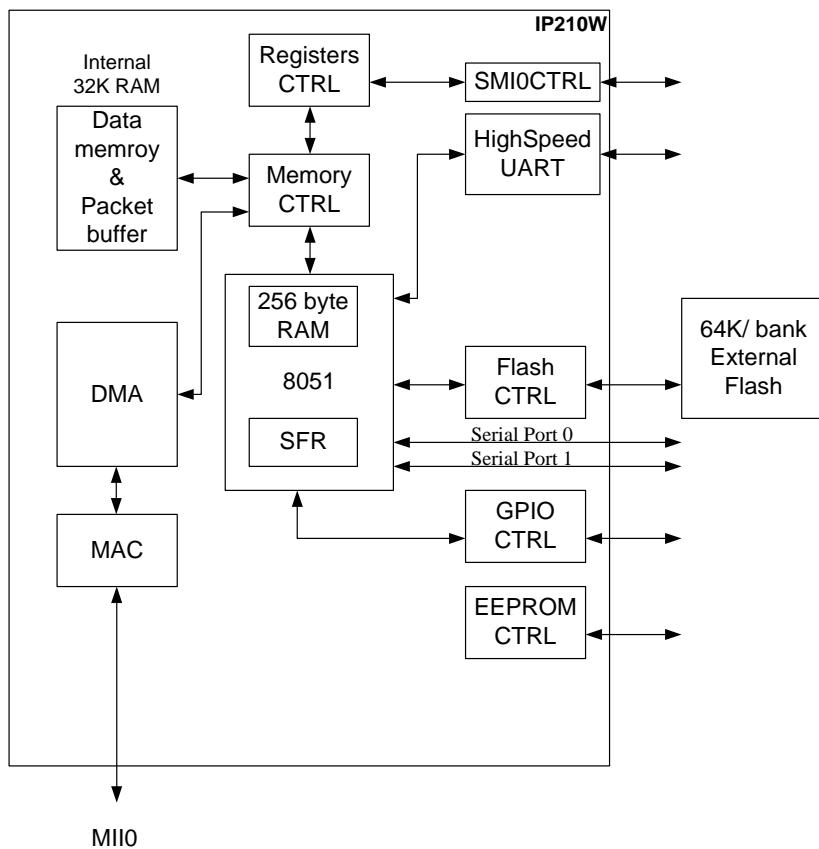
3 Application Diagram

3.1 With flash ROM
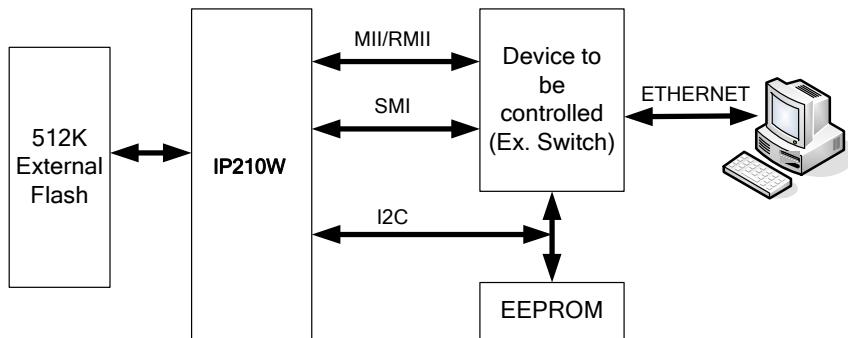
4   Pin description

Type	Description
I	Input pin
O	Output pin
IPL	Input pin with internal pull low 110k ohm
IPH	Input pin with internal pull high 120k ohm

Pin No.	Label	Type	Description
External program memory interface (in normal mode)			
21 22 23 24 25 26 27 28	HA15~HA8	O	**Program memory address bus [15:8]**
51 50 49 48 47 46 45 43	HA7~HA0	O	**Program memory address bus [7:0]**
31 32 34 36 37 38 39 40	HD7~0	IPH/O	**Program memory data bus [7:0]**
29	PSENB	O	**Program memory enable** 0: active 1: not active

Pin No.	Label	Type	Description
GPIO 1			
73 74 75 76 77 78	P1_2/SRXD1 P1_3/STXD1 P1_4/Flash_wenb P1_5(T2)/IODMA_rreqb P1_6(T2EX)/IODMA_wreqb P1_7/IODMA_readyb (HA[18])	IPH/O	Port1 is an 6-bit bidirectional I/O port. Port1 also serves the functions of various special features, as listed below: P1_2 SRXD1, serial input port 1 P1_3 STXD1, serial output port 1 P1_4 Flash_wenb, external flash write enable P1_5 IODMA_rreqb, read request for IODMA P1_6 IODMA_wreqb, write request for IODMA P1_7 IODMA_readyb, data ready for IODMA Note: P1_7 initial value = 1'b0

Pin No.	Label	Type	Description
GPIO 2 & High Speed UART			
55 56 57 58 59 60 61 62	P2_0/HSRXD P2_1/HSTXD P2_2/RTS P2_3/DTR P2_4/CTS P2_5/DSR P2_6/DCD P2_7/RI	IPH/O	Port2 is an 8-bit bidirectional I/O port. Port2 also serves the functions of high speed UART, as listed below: P2_0 HSRXD, serial input signal P2_1 HSTXD, serial output signal P2_2 RTS, Request To Send P2_3 DTR, Data Terminal Ready P2_4 CTS, Clear To Send P2_5 DSR, Data Set Ready P2_6 DCD, Data Carrier Detect P2_7 RI, Ring Indicator

Pin No.	Label	Type	Description
GPIO 3			

79
80
81
82
83
84
101
102
	P3_0/SRXD0
	P3_1/STXD0
	P3_2
	P3_3/INT1B
	P3_4/T0(HA[17])
	P3_5/T1(HA[16])
	P3_6/SDA
	P3_7/SCL
		IPH/O	Port3 is an 8-bit bidirectional I/O port. Port3 also serves the functions of various special features, as listed below:
			P3_0 SRXD0, serial input port 0
			P3_1 STXD0, serial output port 0
			P3_2
			P3_3 INT1, External interrupt 1
			P3_4 T0, Timer 0 external input
			P3_5 T1, Timer 1 external input
			P3_6 SDA, data pin of EEPROM
			       (when (0x8001[4] = 1'b1)
			P3_7 SCL, Clock pin of EEPROM
			       (when (0x8001[4] = 1'b1)
			
			Note: P3_4 initial value = 1'b0
			       P3_5 initial value = 1'b0

Pin No.	Label	Type	Description
GPIO 4			
52 53 65 66 67 68 69 70	P4_0/SCL P4_1/SDA P4_2 P4_3 P4_4 P4_5 P4_6/MDIO0 P4_7/MDC0	IPH/O	Port4 is an 8-bit bidirectional I/O port. Port4 also serves the functions of various special features, as listed below: **a. IODMA** : P4_7 ~ P4_0 IO DMA data bus **b. Other functions** P4_0 SCL, Clock pin of EEPROM (when (0x8001[3] = 1'b1) P4_1 SDA, data pin of EEPROM (when (0x8001[3] = 1'b1) P4_2 P4_3 P4_4 P4_5 P4_6 MDIO0, Management Data interface I/0 0 (when (0x8001[5] = 1'b1) P4_7 MDC0, Management Data Interface Clock 0 (when (0x8001[5] = 1'b1)

Pin No.	Label	Type	Description
MII 0 (for internal mac)			
13	MDC0	O	**Management Data Interface Clock 0** IP210W uses this interface to access the MII registers of external PHY
14	MDIO0	IPH/O	**Management Data interface I/0 0** IP210W uses this interface to access the MII registers of external PHY
MII0 MAC mode			
104	MMCOL0	I	**MII0 collision** It is an input signal and is connected to the MII_COL of external PHY
106 107 108 109	MMTXD0_3 MMTXD0_2 MMTXD0_1 MMTXD0_0	O	**MII0 transmit data** It is connected to MII_TXD of external PHY. It is sent out at the rising edge of MMTXCLK0.
110	MMTXEN0	O	**MII0 transmit enable** It is an output signal and is connected to MII_TXEN of external PHY. It is sent out at the rising edge of MMTXCLK0.
112	MMTXCLK0	I	**MII0 transmit clock** It is an input clock and it is connected to MII_TXCLK of external PHY.
115	MMRXCLK0	I	**MII0 receive clock** It is an input clock and it is connected to MII_RXCLK of external PHY.
114	MMRXER0 (MMCRS0)	IPL	**MII0 receive error or** **MII0 Carrier Sense** (when (0x8001[2]=1'b1)
116	MMRXDV0	IPL	**MII0 receive data valid** It is an input signal and is connected to the MII_RXDV of external PHY.
120 119 118 117	MMRXD0_3 MMRXD0_2 MMRXD0_1 MMRXD0_0	IPL	**MII0 receive data** It is input data and is connected MII_RXD[3:0] of external PHY. It is received at the rising edge of MMRXCLK0
MII0 PHY mode			
104	PMCOL0	O	**MII0 collision** It is active when MII0 is half duplex and a collision event happens
106 107 108 109	PMRXD0_3 PMRXD0_2 PMRXD0_1 PMRXD0_0	O	**MII0 receive data** It is sent out at the rising edge of PMRXCLK0.
110	PMRXDV0	O	**MII0 receive data valid** It is sent out at the rising edge of PMRXCLK0
112	PMRXCLK0	O	**MII0 receive clock**

115	PMTXCLK0	O	**MII0 transmit clock**
114	PMRXER0	O	**MII0 receive error**
116	PMTXEN0	IPL	**MII0 transmit enable** It is sampled at the rising edge of PMTXCLK0.
120 119 118 117	PMTXD0_3 PMTXD0_2 PMTXD0_1 PMTXD0_0	IPL	**MII transmit data** It is sampled at the rising edge of PMTXCLK0.
MII0 RMII mode			
108 109	RMTXD0_1 RMTXD0_0	O	**RMII0 transmit data** It is connected RMII_RXD[1:0] of external MAC or RMII_TXD[1:0] of external PHY.
110	RMTXEN0	O	**RMII0 transmit enable** It is connected RMII_RXDV of external MAC or RMII_TXEN of external PHY.
112	RMII0_CLK_OUT	O	**A 50Mhz reference clock output for other RMII devices**
115	RMII0_CLK_IN	I	**50Mhz RMII0 reference clock input**
116	RMRXDV0	IPL	**RMII0 receive data valid** It is connected RMII_RXDV of external PHY or RMII_TXEN of external MAC.
118 117	RMRXD0_1 RMRXD0_0	IPL	**RMII0 receive data** It is connected RMII_RXD[1:0] of external PHY or RMII_TXD[1:0] of external MAC.

Pin No.	Label	Type	Description
Miscellaneous			
15	RESETB	I	**Reset, low active** It should be kept at "low" for at least 10 microseconds.
17	CLK25/LINK_LED	IPH/O	CLK25(O)/ LINK_LED(O)
20	X1	I	**System clock input or crystal input** It is recommended to connect X1 and X2 to a crystal. If the clock source is from another chip, the clock should be active at least for 1ms before pin 15 RESETB de-asserted
19	X2	O	**Crystal output**
72, 71	TEST1, TEST0	IPH	**Chip Mode Select** **(1, 1) -> normal mode** **(0, 0) -> reserved** **(1, 0) -> reserved** **(0, 1) -> reserved**

Pin No.	Label	Type	Description
POWER&GND			
18 33 54 105	PVDD	I	3.3V (2.5V) PAD Power
41 42 92 111	CVDD	I	2.5V Core Power
16 35 63 103	PGND	I	PAD Ground
44 94 113	CGND	I	Core Ground

5   Function Description

5.1   CPU

5.1.1   Memory organization

IP210W manipulates operands in three memory spaces; they are 64K program memory for external flash, 256 bytes 8051 built-in data RAM, and 32K data memory.

The 256 bytes data RAM space is divided into 256-byte RAM and 128-byte 8051 Special Function Registers. The lower 128-byte of RAM can be accessed by direct or indirect addressing, the SFR can be access by direct addressing, and the upper 128-byte of RAM can be accessed by indirect addressing only. The 32K data RAM is accessed with instructions different from that for 256-byte RAM.

Program memory address map

0x0000		
	64 K External Flash	64K
0xFFFF		
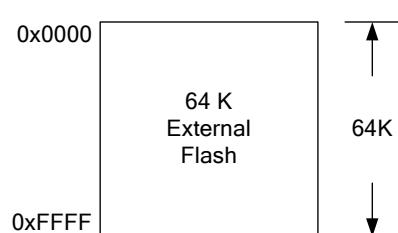
8051 built-in memory address map

0x00		
	128 byte internal direct/ indirect RAM	
0x7F		
0x80		
	128 byte internal indirect RAM	or 8051 SFR
0xFF		

Data memory address map

0x0000		
	Data memory	
0x3FFF		32K internal SRAM
0x4000		
	Packet buffer	
0x7FFF		
0x8000		
	MAC Registers	
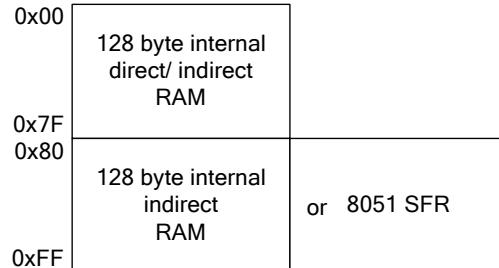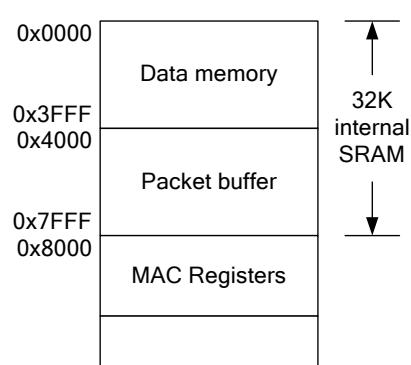
5.1.2 SFR MAP and its Default Value

Address	0 / 8	1 / 9	2 / A	3 / B	4 / C	5 / D	6 / E	7 / F
F8								
F0	B Register (00000000)							
E8	P4 (11111111)							WDTWCYC (00000000)
E0	ACC (00000000)	PDCON (XXXXXX01)						
D8	WDTCON (01000000)							
D0	PSW (00000000)							
C8	T2CON (00000000)	T2MOD (00000000)	RCAP2L (00000000)	RCAP2H (00000000)	TL2 (00000000)	TH2 (00000000)		
C0	SCON1 (00000000)	SBUF1						
B8	IP (00000000)							
B0	P3 (11111111)							
A8	IE (00000000)							
A0	P2 (11111111)							
98	SCON (00000000)	SBUF						
90	P1 (11111111)							
88	TCON (00000000)	TMOD (00000000)	TL0 (00000000)	TL1 (00000000)	TH0 (00000000)	TH1 (00000000)	**CKCON** (00000000)	
80	P0 (11111111)	SP (00000111)	DPL (00000000)	DPH (00000000)	DPL1 (00000000)	DPH1 (00000000)	DPS (XXXXXXX0)	PCON (0XXXXX00)

X bit is reserved bit.

5.1.3 Bit Addressable registers' Bit definition

Reg Address	Reg Name	Bit Address							
		0	1	2	3	4	5	6	7
D8	**WDTCON**	WDTRST	WDTEN	HWWDT_CLR	HWWDT_DIS	Flash_Access_En			SMOD_1
D0	**PSW**	PARITY	F1	OV	RS0	RS1	F0	AC	CY
C8	**T2CON**	CP/RL2	C/T2	TR2	EXEN2	TCLK	RCLK	EXF2	TF2
C0	**SCON1**	RI_1	TI_1	RB8_1	TB8_1	REN_1	Sm2_1	SM1_1	SM0_1
B8	**IP**	PX0	PT0	PX1	PT1	PS	PT2	PS1	
B0	**P3**	RXD	TXD	INT0	INT1	T0 (Bank A17)	T1 (Bank A18)		
A8	**IE**	EX0	ET0	EX1	ET1	ES	ET2	ES1	EA
98	**SCON**	RI	TI	RB8	TB8	REN	SM2	SM1	SM0
90	**P1**			RXD1	TXD1				(Bank A16)
88	**TCON**	IT0	IE0	IT1	IE1	TR0	TF0	TR1	TF1

5.1.4 Non Bit Addressable registers' bit definition

Reg Addres s	Reg Name	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
C9	**T2MOD**	DCEN	T2OE						
89	**TMOD**	M0	M1	C/T	GATE	M0	M1	C/T	GATE
87	**PCON**	IDL	PD						SMOD1
E1	**PDCON**	PDC	JWP						
8E	**CKCON**	MD0	MD1	MD2	T0M	T1M	T2M	WDT0	WDT1

5.1.5 Register Description (different from standard 8051)

Address	Register Name	Access	Description

E1	PDCON	RW	Power-down Control register
			Bit 0: PDC (Power Down Control)
			    0 – pull high of P0/P1/P2/P3 when entering power-down mode
			    1 – pull low of P0/P1/P2/P3 when entering power-down mode
			Bit 1: JWP (Just Wake Up, it works not only in PowerDown mode, but also in Idle mode)
			    0 – issue interrupt after WakeUp
			    1 – don't issue interrupt after WakeUp
			Bit2-7: reserved

D8	WDTCON	RW	Watch-dog control register
			Bit0 (WDTRST): Watch-dog reset
			Bit1 (WDTEN): Watch-dog enable
			Bit2 (HWWDT_CLR): Hardware Watch-dog clear
			Bit3 (HWWDT_DIS): Hardware Watch-dog disable
			Bit4: Flash_Access_En
			    0- default
			    1- redirect XDATA Bus to Flash's Address&DATA Bus
			Bit5-6: reserved
			Bit7: SMOD_1 – Serial modification. Double baud rate of serial port 1 if this bit set 1.

C8	T2CON	RW	Control register of Timer 2

C9	T2MOD	RW	Mode register of Timer 2
			Bit0:T2OE(Timer2 Output Enable)
			    Switching Timer 2 clock-out mode, which connects the programmable clock output to external pin T2.
			Bit1:DCEN(Down Count Enable)
			    1: Timer 2 as Down counter.
			    0: Timer 2 as Up counter.(default)

CA	RCAP2L	RW	Low byte of Timer 2 re-load register
CB	RCAP2H	RW	High byte of Timer 2 re-load register
CC	TL2	RW	Low byte of Timer 2 register
CD	TH2	RW	High byte of Timer 2 register
C0	SCON1	RW	Control register of Serial 1
C1	SBUF1	RW	Buffer register of Serial 1
E8	P4	RW	General Purpose IO

EF	WDTWCYC	RW	For every (WDTWCYC+1) CPU clocks, CPU advances WDT by 1. Used with Watch Dog Timer for easier Time Control.

5.1.6   Register description (8051 Standard)

Address	Register Name	Access	Description
0x87	PCON	RW	Power Control register: Bit 7- SMOD: this bit is used to Double baud rate when TIMER 1 is used to generate baud rate and Serial port is set in mode 1/2/3
0x88	SCON	RW	Serial Port Control Register: Bit [7:6]- Serial Port mode 　　00-Shift Register Baud Rate=Fosc/12 　　01-8 bit Baud Rate=variable 　　10-9 bit Baud Rate=Fosc/64 or Fosc/32 　　11-9 bit Baud Rate=variable Bit5- Enable Multi-processor communication Bit4- Rx_Enable Bit3- the 9th bit TX value when bit[7:6]=10/11 Bit2- the 9th bit RX value when bit[7:6]=10/11 Bit1 (TI)- TX Interrupt Status H/W set TI=1 when H/W has sent out data in SBUF. Bit0 (RI)- RX Interrupt Status H/W set RI=1 when H/W has received data in SBUF.
0x99	SBUF	RW	Serial Port Buffer: store the data to be transmitted out or received in.
0xA8	IE	RW	Interrupt Enable: Bit7- 0: disable all interrupts 　　　1: interrupt is depending on each interrupt's enable bit. Bit6- 0: disable Serial Port 1 interrupt　　(RI_1/TI_1) 　　　1: enable Serial Port 1 interrupt Bit5- 0: disable Timer 2 interrupt 　　　1: enable Timer 2 interrupt Bit4- 0: disable Serial Port interrupt (RI/TI) 　　　1: enable Serial Port interrupt Bit3- 0: disable Timer 1 interrupt 　　　1: enable Timer 1 interrupt Bit2- 0: disable External 1 interrupt 　　　1: enable External 1 interrupt Bit1- 0: disable Timer 0 interrupt 　　　1: enable Timer 0 interrupt Bit0- 0: disable External 0 interrupt 　　　1: enable External 0 interrupt
0xB8	IP	RW	Interrupt Priority: For each interrupt, 1 is high priority and 0 is low priority.
0x88	TCON	RW	Timer Control Register.
0x89	TMOD	RW	Timer Mode Control Register.
0x8A	TL0	RW	Timer0 LSB.
0x8B	TL1	RW	Timer1 LSB.

0x8C	TH0	RW	Timer0 MSB.
0x8D	TH1	RW	Timer1 MSB.
0x83	DPH	RW	DPTR MSB.
0x82	DPL	RW	DPTR LSB.
0x81	SP	RW	Stack Pointer.
0x80	P0	RW	General Purpose IO
0x90	P1	RW	General Purpose IO
0xA0	P2	RW	General Purpose IO
0xB0	P3	RW	General Purpose IO

5.1.7   CPU BootUp

5.1.7.1   Boot from 0xFFFD:

CPU starts up and runs instructions at address 0xFFFD of external Flash.

5.1.8   CPU Interruption

IP210W supports interruption while program's running. The following table shows the interrupt types IP210W adopted, which are quite similar to the standard 8051.

The only difference between IP210W and standard 8051 about interruption is that IP210W supports more than one (IE0, detected from INT0 pin) interrupt flag on External Request0 Interrupt, such as the bits in IP210's Status Register and Interrupt Enable Register. When the AND logic operation of IP210's Status and Interrupt Enable Register doesn't equal to zero, an External Request0 interrupt is issued as EX0=1 and EA=0 (interrupt enabled).

Therefore, when IP210W generates an External Request0 Interrupt, In the ISR (Interrupt Service Routine), IP210's Status Register should also be examined to know whether any of these operations, like Network TX/RX, DMA, TimerCounter and UART, is done or has a state change.

For the interrupt function of External request 0 is designed to respond only to the joint events of Interrupt Enable Register & Interrupt Status Register, pin P3.2 on IP210W is no longer used as the INT0 pin for External Request 0.

Interrupt Source	Vector Address	Request Flag	Enable Flag
External Request0	0003h	Status Register (0x8003)	Interrupt Enable Register and EX0
Timer0	000Bh	TF0/TCON.5	ET0/IE.1
External Request1	0013h	IE1/TCON.3	EX1/IE.2
Timer1	001Bh	TF1/TCON.7	ET1/IE.3
Serial Port0	0023h	T:TI/SCON.1, R:RI/SCON.0	ES/IE.4
Timer2	002Bh	TF2/T2CON.7	ET2/IE.5
Serial Port1	003Bh	T:TI1/SCON1.1, R:RI1/SCON.0	ES1/IE.6

5.1.9 CPU Timers/Counters

IP210, as like a standard 8052, has three timers/counters. SFR TMOD & TCON are used to configure timers' operation modes.

5.1.9.1 Timer0

Timer0 is a 16-bit timer/counter and functions just like one of a standard 8051
What is different here from the timer0 of standard 8051 is that, for INT0 pin is no longer existing, therefore, Gate bit(TMOD.bit3) for timer0 can't be used to control the operation of timer0 as TR0 is turned on.(see also interruption section 5.1.5)

5.1.9.2 Timer1

Timer1 is a 16-bit timer/counter and functions just like one of a standard 8051

5.1.9.3 Timer2

It's a 16-bit Timer and SFR T2MOD & T2CON are used to control its operations. It can count up & count down depending on TMOD.bit0(DCEN). T2 pin is multiplexed through P1.5 and T2EX is multiplexed through P1.6. The operation modes of Timer2 are shown as follows.

SFR 0xc8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/~T2	CP/~RL2

When C/~T2 bit(T2CON.bit1) is set to 1, it operates as a counter and is triggered by T2 pin.
When EXEN2 bit(T2CON.bit3) is set to 1, a negative edge of T2EX is enabled to set EXF2(T2CON.bit6) to 1 and cause a capture or a reload on Timer2.

SFR 0xc9	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
T2MOD							T2OE	DCEN

Mode		Description
Auto-Reload	T2MOD.bit0(CP/~RL2=0)	16-bit timer. TL2&TH2 are reloaded from RCAP2L&RCAP2H when overflow.
Capture	T2MOD.bit0(CP/~RL2=1)	16-bit timer. TL2&TH2 are captured to RCAP2L&RCAP2H when overflow. T2EX(P1.6) triggers the capture operation.
Baudrate Generator	T2CON.RCLK=1 or T2CON.TCLK=1	As the Baudrate Generator for Serial Ports by RCLK&TCLK of SFR T2CON.
ClockOut	T2OE=1	T2(P1.5)outputs as 50/50 duty-cycle clock. Its frequency is determined by the overflow rate of Timer2.
DownCount	DCEN=1	Timer2 counts down.

ClockOut Frequency = 58.9 MHz/(4*(65536-16bit_timer_value))

5.1.10   CPU MirrorMode

5.1.10.1   Purpose:

It's used to redirect CPU's Code Space into XDATA Space (Internal Memory), so that CPU fetches the instructions from Internal Memory instead of External Flash.

5.1.10.2   Action:

When ChipConfigure Register_0.bit5(Mirror_En) is set as 1, CPU runs in Mirror Mode. Before CPU entering Mirror Mode, the code data to be run should be moved to internal Memory. In Mirror Mode, if the content of Mirror_Address_Register(0x8006) is 0xE0, CPU sees Internal Memory address 0x0000 as 0xE000 of its Code Space. For example, if CPU to fetches an instruction at address 0xE005 of Code Space, it returns the content of address 0x0005 of Internal Memory.

5.1.11   CPU FlashAccess Mode

5.1.11.1   Purpose:

To allow CPU to do Erase, Write and Read operations on external Flash.This Mode only runs in Mirror Mode.

5.1.11.2   Erase & Write operation

When CPU's SFR register WDTCON.bit4 (Flash_Access_En) is set as 0, CPU's XDATA Space is mapped into internal Memory. When CPU's SFR register WDTCON.bit4 (Flash_Access_En) is set as 1, it enters FlashAccess Mode and its XDATA Space is redirected to the Address[15:0] and Data[7:0] BUS of external Flash, which means any of CPU's access to XDATA Space is redirected to Flash.. External Flash can be programmed to do Erase and Write operations through its Address[15:0] and Data[7:0] BUS.

5.1.11.3   Read operation

When CPU is in FlashAccess Mode, a (MOVX A, @DPTR) instruction makes CPU return to A the data at address DPTR of external Flash.

5.1.12   PowerManagement

5.1.12.1   Idle Mode

When PCON.IDL=1, IP210W enters Idle mode. In idle mode, IP210's CPU is idle but all the peripherals are remain active. The internal RAM and SFR registers remain unchanged too. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

5.1.12.2   Power Down Mode

When PCON.PD=1, IP210W will enter power-down mode. The CPU clock is stopped in this mode. This mode can be wakened up by hardware reset or by external enabled interrupt (INT0 or INT1) with level trigger activation (TCON.IT0=0 or TCON.IT1=0). The Program Counter, internal RAM and SFR registers retain their values and will not be changed after the power-down mode is terminated by external interrupt. The reset will restart the CPU, while the SFR with initial values except P0/P1/P2/P3 and the internal RAM retain same value as before entering power-down mode.

The value of P0/P1/P2/P3 after entering the power-down mode will be as the value defined by PDCON.PDC.
For example, when PDCON.PDC=0, port0 and port3 will be 0xFF.

5.2 DMA

5.2.1 Internal memory to internal memory

5.2.1.1 Purpose:

This function is used to move an amount of data from one internal Memory location to another.
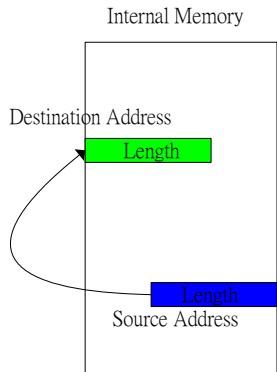
5.2.1.2 Action:

a. Initialize DMA_Source_Address, DMA_Destination_Address and DMA_Length registers.
b. Write 0x1 to DMA_COMMAND register to invoke Internal Memory to Internal Memory DMA operation.
c. IP210W starts this operation. When it's done, IP210W sets DMA_COMMAND=0 and StatusRegister.bit2=1.
d. Firmware keeps on polling to the content of DMA_COMMAND register. When it becomes 0x0, this DMA operation is finished.

5.2.2 External flash to internal memory

5.2.2.1 Purpose:

This function is used to move an amount of data from external Flash into internal Memory.
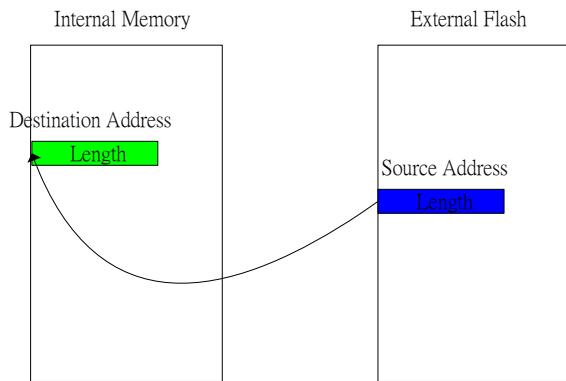
5.2.2.2 Action:

a. Initialize DMA_Source_Address, DMA_Destination_Address and DMA_Length registers.
b. Set CPU SFR EA=1 and Ex0=1 to enable external request interrupt so that an interrupt will occur to bring CPU out of IDLE mode when DMA operation's done.
c. Set Interrupt_EnableRegister.bit2=1 to allow an interrupt caused by DMA operation.
d. Write 0x4 to DMA_COMMAND register to tell IP210W an external Flash to internal Memory DMA operation will be started.
e. Set PDCON.bit1(JWP)=1 to not execute Interrupt Vector after
f. Set CPU SFR PCON.bit0(IDL) to 1 to switch CPU to IDLE mode.
g. After CPU switching to IDLE mode, IP210W starts this operation.

h. When this operation is done, IP210W sets StatusRegister.bit2=1 to generate an external request interrupt to bring back CPU from IDLE mode. IP210W sets DMA_COMMAND to 0x0.

5.2.3 GPIO to internal memory

5.2.3.1 Purpose

This DMA function is used to transfer an amount of data from an External Device to IP210's Internal Data Memory through multiplexed GPIO pins

5.2.3.2 Action

As shown in the following figure, P1.6(IODMA_WR#), P1.7(IODMA_RDY#), P4[7:0](DATA) and CPU clock are used to perform this operation.

   a. Firmware sets ChipConfigure Register_1.bit0 (IO_DMA_En) = 1 to enable this function.
    b. Initialize DMA_Source_Address, DMA_Destination_Address and DMA_Length registers.
    c. Write 0x3 to DMA_COMMAND register to put IP210W into a state that waits for GPIO transactions.
    d. Firmware keeps on polling whether DMA_COMMAND=0 to know that all data are moved into IP210.
    e. Device outputs the data to be written on P4[7:0] from External Device.
    f. Set P1.6=0 to tell IP210W about the incoming data.
    g. IP210W starts to move the target data to the destination address.
    h. IP210W sets P1.7=0 to tell External Device the data has been written successfully.
    i. IP210W sets P1.7=1 when it detect External Device sets P1.6 back to 1.
    j. IP210W continues to wait for the next data to be written.
    k. Device repeats step e to h until all data are moved into IP210.
    l. IP210W sets DMA_COMMAND back to zero and StatusRegister.bit2=1.
    m. This operation is done and firmware stop polling DMA_COMMAND register.
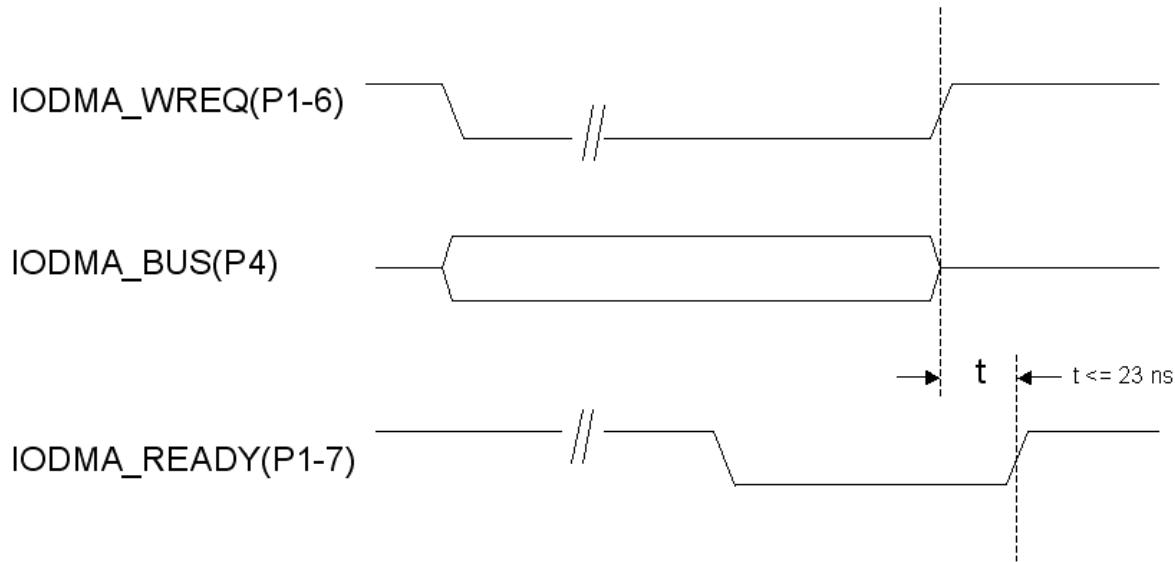
5.2.4 Internal memory to GPIO

5.2.4.1 Purpose

This DMA function is used to transfer an amount of data from IP210's Internal Data Memory to an External Device through multiplexed GPIO pins

5.2.4.2   Action

As shown in the following figure, P1.5(IODMA_RD#), P1.7(IODMA_RDY#), P4[7:0](DATA) and CPU clock are used to perform this operation.

a.   Firmware sets ChipConfigure Register_1.bit0 (IO_DMA_En) = 1 to enable this function.
b.   Initialize DMA_Source_Address, DMA_Destination_Address and DMA_Length registers.
c.   Write 0x2 to DMA_COMMAND register to put IP210W into a state that waits for GPIO transactions.
d.   Firmware keeps on polling whether DMA_COMMAND=0 to know that all data are moved into IP210.
e.   Device sets P1.5 to tell IP210W to send out the first data to be read on P4[7:0] from Internal Memory.
f.   IP210W sets P1.7=0 and outputs P4=(READ DATA).
g.   IP210W sets P1.7=1 when it detect External Device sets P1.5 back to 1.
h.   IP210W continues to wait for the next data to be read.
i.   Device repeats step e to g for the remaining data until all data are read from IP210.
j.   IP210W sets DMA_COMMAND back to zero StatusRegister.bit2=1.
k.   This operation is done and firmware stop polling DMA_COMMAND register.
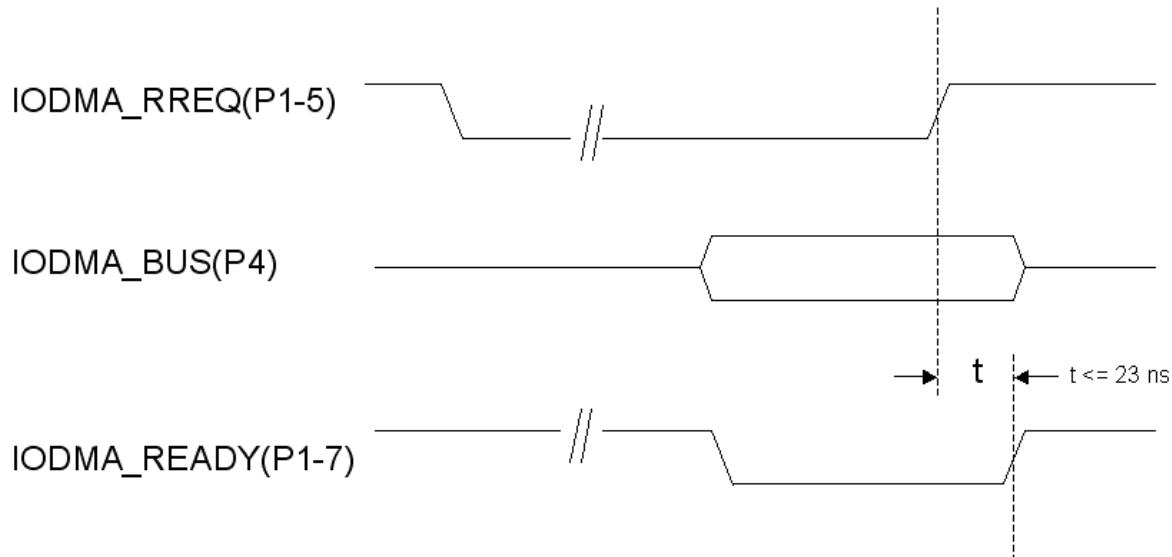
5.3   TimerCounter

5.3.1   Purpose:

To maintain a real time counter on IP210.

Related Registers
Timer Counter Register latch enable
Timer Counter Register_0 (LSB)
Timer Counter Register_1
Timer Counter Register_2
Timer Counter Register_3 (MSB)

5.3.2   Initialize real time counter:

   a. Set CPU's Timer/Counter 2 as 16bit auto-reload mode. Then make Timer/Counter2 run to generate overflows with fixed time interval.
   b. Write an initial value to TimerCounter register like 0x0000.
   c. TimerCounter register increments every time Timer/Counter 2 wraps around from 0xFFFF to 0x0000.

5.3.3   Read real time counter:

   a. Write 0x1 to TimerCounter_Latch_Enable register to latch real time counter to TimerCounter register.
   b. Read TimerCounter registers.

5.3.4   TimerCounter overflow

When the realtime TimerCounter wraps around from 0xffffffff to 0x00000000, IP210W sets StatusRegister.bit3=1 to notify Firmware (by Polling to StatusRegister or EX0 Interrupt).

5.4 MAC

5.4.1 TX MAC

5.4.1.1 TX descriptor registers

IP210W provides 4 TX descriptors as follows:
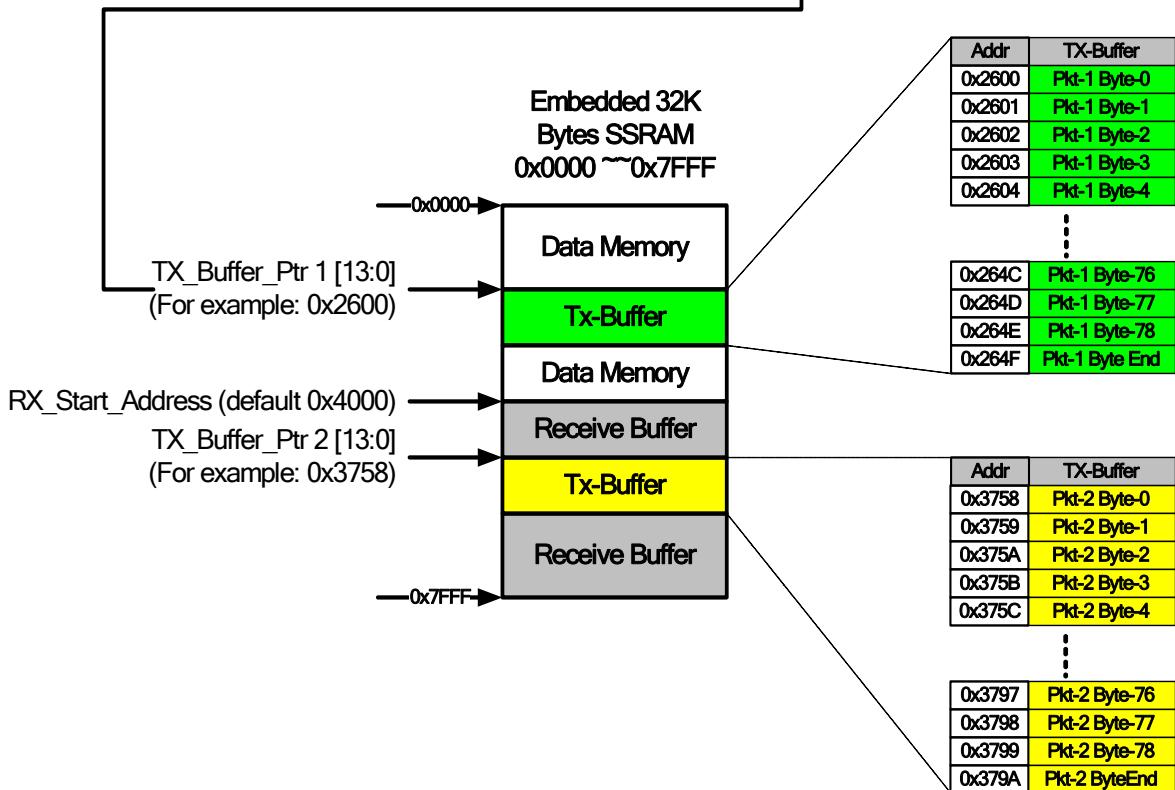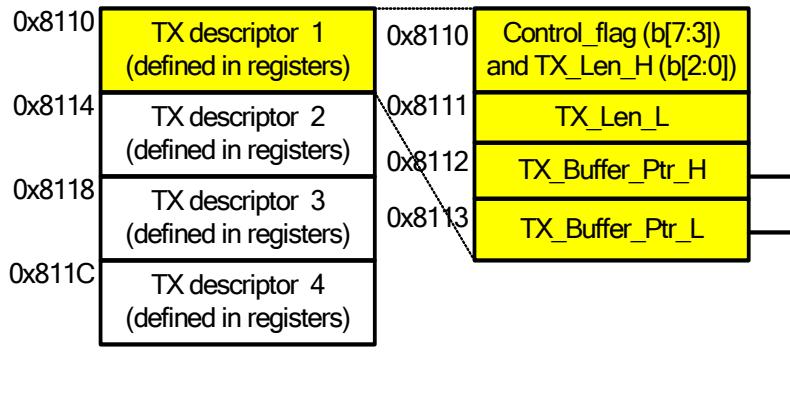
5.4.1.2   IP Checksum and CRC32 calculation for a proprietary packet

In some system, it uses packets with proprietary format. Because of the variable length of tag field, IP210W can't calculate the IP checksum or CRC excluding the tag field automatically. To solve the problem, IP210W supports a DMA function to calculate IP Checksum and CRC32.

5.4.1.3   RX buffer

RX Buffer is a block of internal 32K SSRAM for MAC to store the received frames. Its area is from the value of RX buffer start address to the end address of internal RAM (16Kbytes). It is a ring buffer. When the frame is cross the boundary of internal RAM, the MAC will automatically wrap around to the start address of RX Buffer.

5.4.1.4   RX Filter

IP210W supports two RX filter registers, RX_Filter_Registers_0 and RX_Filter_Registers_1. IP210W receives a packet if it meets any one of the conditions turned on in RX_Filter_Registers_0. IP210W receives a packet if it meets all of the conditions turned on in RX_Filter_Registers_1.

5.5 EEPROM I/F

IP210W supports access to EEPROM through SCL and SDA pins.

5.5.1 Configuration
5.5.1.1 Access through P3[7:6]

If Chip Configure Register_1[4:3] =2'b10(P3_I2C_En=1, P4_I2C_En=0), EEPROM function is enabled and P3.7=SCL & P3.6=SDA.
5.5.1.2 Access through P4[1:0]

If Chip Configure Register_1[4:3] =2'b01(P3_I2C_En=0, P4_I2C_En=1), EEPROM function is enabled and P4.0=SCL & P4.1=SDA.
5.5.1.3 Supported EEPROM types

EEPROM types	24C01	24C02	24C04	24C08	24C16
Read Page Max Byte Count	32	32	32	32	32
Write Page Max Byte Count	8	8	16	16	16

5.5.2 Related Registers

EEPROM_Data_Register
EEPROM_Addresss_ Register
EEPROM_ID_ Register
EEPROM_ Command_Register
EEPROM_ Control_Register

5.5.3 Address of EEPROM

EEPROM types	24C01	24C02	24C04	24C08	24C16
EEPROM_Addresss_Register	Address[7:0]	Address[7:0]	Address[7:0]	Address[7:0]	Address[7:0]
EEPROM_ID_Register.bit0	0	0	Address[8]	Address[8]	Address[8]
EEPROM_ID_Register.bit1	0	0	0	Address[9]	Address[9]
EEPROM_ID_Register.bit2	0	0	0	0	Address[10]
EEPROM_ID_Register.bit[7:3]	0	0	0	0	0

5.6   RS232

There are two kinds of RS232s in IP210. One is the standard 8051-built-in UART without FIFO (Serial Port0 & Serial Port1). Its control registers is in SFR. The other one is 16C650 compatible UART designed with FIFO and can support high-speed data transfer up to 1Mbps (determined by Divisor Register).

5.6.1   The operation of the RS232 in 8051

5.6.1.1   Serial Port0

P3.0(RXD) and P3.1(TXD) are used to perform data transfer through Serial Port0. SCON(0x98), PCON(0x87).bit7(SMOD) and SBUF(0x99) are SFRs that control its communication operations just like what it's like in a standard 8051. If needed, IE(0xA8).bit4(ES) is used to activate interrupt.

5.6.1.2   Serial Port1

P1.2(RXD_1) and P1.3(TXD_1) are used to perform data transfer. SCON1(0xC0), WDTCON(0xD8).bit7(SMOD_1) and SBUF1(0xC1) are SFRs that control its communication operations. If needed, IE(0xA8).bit6(ES1) is used to activate interrupt.

5.6.1.3   Modes

This RS232 clock source is~~ can be from Timer1 or Timer2 of 8051 so user has to do the correct Timer1 or Timer2 programming before RS232 starts working. Both Serial Ports share the same timer as their Baudrate Generator. As shown below are the four operation modes each Serial Port supports, as like those of standard 8051s:

SM0	SM1	Mode	Function	Baudrate
0	0	0	Synchronous Mode	Sys_CLK/12
0	1	1	8bit-UART	Variable
1	0	2	9bit-UART	Sys_CLK/64 or Sys_CLK/32
1	1	3	9bit-UART	Variable

5.6.2   The operation of High speed RS232 with FIFO

This RS232 is compatible with 16C650 UART, which supports full set of MODEM control signals. By setting the Divisor Register, it can support data transfer at up to 921.6kbps baudrate.

5.6.2.1   Enable High Speed UART

After Chip_Configure_1_Register.bit1(HSP_UART_En) is set to 1, GPIO 2(P2) is switched to function high speed UART operations. Its pin mapping is as follows:

P2.0	HSRXD
P2.1	HSTXD
P2.2	RTS
P2.3	DTR
P2.4	CTS
P2.5	DSR
P2.6	DCD
P2.7	RI

5.6.2.2 Related Registers' Briefing (Detailed definition can be found in IO Register Map)

5.6.2.2.1 Interrupt Enable Register (IER)

When one of bit0-bit4 of this register is set to 1, any related event will set STATUS.bit4 (HighSpeed_UART_Status_change) to 1, which possibly leads to CPU interrupt.

5.6.2.2.2 Interrupt Identification Register (IIR)

When an UART interrupt is issued, this register should be checked to know which event has occurred.

5.6.2.2.3 FIFO Control Register (FCR)

Bit0 is used to reset UART Receive module and bit1 is used to reset UART Transmit module. Bits[7:2] is used to determined the threshold number of bytes in FIFO required to enable the Received Data Available interrupt.

5.6.2.2.4 Line Control Register (LCR)

The line control register allows the specification of the format of the asynchronous data communication used, including the number of bits in a character, stop bit and parity setting. Bit7 is used to control the write action to Divisor register.

5.6.2.2.5 Modem Control Register (MCR)

The modem control register allows transferring control signals to a modem connected to the UART.

5.6.2.2.6 Line Status Register (LSR)

This register is used to tell the some status of UART, especially some error or notable events.

5.6.2.2.7 Modem Status Register (MSR)

The register displays the current state of the modem control lines

5.6.2.2.8 UART_RX_FIFO_STATUS

Read this register to get the current number of data received in the RX FIFO.

5.6.2.2.9 UART_TX_FIFO_STATUS

Read this register to get the max number of data that is allowed to push into TX FIFO before it turns full.

5.6.2.2.10 UART_Receiver Buffer

IP210W owns 256 bytes of UART Receive FIFO buffer. Read UART_Receiver_Buffer Register to get a byte of received data from Receive FIFO. After a read access to this register, the number of data in FIFO is decreased by one.

5.6.2.2.11 UART_Transmit Buffer

IP210W owns 256 bytes of UART Transmit FIFO buffer. Write UART_Transmit_Buffer Register to put a byte of data to Transmit FIFO. After a write access to this register, the number of data in FIFO is increased by one.

5.6.2.2.12 Divisor

The value of Divisor allows the selection of UART baudrate. Baudrate=(58.9M)/(16*divisor) bps.

5.7 MDC/MDIO I/F

5.7.1 The operation of the MDC/MDIO

IP210W supports a serial management interfaces (SMI). User can access IP210's MII registers through MDC and MDIO. Its format is shown in the following table. To access MII register in IP210, MDC should be at least one more cycle than MDIO. When the SMI is idle, MDIO is in high impedance.

Frame format	<Idle><preamble ><start><op code><PHY address><Registers address><turnaround><data><idle>
Read Operation	<Idle><optional 32-bit preamble><01><10><$A_4A_3A_2A_1A_0$><$R_4R_3R_2R_1R_0$><Z0><$b_{15} b_{14} b_{13} b_{12} b_{11} b_{10} b_9 b_8$ $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$><Idle>
Write Operation	<Idle><optional 32-bit preamble><01><01><$A_4A_3A_2A_1A_0$><$R_4R_3R_2R_1R_0$><10><$b_{15} b_{14} b_{13} b_{12} b_{11} b_{10} b_9 b_8$ $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$><Idle>

An example of a write frame without preamble, PHY address= 01h, and Register address=00h.
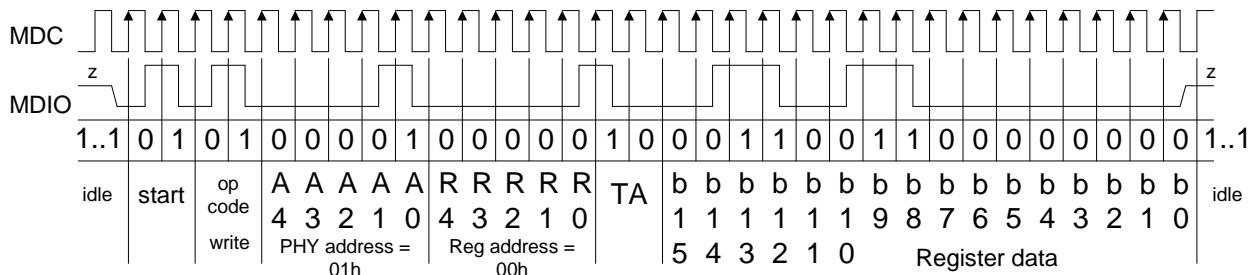
An example of a read frame without preamble, PHY address= 01h, and Register address=00h.
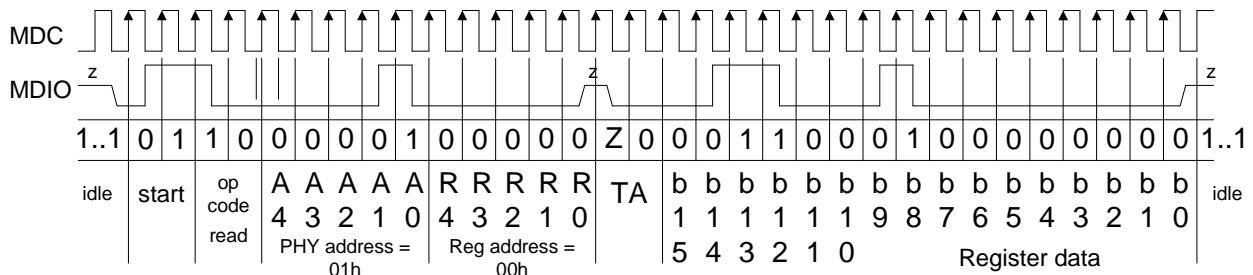
5.7.2 MDC/MDIO frame format

1. MD_Control_reg. Preamble_Disable = 0

	Preamble (32bit)	ST	OP	PHY ADDRESS	REG ADDRESS	DATA	IDLE
READ	1.............1	01	10	AAAAA	RRRRR	DDDDDDDDDDDDDDDD	Z
WRITE	1.............1	01	01	AAAAA	RRRRR	DDDDDDDDDDDDDDDD	Z

2. MD_Control_reg. Preamble_Disable=1

	Preamble (0 bit)	ST	OP	PHY ADDRESS	REG ADDRESS	DATA	IDLE
READ	--	01	10	AAAAA	RRRRR	DDDDDDDDDDDDDDDD	Z
WRITE	--	01	01	AAAAA	RRRRR	DDDDDDDDDDDDDDDD	Z

6 Register Description

6.1 Register Address Mapping

Address	Register
0x8000	Chip Configure Register_0
0x8001	Chip Configure Register_1
0x8002	CPU Control Register
0x8003	Status Register
0x8004	Interrupt Enable Register
0x8005	SW Reset Register
0x8006	Mirror Address Register
0x800f	**Timer Counter Register latch enable**
0x8010	**Timer Counter Register_0** (LSB)
0x8011	**Timer Counter Register_1**
0x8012	**Timer Counter Register_2**
0x8013	**Timer Counter Register_3** (MSB)
0x8014	PAD Control Register
0x8100	**MAC_Control_Register_0**
0x8101	**MAC_Control_Register_1**
0x8102	**MAC_Control_Register_2**
0x8103	**Pause-On-Threshold_Register**
0x8104	**Pause-Off-Threshold_Register**
0x8110	TX Descriptor 0_0 - **TX_buffer_pointer_L**
0x8111	TX Descriptor 0_1 - **TX_buffer_pointer_H**
0x8112	TX Descriptor 0_2 - **Tx_Pkt_Length_L**
0x8113	TX Descriptor 0_3 - **Control_flag and Tx_Pkt_Length_H**
0x8114	TX Descriptor 1_0 - **TX_buffer_pointer_L**
0x8115	TX Descriptor 1_1 - **TX_buffer_pointer_H**
0x8116	TX Descriptor 1_2 - **Tx_Pkt_Length_L**
0x8117	TX Descriptor 1_3 - **Control_flag and Tx_Pkt_Length_H**
0x8118	TX Descriptor 2_0 - **TX_buffer_pointer_L**
0x8119	TX Descriptor 2_1 - **TX_buffer_pointer_H**
0x811a	TX Descriptor 2_2 - **Tx_Pkt_Length_L**
0x811b	TX Descriptor 2_3 - **Control_flag and Tx_Pkt_Length_H**
0x811c	TX Descriptor 3_0 - **TX_buffer_pointer_L**
0x811d	TX Descriptor 3_1 - **TX_buffer_pointer_H**
0x811e	TX Descriptor 3_2 - **Tx_Pkt_Length_L**
0x811f	TX Descriptor 3_3 - **Control_flag and Tx_Pkt_Length_H**
0x8130	RX_Buffer_Start_Address
0x8131	RX_Buffer_Read_Pointer_L
0x8132	RX_Buffer_Read_Pointer_H

Address	Register
0x8133	RX_Buffer_Write_Pointer_L
0x8134	RX_Buffer_Write_Pointer_H
0x8135	RX_Filter_Registers_0
0x8136	RX_Filter_Registers_1
0x8137	Ether_Type_Start_Offset_Register
0x8138	Special_Source_Port_Tag_Type_Register_L
0x8139	Special_Source_Port_Tag_Type_Register_H
0x8200	**DMA_Command_Register**
0x8201	**DMA_Source_Address_Register_L**
0x8202	**DMA_Source_Address_Register_H**
0x8203	**DMA_Destination_Address_Register_L**
0x8204	**DMA_Destination _Address_Register_H**
0x8205	**DMA_Length_Register_L**
0x8206	**DMA_Length_Register_H**
0x8207	CRC_Result_Register_0
0x8208	CRC_Result_Register_1
0x8209	CRC_Result_Register_2
0x820a	CRC_Result_Register_3
0x820b	Preset_CRC_Value_Register
0x8310	EEPROM_Data_Register
0x8311	EEPROM_Addresss_ Register
0x8312	EEPROM_ID_ Register
0x8313	EEPROM_ **Command_Register**
0x8314	EEPROM_ **Control_Register**
0x8320	MD_Control_reg
0x8321	MD_PhyAddress
0x8322	MD_RegAddress
0x8323	MD_Data_Low
0x8324	MD_Data_High
0x8330	My MAC Address Byte 0 (LSB)
0x8331	My MAC Address Byte 1
0x8332	My MAC Address Byte 2
0x8333	My MAC Address Byte 3
0x8334	My MAC Address Byte 4
0x8335	My MAC Address Byte 5 (MSB)
0x8336	My IPV4 Byte 0 (LSB)
0x8337	My IPV4 Byte 1

Address	Register
0x8338	My IPV4 Byte 2
0x8339	My IPV4 Byte 3 (MSB)
0x8350	RMT_MAC Byte 0 (LSB)
0x8351	RMT_MAC Byte 1
0x8352	RMT_MAC Byte 2
0x8353	RMT_MAC Byte 3
0x8354	RMT_MAC Byte 4
0x8355	RMT_MAC Byte 5 (MSB)
0x8356	RMT IPV4 Byte 0 (LSB)
0x8357	RMT IPV4 Byte 1
0x8358	RMT IPV4 Byte 2
0x8359	RMT IPV4 Byte 3 (MSB)
0x8400	Chip ID LO
0x8401	Chip ID HI
0x8402	Chip Revision
0x8800	UART Receive Buffer (RO)
0x8801	UART_Transmit Buffer (WO)
0x8802	UART_Interrupt Enable
0x8803	UART_Interrupt Identification (RO)
0x8804	UART_FIFO Control
0x8805	UART_Line Control Register
0x8806	UART_Modem Control
0x8807	UART_Line Status(RO)
0x8808	UART_Modem Status(RO)
0x8809	UART_TX FIFO Status (RO)
0x880a	UART_RX FIFO Status (RO)
0x880b	UART_Clock Divisor Registers_L **(when the 7th (DLAB) bit of the Line Control Register is set to '1')**
0x880c	UART_Clock Divisor Registers_H **(when the 7th (DLAB) bit of the Line Control Register is set to '1')**

6.2 Register descriptions

Chip Configure Register_0 (0x8000)

Bit	Name	Access	Description	Default
0	Reserved	R		1
1	Reserved	R		1
3-2	MII_RMII_SEL [1:0]	RW	MAC I/F selection 00: reserved 01: MII 10: reverse MII 11: RMII	01
4	Reserved	RW		1
5	Mirror_En	RW	Redirect Code Bus to first 8K or 16K SRAM and the start address is defined in MirrorAddress register. 1: Enable redirect 0: Code Bus to Flash	0
6	Reserved	R/W		0
7	Reserved	R/W		0

Chip Configure Register_1 (0x8001)

Bit	Name	Access	Description	Default
0	IO_DMA_En	R/W	1: IO DMA enable 0: GPIO4	0
1	HSP_UART_En	R/W	1: High Speed UART enable 0: GPIO2	0
2	CRS0_En	R/W	1: CRS0 enable 0: CRS0 disable, use as RXER0	0
3	P4_I2C_En	R/W	1: P4 I2C interface enable (P4.0 = SCL, P4.1 =SDA) 0: GPIO4 bit0 & bit1	0
4	P3_I2C_En	R/W	1: P3 I2C interface enable (P3.7 = SCL, P3.6 =SDA) 0: GPIO3 bit6 & bit7	0
5	SMI_En	R/W	1: SMI interface enable (P4.7 = MDC, P4.6 =MDIO) 0: GPIO4 bit6 & bit7	0
6	Reserved	R/O		X
7	Reserved	R/O		X

CPU Control Register (0x8002)

Bit	Name	Access	Description	Default
0	Write_En	R/W	This bit is used to change this register bit[7:1] from RO to RW 1 – Enable Write Access to bit[7:1] 0 – RO to bit[7:1]	0
4-1	Flash_WaitState	R/W	Default is 0x7	0111
7-5	Reserved			

Status register (0x8003)

Bit	Name	Access	Description	Default
0	RX_Packet_Done	RC	H/W sets 1 to inform CPU that one or more packets are received in RX buffer.	0
1	TX_Packet_Done	RC	H/W sets 1 to inform CPU that TX packet is done.	0
2	DMA_Access_Done	RC	H/W sets 1 to indicate that DMA is done.	0
3	Timer_Counter_overflow	RC	H/W sets 1 to indicate that timer counter overflow.	0
4	HighSpeed_UART_Status_change	RC	H/W sets 1 to indicate that High Speed UART Status Changed	0
5	Reserved	RC		0
6	Reserved	RC		0
7	Reserved	RC		0

Interrupt Enable register (0x8004)

Bit	Name	Access	Description	Default
0	RX_Packet_Done_Enable	RW	1: enable RX_Packet_Done interrupt 0: disable RX_Packet_Done interrupt	0
1	TX_Packet_Done_Enable	RW	1: enable TX_Packet_Done interrupt 0: disable TX_Packet_Done interrupt	0
2	DMA_Access_Don_Enable	RW	1: enable DMA_Access_Done interrupt 0: disable DMA_Access_Done interrupt	0
3	Timer_Counter_overflow_Enable	RW	1: enable Timer_Counter_overflow interrupt 0: disable Timer_Counter_overflow interrupt	0
4	HighSpeed_UART_Status_Enable	RW	1: enable HighSpeed_UR_Status_change interrupt 0: disable HighSpeed_UR_Status_change interrupt	0
5	Reserved			0
6	Reserved			0
7	Reserved			0

SW Reset Register (0x8005)

Bit	Name	Access	Description	Default
0	**Reset**	RW	This bit is used to reset all the peripherals and registers except ChipConfig, CPU Control register and bit0/1 of MAC Control register 0 and CPU 1. Write 0 to this bit is ignored by IP210. 2. Write 1 to this bit will cause IP210W doing reset to all peripherals. This bit will be auto-cleared when reset is done.	0
7-1	Reserved			

Mirror Address Register (0x8006)

Bit	Name	Access	Description	Default
7-0	**Mirror Start Address**	RW	This register defines the starting address bit[15:8] of SRAM code when 8K_Mirror_En=1 or 16K_Mirror_En=1	00 h

Timer Counter Register latch enable (0x800f)

Bit	Name	Access	Description	Default
0	Timer Count latch enable	WO	1: Latch Current Timer Counter Register	0
7-1	Reserved			

Timer Counter Register 4 bytes (0x8013[MSB] ~ 0x8010[LSB])

Bit	Name	Access	Description	Default
31-0	Timer Count	R/W	Timer Counter Register is a 32-bit counter and is incremented upon the overflow of Timer2 (TF2). User can set Timer2 to determine the overflow intervals. Its value wraps around to 0x00 00 00 00 at Timer2's overflow while its previous value is 0xff ff ff ff.	00 00 00 00 h

PAD Control Register (0x8014)

Bit	Name	Access	Description	Default
1-0	PAD_Driving	R/W	00: 2 mA 01: 4 mA 10: 8 mA 11 12 mA	01
2	PAD_Speed	R/W	0: Normal 1: Fast	0
7-3	Reserved			

MAC_Control_register_0 (0x8100)

Bit	Name	Access	Description	Default
0	Speed100	RW	Speed setting bit (This bit is for RMII only): 1-100Mbps 0-10MbpsDriver use it to force the speed of MAC.	1
1	Duplex_F	RW	Duplex setting bit: 1- Full duplex 0- Half duplex, Driver use it to force the duplex mode of MAC.	1
2	TX_Enable	RW	Enable Transmission function of MAC: 0- disable 1- enable	0
3	RX_Enable	RW	Enable Receive function of MAC: 0- disable 1- enable	0
4	FlowControl_Enable	RW	1: Enable Flow Control function of MAC. In full duplex mode, MAC will act as follows: a.MAC will issue Pause frame with 0xFFFF when used RX buffer is over Pause-On-Threshold and continue to issue Pause frame with 0xFFFF only if remote node keeps on transmitting. b. MAC will send Pause frame with 0x0000 when used RX buffer is under Pause-Off-Threshold. c. MAC will stop transmitting if MAC receive a Pause frame with time > 0 and resume TX if MAC received a Pause frame with 0x0000 frame or timeout which is set by Pause frame with time > 0. In half duplex mode, MAC will do nothing. 0: Disable FlowControl	0
5	Boff_16_off	RW	This bit will disable maximum 16-retry limit and do infinite retry when the bit is set to 1.	0
6	LoopBack	RW	Enable MII-Internal-LoopBack when the bit is set to 1.	0
7	MaxFrameLen	RW	This bit set the maximum receive packet length. 1- 1536 bytes 0- 1522 bytes	1

MAC_Control_register_1 (0x8101)

Bit	Name	Access	Description	Default
0	FCS-append-disable	RW	0: TXMAC auto-calculates and auto-appends 4 bytes CRC at the end of packet. 1: TXMAC do NOT append 4 bytes CRC at the end of packet.	0
1	FCS-receive-enable	RW	0: RXMAC do NOT receive 4 bytes CRC into RX-Buffer. 1: RXMAC receive 4 bytes CRC into RX-Buffer.	0
2	SourcePortTagInserted_En	RW	0: RXMAC will not check if there is a SourcePortTag inserted right after SA. RXMAX treats the word right after SA as an EtherType. 1: RXMAC will check if there is a SourcePortTag, which has type value same as the SourcePortTagType register, inserted right after SA. If yes, RXMAX will skip 4 bytes from SA and treat the subsequent word as EtherType. If not, EtherType is considered right after SA. If SourcePortTagInserted_En="1" and first type != 0x9126(or SourcePortTag type Register), then drop the frame.	0
4-3	IGMP_Mode_En	RW	00 or 11 – RX MAC will treat IGMP frame as normal frame and filter the frame according to RX Filter rules. 01: RX MAC will receive in IGMP frame (IP frame with IP protocol=2) with DA=Multicast address of range 01-00-5E-00-00-00~~01-00-5E-7F-FF-FF and set the frame type to 1011b (IGMP frame). 10: RX MAC will receive in IGMP frame (IP frame with IP protocol=2) without DA constraint – Multicast or Unicast and set the frame type in RX buffer to 1011b (IGMP frame) **IGMP frame is an IP frame with IP protocol=2	00
5	Rx_8021X_En	R/W	0: RX-MAC will not receive frame with Multicast DA = 01-80-C2-00-00-03 1: RX-MAC will receive 802.1X frame with DA=01-80-C2-00-00-03 and set the frame type in RX buffer to 1100b (802.1X frame)	0
6	Reserved			
7	Reset_TXMAC	RW	When this bit is written to 1, the system will reset TXMAC and clear following registers. 1. Tx_Enable (0x8100[2]) 2. Boff_16_off (0x8100[5])	0

| | | | 3. FCS-append-disable (0x8101[0])
4. Tx Descriptors (0x8110~0x811f)
5. DMA_CMD_mode(0x8200[2:0])
P.S. when 0x 8200[2:0]=101 or 110
6. CRC_Result_Register
 (0x8207~0x820a)
This bit will be self-cleared after TXMAC reset is done. To write 0 to this bit is ignored by H/W. | |

MAC_Control_register_2 (0x8102)

Bit	Name	Access	Description	Default
0	IP_Checksum_Insp	RW	IP Checksum inspection function 1: enable 0: disable	1
1	TCP_Checksum_Insp	RW	TCP Checksum inspection function 1: IP210W drops the incoming packet if its TCP Checksum has errors 0: disable	1
2	UDP_Checksum_Insp	RW	UDP Checksum inspection function 1: IP210W drops the incoming packet if its UDP Checksum has errors 0: disable	1
3	ICMP_Checksum_Insp	RW	ICMP Checksum inspection function 1: IP210W drops the incoming packet if its ICMP Checksum has errors 0: disable	1
7-4	Reserved	-		

Pause -On-Threshold_register (0x8103)

Bit	Name	Access	Description	Default
7-0	Pause -On-Threshold	RW	(Unit is 256 bytes) Pause-On-Threshold[7:0] = 0x30= 8'd48 ---> 48*256bytes = 12K bytes	30 h

Pause -Off-Threshold_register (0x8104)

Bit	Name	Access	Description	Default
7-0	Pause -Off-Threshold	RW	(Unit is 256 bytes) Pause-Off-Threshold[7:0] = 0x18= 8'd24 ---> 24*256bytes = 6K bytes	18 h

TX Descriptor 0_0 - TX_buffer_pointer_L (0x8110)

Bit	Name	Access	Description	Default
7-0	TX_Buffer_Ptr_L	RW	This field defines TX buffer pointer bit [7:0]	00 h

TX Descriptor 0_1 - TX_buffer_pointer_H (0x8111)

Bit	Name	Access	Description	Default
7-0	TX_Buffer_Ptr_H	RW	This field defines TX buffer pointer bit [13:8]	00 h

TX Descriptor 0_2 - Tx_Pkt_Length_L (0x8112)

Bit	Name	Access	Description	Default
7-0	TX_Len_L	RW	This field defines TX frame len bit [7:0]	00 h

TX Descriptor 0_3 - Control_flag and Tx_Pkt_Length_H (0x8113)

Bit	Name	Access	Description	Default
2-0	TX_Len_H	RW	This field defines TX frame len bit [10:8]	000
5-3	Checksum_Packet_Type	RW	This field enable packet by packet MAC checksum insertion action: 000- No IP checksum insertion requirement 001- IP/TCP IPv4 checksum insertion 010- IP/UDP IPv4 checksum insertion 011- IP/ICMP IPv4 checksum insertion	000
6	TX_Error	RO	This bit will be auto-clear when bit 7 Start_TX is written to 1. This bit is valid only when bit 7 Start_TX is written to 0. 0: TX no error 1: TX error such as Txunderrun, MaxCollision, LateCollision, etc.	0
7	Start_TX	RW	The TX MAC will start to send this packet when this bit is set one and auto-clear when TX is done. To write this bit to 0 will be ignored by H/W.	0

TX Descriptor 1_0 ~ TX Descriptor 1_3 (0x8114 ~ 0x8117)
TX Descriptor 2_0 ~ TX Descriptor 2_3 (0x8118 ~ 0x811b)
TX Descriptor 3_0 ~ TX Descriptor 3_3 (0x811c ~ 0x811f)

Rx-Buffer-Start-Address (0x8130)

Bit	Name	Access	Description	Default
7-0	Rx-Buffer-Start-Address	RW	This byte defines the high byte of RX Buffer Start address in internal RAM. The default value is 40h. The low byte of RX Buffer Start address is always 00h	40 h

RX_Buffer_Read_Pointer_L (0x8131)

Bit	Name	Access	Description	Default
7-0	Rx-Buffer-Read-Pointer-L	RW	Rx-Buffer-Read-Pointer is maintained by S/W to store the address of the first unread received frame. The default value is 00h	00 h

RX_Buffer_Read_Pointer_H (0x8132)

Bit	Name	Access	Description	Default
7-0	Rx-Buffer-Read-Pointer-H	RW	Rx-Buffer-Read-Pointer is maintained by S/W to store the address of the first unread received frame. The default value is 40h	40 h

RX_Buffer_Write_Pointer_L (0x8133)

Bit	Name	Access	Description	Default
7-0	Rx-Buffer-Write-Pointer-L	RW	Rx-Buffer-Write-Pointer is maintained by H/W to store the address of the incoming frame. The default value is 00h	00 h

RX_Buffer_Write_Pointer_H (0x8134)

Bit	Name	Access	Description	Default
7-0	Rx-Buffer-Write-Pointer-H	RW	Rx-Buffer-Write-Pointer is maintained by H/W to store the address of the incoming frame. The default value is 40h.	40 h

RX_Filter_registers_0 (0x8135)

Bit	Name	Access	Description	Default
0	Rx_My_Mac_En	RW	When this bit is set, MAC will filter out the unicast frame except its DA equal to MY_MAC_ADDRESS. Otherwise the MAC will filter out all unicast frames. Pause frame with my MAC unicast DA is an exception. It is controlled by bit 5 – Rx_Pause_En.	1
1	Rx_Mcst_En	RW	This bit=1 enable MAC to receive all multicast and broadcast frame except BPDU and PAUSE and IGMP and 802.1X frame. Otherwise MAC will filt out all multicast and broadcast frame except BPDU and PAUSE and IGMP and 802.1X frame (Default 0) BPDU and PAUSE frame filters are defined in bit 4 and bit5. (Default 0) IGMP and 802.1X are defined in MAC_Control_register_1, Bit-4-3 and Bit-5 (Default 0) Note: DA of BPDU : 01-80-C2-00-00-00 DA of PAUSE : 01-80-C2-00-00-01 DA of IGMP: 01-00-5E-00-00-00 ~~ 01-00-5E-7F-FF-FF DA of 802.1X: 01-80-C2-00-00-03	0
2	Rx_Bcast_En	RW	This bit =1 enable MAC to receive Broadcast frame. Otherwise MAC will filter out Broadcast frame except Broadcast ARP and Broadcast RARP frame.	0
3	Rx_All_En	RW	This bit =1 enable MAC to receive all good frame except Pause packet. Otherwise MAC will depend on the setting of the other bits of RX_Mode_0 and RX_Mode_1.	0
4	Rx_Bpdu_En	RW	MAC will receive BPDU packet only if Rx_Bpdu_En=1. Otherwise MAC will filter out BPDU packet. BPDU is a frame with DA=01-80-C2-00-00-00	0
5	Rx_Pause_En	RW	MAC will receive Pause packet only if Rx_Pause_En=1 and MAC_Control_register_0.FlowControl_En =0. Otherwise MAC will filter out Pause packet. Pause packet is a frame with DA =01-80-C2-00-00-01 or	0

Bit	Name	Access	Description	Default
			My_MAC_Address Type = 0x8808 OP Code= 0x0001	
6	Rx_Remote_Mac	RW	MAC will receive only the frame with SA=Remote_MAC_Address when this bit set 1. This bit is used to lock remote node's MAC Address.	0
7	Rx_CRCErr_En	RW	0: MAC will filter out CRC error frame. 1: Enable MAC to receive CRC error frame.	0

RX_Filter_registers_1 (0x8136)

Bit	Name	Access	Description	Default
0	Rx_MyIP_En	RW	MAC will receive only the frame with Destination IP = My_IP when this bit is set 1.	1
1	Rx_RemoteIP_En	RW	MAC will receive only the frame with Source IP = Remote_IP when this bit is set 1.	0
3-2	Rx_IP_Type_En	RW	00 - Receive all EtherType frame except the setting defined in RX_Filter_register_0.Rx_Pause_En 01- Receive EtherType only IPv4(0x800), ARP(0x806), RARP(0x8035). *Notice: If the EtherType is 0x0800 but the subsequent byte is not equal to IPV4 version, MAC drops the frame. ****Notice: If the EtherType is 0x8100 (VLAN packet), No matter the setting of Rx_IP_Type_En value, RX-MAX will always skip 4 bytes VLAN tagging and treat the subsequent word as EtherType.	01
7-4	Reserved	-	-	

Ether_Type_Start_Offset_Register (0x8137)

Bit	Name	Access	Description	Default
7-0	Ether_Type_Start_Offset	RW	Define the Byte-Offset of EtherType Field from SA-Field of receiving packet The default value is 0. It means EtherType field is right after SA field in the receiving frame. Ex: If Ether_Type_Start_Offset= 0x6, the EtherType field will be located at Byte 18/19 in the receiving frame start the byte count from 0.	00 h

Special_Source_Port_Tag_Type_Register_L (0x8138)

Bit	Name	Access	Description	Default
7-0	Special_Source_Port_Tag_ Type_Register_L	RW	Source Port Tag Type Value low byte	26 h

Special_Source_Port_Tag_Type_Register_H (0x8139)

Bit	Name	Access	Description	Default
7-0	Special_Source_Port_Tag_ Type_Register_H	RW	Source Port Tag Type Value high byte	91 h

DMA_Command_Register (0x8200)

Bit	Name	Access	Description	Default
2-0	CMD_mode	RW	000-No DMA operating or DMA done. 001-Internal Data to Internal Data transfer. 010-Internal Data to IO transfer. 011-IO to Internal Data transfer. 100-Flash to Internal Data transfer. 101-Interanl to CRC32 generation. The CRC32 result will be calculated by IP210W and be stored in CRC_Result_register (0-3). 110-Interanl to checksum generation and insert into checksum fields in IP packet memory pointed by DMA_Source_Address.	000
3	Reserved	-	-	
7-4	Packet_Type	RW	Tell MAC the packet type when Command_mode is 'b110: 0000- no checksum generation needed 0001- IP and TCP (IPv4) 0010- IP and UDP (IPv4) 0011- IP and ICMP (IPv4) Others- no checksum generation needed.	0000

DMA_Source_Address_Register_L (0x8201)

Bit	Name	Access	Description	Default
7-0	DMA_Source_Address_L	R/W	S/W sets DMA source address low byte	00 h

DMA_Source_Address_Register_H (0x8202)

Bit	Name	Access	Description	Default
7-0	DMA_Source_Address_H	R/W	S/W sets DMA source address high byte	00 h

DMA_Destination_Address_Register_L (0x8203)

Bit	Name	Access	Description	Default
7-0	DMA_Destination_Address_L	R/W	S/W sets DMA Destination address low byte	00 h

DMA_Destination_Address_Register_H (0x8204)

Bit	Name	Access	Description	Default
7-0	DMA_Destination_Address_H	R/W	S/W sets DMA Destination address high byte	00 h

DMA_Length_Register_L (0x8205)

Bit	Name	Access	Description	Default
7-0	DMA_Length_L	R/W	It specifies the length of data for IP checksum or CRC32 calculation. The maximum value of DMA_length is 2047.	00 h

DMA_Length_Register_H (0x8206)

Bit	Name	Access	Description	Default
7-0	DMA_Length_H	R/W	It specifies the length of data for IP checksum or CRC32 calculation. The maximum value of DMA_length is 2047.	00 h

CRC_Result_Register(3~0) (0x820a ~ 0x8207)

Bit	Name	Access	Description	Default
31-0	CRC_Result	R/W	These registers store the calculating result of CRC32. There are two conditions to set the register to 0xFFFFFFFF: Power on reset Write 1 to Preset_CRC_Value_Register	FF FF FF FF h

Preset_CRC_Value_Register (0x820b)

Bit	Name	Access	Description	Default
0	Preset_CRC_Value	RW	Setting this bit to 1 will trigger HW_CRC to preset CRC_Result_register(0-3) to 0xFFFFFFFF. This bit will be self-cleared when HW-CRC has done the reset of CRC_Result_register. Write 0 to this bit is ignored by HW_CRC.	0
7-1	Reserved			

EEPROM_Data_Register (0x8310)

Bit	Name	Access	Description	Default
7-0	EEPROM_Data	RW	It stores the data to/from EEPROM.	00 h

EEPROM_Addresss_ Register (0x8311)

Bit	Name	Access	Description	Default
7-0	EEPROM_Addr	RW	It defines the address of data in EEPROM.	00 h

EEPROM_ID_ Register (0x8312)

Bit	Name	Access	Description	Default
2-0	EEPROM_ID	RW	It defines EEPROM ID or EEPROM address high bits for EEPROM with size over 256 bytes such as 24C16/24C08/24C04. <table><tr><td></td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr><tr><td>24C01</td><td>ID</td><td>ID</td><td>ID</td></tr><tr><td>24C02</td><td>ID</td><td>ID</td><td>ID</td></tr><tr><td>24C04</td><td>ID</td><td>ID</td><td>Addr[8]</td></tr><tr><td>24C08</td><td>ID</td><td>Addr[9]</td><td>Addr[8]</td></tr><tr><td>24C16</td><td>Addr[10]</td><td>Addr[9]</td><td>Addr[8]</td></tr></table> The ID bit(s) should be filled with 0. <u>For example:</u> 　24C16 – bit[2:0] define address[10:8] 　24C08 – bit[2] is ID[A2], bit[1:0] define address[9:8]	000
7-3	Reserved			

EEPROM_ Command_Register (0x8313)

Bit	Name	Access	Description	Default
4-0	Byte_count	RW	Define the access size. The access size is (Byte_count+1) bytes. In EEPROM writing operation, if the access size over a page (16bytes or 8 bytes which is depending EEPROM type), the over part will turn back to the beginning address of the page and overwrite the previous data.	00000
5	RW_op	RW	Define the operation: 0- Write 1- Read	0
6	Abort	RW	This bit=1 is used to indicate H/W abort the access operation defined by RW_op This bit will be auto-clear by H/W after S/W read or write the data.	0
7	Next	RW	This bit=1 is used to indicate H/W can read or write the next data when S/W asks multiple byte access. This bit will be auto-clear by H/W after S/W read or write the data.	0

EEPROM_Control_ Register (0x8314)

Bit	Name	Access	Description	Default
0	EE_Clk_Sel	RW	0: EEPROM clock rate is 58.9KHz, for normal usage. 1: Speed-up mode, EEPROM clock rate is 5.89MHz.	0
7-1	Reserved		.	

MD_Control_reg (0x8320)

Bit	Name	Access	Description	Default
2-0	MDC_Clock_Select	RW	000 – 34ns 001 – 68ns 010 – 136ns 011 – 272ns 100 – 544ns(recommend) 101 –1088ns 110 – 2176ns	000
3	Reserved			
4	RW_op	RW	1: for Read operation 0: for Write operation	0
5	Preamble_Disable	RW	0: for Preamble enable 1: for Preamble disable	0
6	Reserved			
7	Start	RW	Set 1 to start MDC/MDIO operation Auto-clear when the operation is completed.	0

MD_PhyAddress (0x8321)

Bit	Name	Access	Description	Default
4-0	Phy Address	RW	Phy Address	00000
7-5	Reserved			

MD_RegAddress (0x8322)

Bit	Name	Access	Description	Default
4-0	Register Address	RW	Register Address	00000
7-5	Reserved			

MD_Data_Low (0x8323)

Bit	Name	Access	Description	Default
7-0	MD_Data_Low	RW	MD_Data[7:0]	00 h

MD_Data_High (0x8324)

Bit	Name	Access	Description	Default
7-0	MD_Data_High	RW	MD_Data[15:8]	00 h

My MAC Address 6 bytes (0x8335[MSB] ~ 0x8330[LSB])

Bit	Name	Access	Description	Default
47-0	My MAC Address	R/W	My MAC Address [47:0]	00 00 00 00 00 00 h

My IPv4 Address 4 bytes (0x8339[MSB] ~ 0x8336[LSB])

Bit	Name	Access	Description	Default
31-0	My IPv4 Address	R/W	My IPv4 Address [31:0]	00 00 00 00 h

Remote MAC Address 6 bytes (0x8355[MSB] ~ 0x8350[LSB])

Bit	Name	Access	Description	Default
47-0	Remote MAC Address	R/W	Remote MAC Address [47:0]	00 00 00 00 00 00 h

Remote IPv4 Address 4 bytes (0x8359[MSB] ~ 0x8356[LSB])

Bit	Name	Access	Description	Default
31-0	Remote IPv4 Address	R/W	Remote IPv4 Address [31:0]	00 00 00 00 h

Chip ID LO (0x8400)

Bit	Name	Access	Description	Default
7-0	Chip ID LO	RO	Chip ID Number Low byte	10 h

Chip ID HI (0x8401)

Bit	Name	Access	Description	Default
7-0	Chip ID HI	RO	Chip ID Number high byte	02 h

Chip Revision (0x8402)

Bit	Name	Access	Description	Default
7-0	Chip Revision	RO	Chip reversion Number	00 h

UART_Receiver Buffer (0x8800)

Bit	Name	Access	Description	Default
7-0	UART_Receiver Buffer	RO	UART Receiver FIFO output.	00 h

UART_Receiver Buffer (0x8801)

Bit	Name	Access	Description	Default
7-0	UART_Receiver Buffer	WO	UART Transmit FIFO input.	

Interrupt Enable Register IER (0x8802)
This register allows enabling and disabling interrupt generation by the UART

Bit	Name	Access	Description	Default
0	Received Data available interrupt	RW	'0' – disabled '1' – enabled	0
1	Transmitter Holding Register empty interrupt	RW	'0' – disabled '1' – enabled	0
2	Receiver Line Status Interrupt	RW	'0' – disabled '1' – enabled	0
3	Reserved	RW	Reserved. Should be logic '0'.	
4	Received Data Timeout interrupt	RW	'0' – disabled '1' – enabled	0
7-5	Reserved	RW	Reserved. Should be logic '0'.	

Interrupt Identification Register IIR (0x8803)
The IIR enables the programmer to retrieve what is the current highest priority pending interrupt.
Bit 0 indicates that an interrupt is pending when it's logic '0'. When it's '1' – no interrupt is pending.
The following table displays the list of possible interrupts along with the bits they enable, priority, and their source and reset control.

Bit 3	Bit 2	Bit 1	Priority	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	1	1	1st	Receiver Line Status	Parity, Overrun or Framing errors or Break Interrupt	Reading the Line Status Register
0	1	0	2nd	Receiver Data available	FIFO trigger level reached	FIFO drops below trigger level
1	1	0	2nd	Timeout Indication	There's at least 1 character in the FIFO but no character has been input to the FIFO or read from it for the last 4 Char times.	Reading from the FIFO (Receiver Buffer Register)
0	0	1	3rd	Transmitter Holding Register empty	Transmitter Holding Register Empty	Writing to the Transmitter Holding Register or reading IIR.

Bits 4 and 5: Logic '0'.
Bits 6 and 7: Logic '1' for compatibility reason.
Default Value: C1 h

FIFO Control Register FCR (0x8804)

The FCR allows selection of the FIFO trigger level (the number of bytes in FIFO required to enable the Received Data Available interrupt). In addition, the FIFOs can be cleared using this register.

Bit	Name	Access	Description	Default
0	Clears the Receiver FIFO	RW	Writing a '1' to bit 1 clears the Receiver FIFO and resets its logic. But it doesn't clear the shift register, i.e. receiving of the current character continues.	0
1	Clears the Transmitter FIFO	RW	Writing a '1' to bit 2 clears the Transmitter FIFO and resets its logic. The shift register is not cleared, i.e. transmitting of the current character continues.	0
7-2	Receiver FIFO Interrupt trigger level	RW	Define the Receiver FIFO Interrupt trigger level[7:2], Interrupt trigger level[1:0] always equal to 2'b00 Ex '000001' – 4 bytes {000001, 00} '010000' –64 bytes {010000, 00} (default value)	010000

Line Control Register LCR (0x8805)
The line control register allows the specification of the format of the asynchronous data communication used. A bit in the register also allows access to the Divisor Latches, which define the baud rate. Reading from the register is allowed to check the current settings of the communication.

Bit	Name	Access	Description	Default
1-0	number of bits in each character	RW	Select number of bits in each character '00' – 5 bits '01' – 6 bits '10' – 7 bits '11' – 8 bits	11
2	number of generated stop bits	RW	Specify the number of generated stop bits '0' – 1 stop bit '1' – 1.5 stop bits when 5-bit character length selected and 2 bits otherwise Note that the receiver always checks the first stop bit only.	0
3	Parity Enable	RW	'0' – No parity '1' – Parity bit is generated on each outgoing character and is checked on each incoming one.	0
4	Even Parity select	RW	'0' – Odd number of '1' is transmitted and checked in each word (data and parity combined). In other words, if the data has an even number of '1' in it, then the parity bit is '1'. '1' – Even number of '1' is transmitted in each word.	0
5	Stick Parity bit	RW	'0' – Stick Parity disabled '1' - If bits 3 and 4 are logic '1', the parity bit is transmitted and checked as logic '0'. If bit 3 is '1' and bit 4 is '0' then the parity bit is transmitted and checked as '1'.	0
6	Break Control bit	RW	'1' – the serial out is forced into logic '0' (break state). '0' – break is disabled (default)	0
7	Divisor Latch Access bit	RW	'1' – The divisor latches can be accessed '0' – The normal registers are accessed	0

Modem Control Register MCR (0x8806)

The modem control register allows transferring control signals to a modem connected to the UART.

Bit	Name	Access	Description	Default
0	Data Terminal Ready (DTR) signal control	RW	'0' – DTR is '0' '1' – DTR is '1'	0
1	Request To Send (RTS) signal control	RW	'0' – RTS is '0' '1' – RTS is '1'	0
2	Out1	RW	In loopback mode, connected Ring Indicator (RI) signal input.	0
3	Out2	RW	In loopback mode, connected to Data Carrier Detect (DCD) input.	0
4	Loopback mode	RW	'0' – normal operation '1' – loopback mode. When in loopback mode, the Serial Output Signal (STX_PAD_O) is set to logic '1'. The signal of the transmitter shift register is internally connected to the input of the receiver shift register. The following connections are made: DTR ➜ DSR RTS ➜ CTS Out1 ➜ RI Out2 ➜ DCD	0
5	HW_RTS_Stop_TX_En	RW	'0' – normal operation '1' –When RTS = 1 & TX finish sending current character, will stop sending other characters	0
6	HW_TX_Disable	RW	'0' – normal operation '1' –When finish sending current character, will stop sending other characters	0
7	HW_FlowControl_En	RW	'0' – normal operation (default) '1' – enable HW Flow Control, by RTS & CTS Rx part: when RXFIFO content characters higher than RXFIFO interrupt level (0x8804), will pull high RTS, otherwise will pull low RTS Tx part: When finish sending current character, if CTS is pull high, will stop sending other characters	0

Line Status Register LSR (0x8807)

Bit	Name	Access	Description	Default
0	Data Ready (DR) indicator	RO	'0' – No characters in the FIFO '1' – At least one character has been received and is in the FIFO.	
1	Overrun Error (OE) indicator	RO	'1' – If the FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register but the FIFO will remain intact. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. '0' – No overrun state	
2	Parity Error (PE) indicator	RO	'1' – At least one character in FIFO has been detected as having parity error. When this bit is read as '1', all characters in FIFO should be read and dropped. The bit is cleared upon a read from the register. IP210W generates Receiver Line Status interrupt when detecting Parity Error. '0' – No parity error in the current character	
3	Framing Error (FE) indicator	RO	'1' – The received character at the top of the FIFO did not have a valid stop bit. Of course, generally, it might be that all the following data is corrupt. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. '0' – No framing error in the current character	
4	Break Interrupt (BI) indicator	RO	'1' –A break condition has been reached in the current character. The break occurs when the line is held in logic 0 for a time of one character (start bit + data + parity + stop bit). In that case, one zero character enters the FIFO and the UART waits for a valid start bit to receive next character. The bit is cleared upon reading from the register. Generates Receiver Line Status interrupt. '0' – No break condition in the current character	
5	Transmit FIFO is empty	RO	'1' – The transmitter FIFO is empty. Generates Transmitter Holding Register Empty interrupt. The bit is cleared when data is being been written to the transmitter FIFO. '0' – Otherwise	
6	Transmitter Empty indicator	RO	'1' – Both the transmitter FIFO and transmitter shift register are empty. The bit is cleared when data is being been written to the transmitter FIFO. '0' – Otherwise	
7	Reserved	RO	Reserved	

Modem Status Register MSR (0x8808)
The register displays the current state of the modem control lines.

Bit	Name	Access	Description	Default
3-0	Reserved	RO	Reserved	
4	CTS input	RO	CTS input or equals to RTS in loopback mode.	
5	DSR input	RO	DSR input or equals to DTR in loopback mode.	
6	RI input	RO	RI input or equals to Out1 in loopback mode.	
7	DCD input	RO	DCD input or equals to Out2 in loopback mode.	

UART_TX FIFO Status (0x8809)

Bit	Name	Access	Description	Default
7-0	TX_FIFO_CNT	RO	TX FIFO available space counter	

UART_RX FIFO Status (0x880a)

Bit	Name	Access	Description	Default
7-0	RX_FIFO_CNT	RO	RX FIFO occupied space counter	

UART_Clock Divisor Registers
In addition, there are 2 Clock Divisor registers that together form one 16-bit.
The registers can be accessed when the 7th (DLAB) bit of the Line Control Register is set to '1'. At this time the above registers UART_Receiver Buffe, UART_Transmit Buffer & UART_Interrupt Enable can't be accessed.

(Input Clock Speed)/(Divisor Latch value) = 16 x the communication baud rate

UART_Clock Divisor Registers_L (0x880b)

Bit	Name	Access	Description	Default
7-0	UART_Clock Divisor Registers_L	RW	UART_Clock Divisor Registers_low byte.	00 h

UART_Clock Divisor Registers_H (0x880c)

Bit	Name	Access	Description	Default
7-0	UART_Clock Divisor Registers_H	RW	UART_Clock Divisor Registers_high byte.	00 h

7 Crystal & OSC(X1) Specifications

7.1 Crystal Specifications

Item	Parameter	Range
1	Nominal Frequency	25.000 MHz
2	Oscillation Mode	Fundamental Mode
3	Frequency Tolerance at 25℃	+/- 50 ppm
4	Temperature Characteristics	+/- 50 ppm
5	Operating Temperature Range	-10℃ ~ +70℃
6	Equivalent Series Resistance	40 ohm Max.
7	Drive Level	100 μW
8	Load Capacitance	20 pF
9	Shunt Capacitance	7 pF Max
10	Insulation Resistance	Mega ohm Min./DC 100V
11	Aging Rate A Year	+/- 5 ppm/year

7.2 OSC(X1) Clock Input Specifications

Item	Parameter	Range
1	Frequency	25.000 MHz
2	Duty cycle	40~60%
3	Rise time	Max: 4 ns
4	Fall time	Max: 4 ns
5	Max voltage	3.63 V
6	Vih	2 V
7	Vil	1 V
8	Jitter	300 ps

8   Electrical Characteristics

8.1   Absolute Maximum Rating

Symbol	Conditions	Minimum	Typical	Maximum
Supply Voltage		3.0 V	3.3V	3.6V
Junction Temperature		0°C		125°C
Storage Temp		-55°C		125°C

8.2   Power Dissipation

Operating Condition	Power consumption (W)
active	0.460W

8.3   DC Characteristic

8.3.1   Operating Condition

Symbol	Conditions	Minimum	Typical	Maximum
Vcc3.3	3.3V Supply voltage	3.0 V	3.3V	3.6V
Vcc2.5	2.5V Supply voltage	2.25V	2.5V	2.75V
TA	Operating Temperature	0°C		70°C

8.3.2   Supply Voltage

Symbol	Specific Name	Condition	Min	Max
V_{IH}	Input High Vol.		0.57*PVDD	
V_{IL}	Input Low Vol.			0.35*PVDD
V_{OH}	Output High Vol.		0.9*PVDD	
V_{OL}	Output Low Vol.			0.1*PVDD
V_{IH}	X1 Input High Voltage	PVDD=3.3V	3.15 V	
V_{IL}	X1 Input Low Voltage	PVDD=3.3V		0.125 V
V_{RST}	RESETB Threshold Voltage		0.57*PVDD	

8.4 AC Timing

8.4.1 Power On Sequence and Reset Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{clk_lead}	X1 clock valid period before reset released	1	-	-	ms
T_{rst}	Reset period	1	-	-	ms
$T_{clk\_out\_rdy}$	CLK25 clock out ready after reset released (Pin 17 output)	0	-	20	ns
T_{diff}	Time difference between VCC3.3 and VCC2.5	-2		2	ms
T_{pwr_lead}	All power source ready before reset released	1			ms
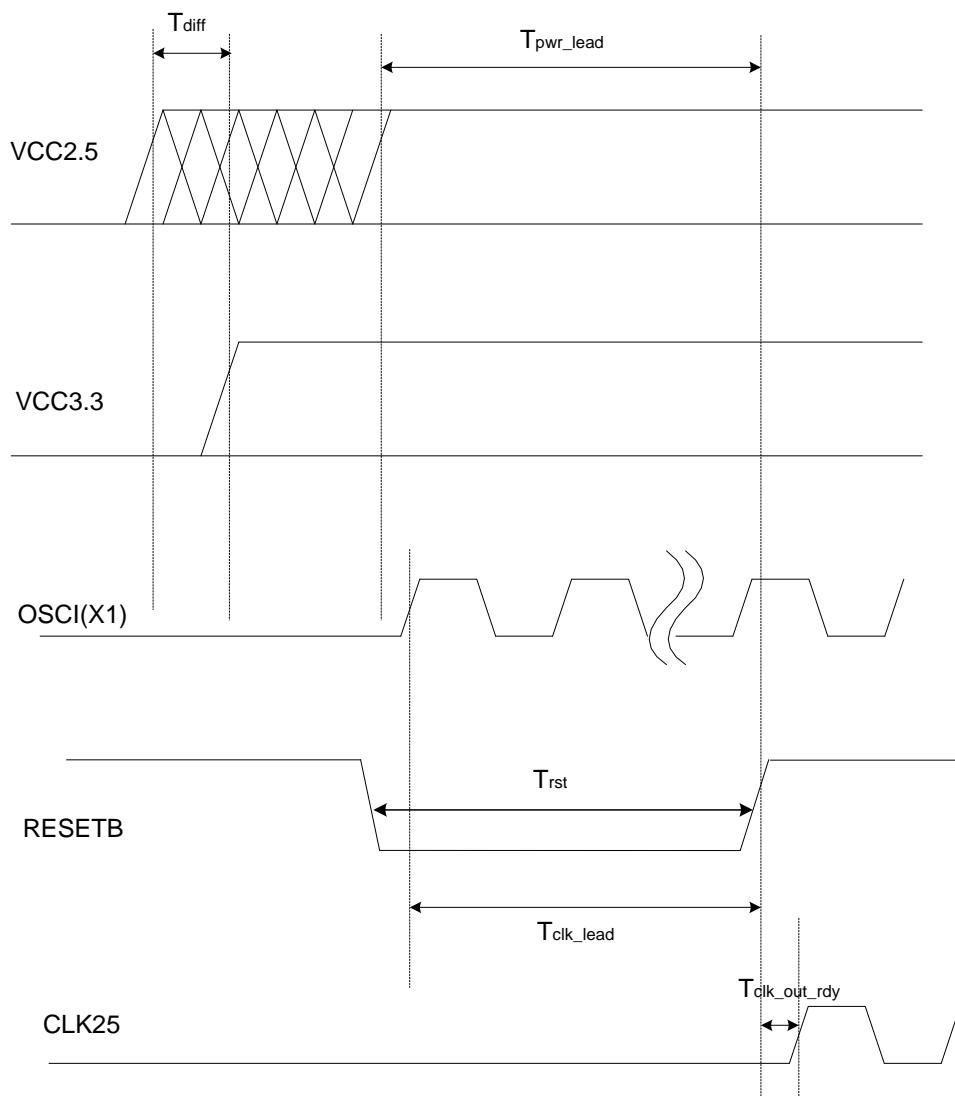
8.4.2 PHY Mode MII Timing

a. Transmit Timing Requirements

Symbol	Description	Min.	Typ.	Max.	Unit
T_{TxClk}	Transmit clock period 100M MII	-	40	-	ns
T_{TxClk}	Transmit clock period 10M MII	-	400	-	ns
T_{sTxClk}	PMTXEN0, PMTXD0 to PMTXCLK0 setup time	10	-	-	ns
T_{hTxClk}	PMTXEN0, PMTXD0 to PMTXCLK0 hold time	5	-	-	ns
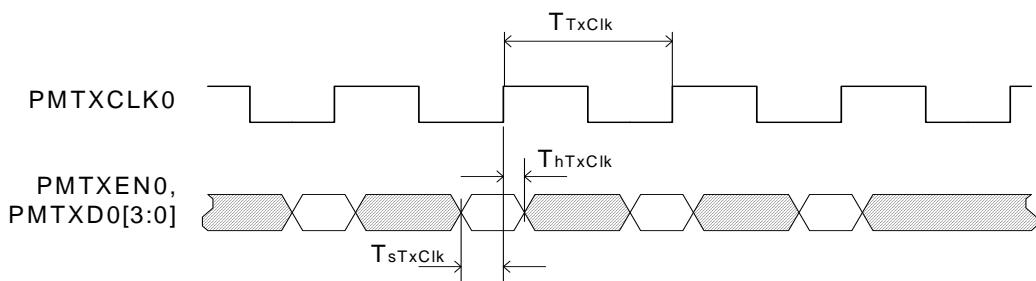
b. Receive Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{RxClk}	Receive clock period 100M MII	-	40	-	ns
T_{RxClk}	Receive clock period 10M MII	-	400	-	ns
T_{dRxClk}	PMRXCLK0 falling edge to PMRXDV0, PMRXD0	1	-	4	ns
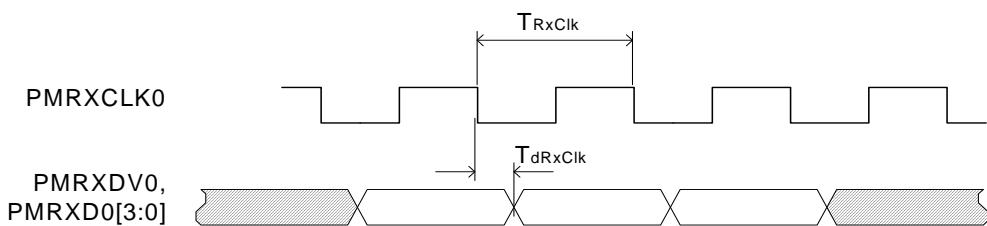
8.4.3 MAC Mode MII Timing

a. Receive Timing Requirements

Symbol	Description	Min.	Typ.	Max.	Unit
T_{RxClk}	Receive clock period 100M MII	-	40	-	ns
T_{RxClk}	Receive clock period 10M MII	-	400	-	ns
T_{sRxClk}	MMRXDV0, MMRXD0 to MMRXCLK0 setup time	10	-	-	ns
T_{hRxClk}	MMRXDV0, MMRXD0 to MMRXCLK0 hold time	5	-	-	ns
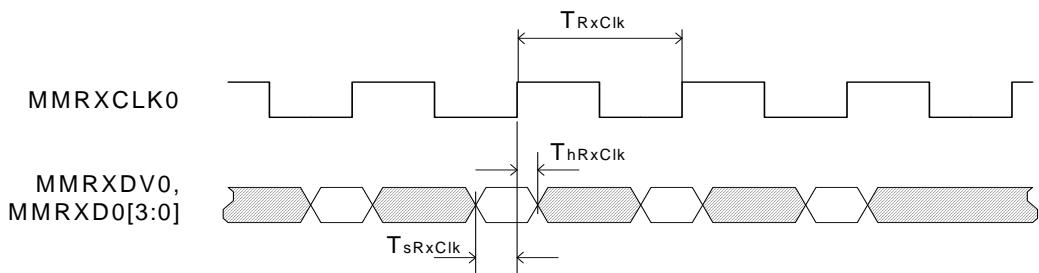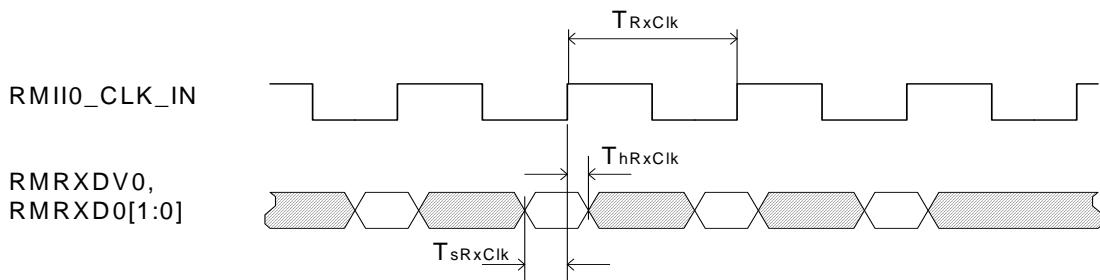
b. Transmit Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{TxClk}	Transmit clock period 100M MII	-	40	-	ns
T_{TxClk}	Transmit clock period 10M MII	-	400	-	ns
T_{dTxClk}	MMTXCLK0 rising edge to MMTXEN0, MMTXD0	6	-	22	ns
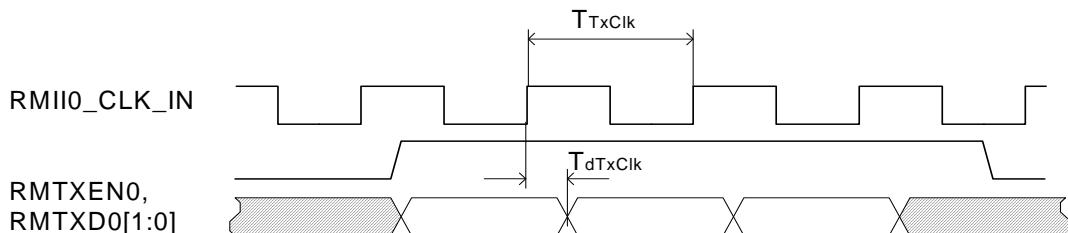
8.4.5 SMI Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{ch}	MDC High Time	17	-	1088	ns
T_{cl}	MDC Low Time	17	-	1088	ns
T_{cm}	MDC period	34	-	2176	ns
T_{mdb}	MDIO output before MDC	15	-	18	ns
T_{md}	MDIO output delay	T_{cm} -18	-	T_{cm} -15	ns
T_{ms}	MDIO setup time	18	-	-	ns
T_{mh}	MDIO hold time	0	-	-	ns

M D C

T_{ms} T_{mh}

M D I O

Input Cycle

M D C

T_{cl} T_{ch} T_{md}

T_{cm} T_{mdb}

M D I O

Output Cycle
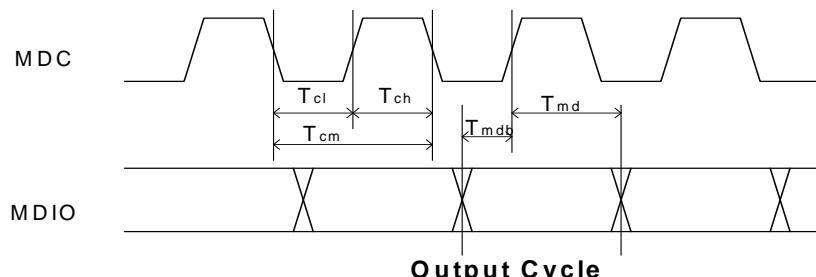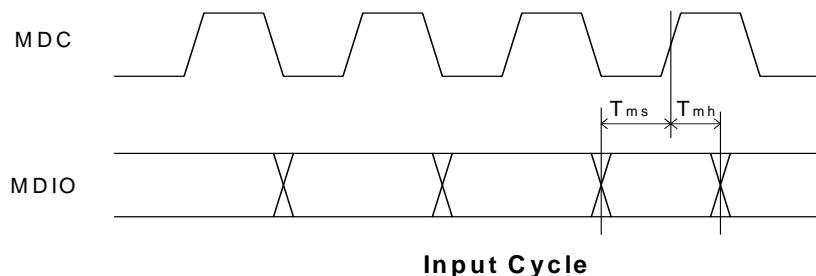
8.4.6 External Flash Timing

Flash Read Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{frw}	Flash read assert width	-	$(1+n)T$	-	ns
T_{frav}	Address valid to read assert	-1	-	0	ns
T_{fraiv}	Read de-assert to address de-assert	-2	-	0	ns
T_{frdv}	Read assert to data valid			$4+nT$	ns
T_{frdiv}	Read de-assert to data de-assert	0			ns

Note: 1. T is system clock cycle time (16.9 ns)
2. n is flash wait state number (Default is 0x7, cpu control register 0x8002)
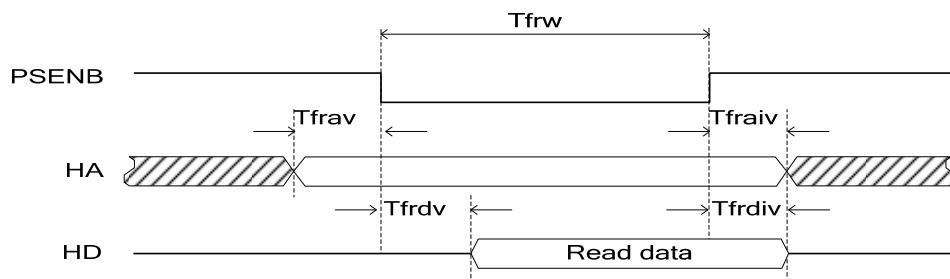
Flash Write Timing

Symbol	Description	Min.	Typ.	Max.	Unit
T_{fww}	Receive clock period	-	8T	-	ns
T_{fbv}	Address/Data bus valid to write assert	8	-	16	ns
T_{fbiv}	Write de-assert to address/data bus de-assert	8	-	16	ns

Note: 1. T is system clock cycle time (16.9 ns)
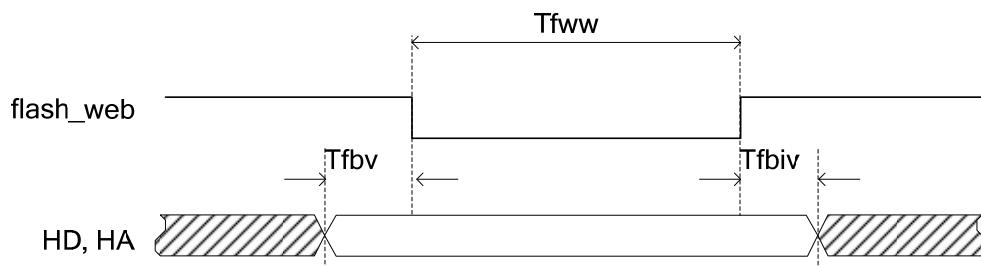
8.4.7 EEPROM Timing

a.

Symbol	Description	Min.	Typ.	Max.	Unit
T_{SCL}	Receive clock period	-	20480	-	ns
T_{sSCL}	SDA to SCL setup time	20	-	-	ns
T_{hSCL}	SDA to SCL hold time	20	-	-	ns

Read data cycle
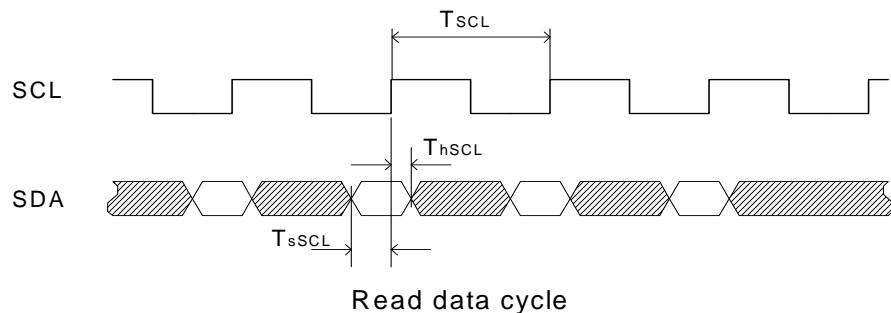
b.

Symbol	Description	Min.	Typ.	Max.	Unit
T_{SCL}	Transmit clock period	-	20480	-	ns
T_{dSCL}	SCL falling edge to SDA	-	-	5200	ns

Comand cycle
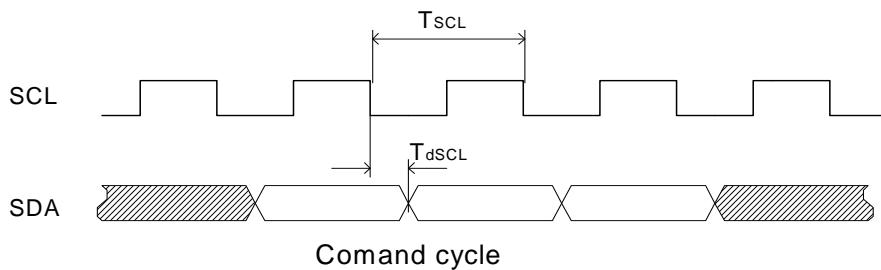
8.5 Thermal Data

Theta Ja	Theta Jc	Conditions	Units
	--	2 Layer PCB	^{o}C/W

9   Order Information

Part No.	Package	Notice
IP210W LF	128-PIN PQFP	Lead free

10 Package Detail

IP210W-128 PIN (PQFP-128(14X20))
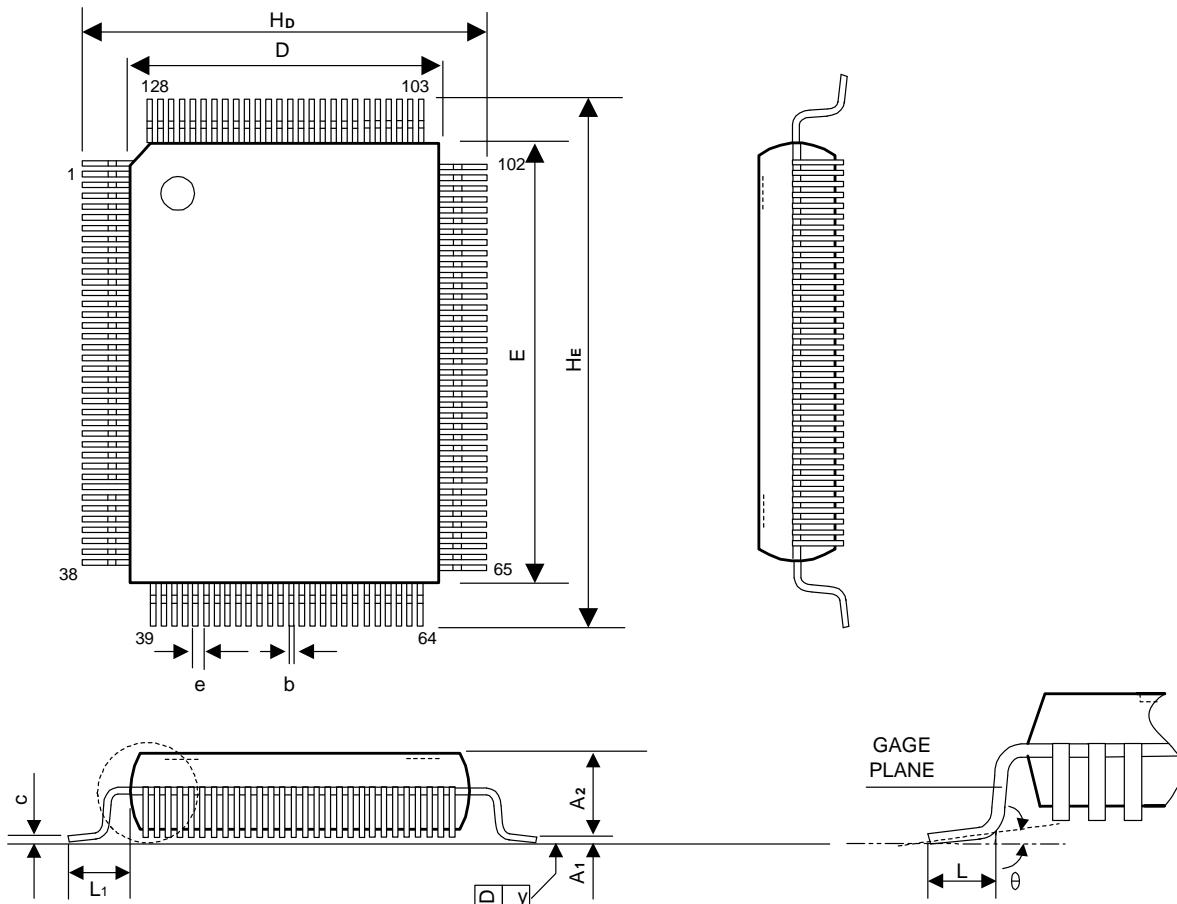
Symbol	Dimensions In Inches			Dimensions In mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A1	0.010	0.014	0.018	0.25	0.35	0.45
A2	0.107	0.112	0.117	2.73	2.85	2.97
b	0.007	0.009	0.011	0.17	0.22	0.27
c	0.004	0.006	0.008	0.09	0.15	0.20
HD	0.669	0.677	0.685	17.00	17.20	17.40
D	0.547	0.551	0.555	13.90	14.00	14.10
HE	0.906	0.913	0.921	23.00	23.20	23.40
E	0.783	0.787	0.791	19.90	20.00	20.10
e	-	0.020	-	-	0.50	-
L	0.025	0.035	0.041	0.65	0.88	1.03
L1	-	0.063	-	-	1.60	-
y	-	-	0.004	-	-	0.10
θ	0°	-	12°	0°	-	12°

Note:
1. Dimension D & E do not include mold protrusion.
2. Dimension B does not include dambar protrusion. Total in excess of the B dimension at maximum material condition.
   Dambar cannot be located on the lower radius of the foot.

IC Plus Corp.

Headquarters	**Sales Office**
10F, No.47, Lane 2, Kwang-Fu Road, Sec. 2,	4F, No. 106, Hsin-Tai-Wu Road, Sec.1,
Hsin-Chu City, Taiwan 300, R.O.C.	Hsi-Chih, Taipei Hsien, Taiwan 221, R.O.C.
TEL : 886-3-575-0275 FAX : 886-3-575-0475	TEL : 886-2-2696-1669 FAX : 886-2-2696-2220
Website : www.icplus.com.tw	

April 26, 2011

IP210W-DS-R05

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for ic plus *manufacturer:*

Other Similar products are found below :

IP179N IP1725LF IP179H IP1717BLF IP210WLF IP1717LLF IP1001MLF