ZEN8105

Preliminary

Enhanced 8051 Microcontroller with 10-bit ADC

Features

- 8051 compatible Pipe-lined instruction based on the single-chip 8-bit micro-controller
- Flash ROM: 16K Bytes
- RAM: internal 256 Bytes and external 512 Bytes
- Operation voltage:
  f_{OSC}=400k - 12MHz, V_{DD}=4.0V – 5.5V
- Oscillator (Code option):
  - Ceramic resonator: 400K - 12MHz
  - Internal RC oscillator: 12MHz
- 29 CMOS general purpose I/O ports
- 4 open-drain type I/O available
- Built-in pull-high resistor for I/O
- Three 16-bit timer / counters: T0, T1 & T2
- Powerful interrupt sources:
  - Timer0, Timer1, Timer2
  - External interrupt 0~3
  - External interrupt 4: 8 inputs
  - ADC, EUART,
- Enhanced UART
- 10-bit 8 channel Analog Digital Converter (ADC) with built-in compare function
- Built-in low voltage reset function (enabled by code option)
  - LVR voltage level: 3.1V
- CPU Machine cycle:
  - 1 oscillator clock
- Built-in Watch Dog Timer (WDT)
- Warm-up timer for power-on reset
- Support Low power operation modes:
  - IDLE Mode
  - Power-Down Mode
- Flash type
- Package:
  - 20-pin plastic DIP Package

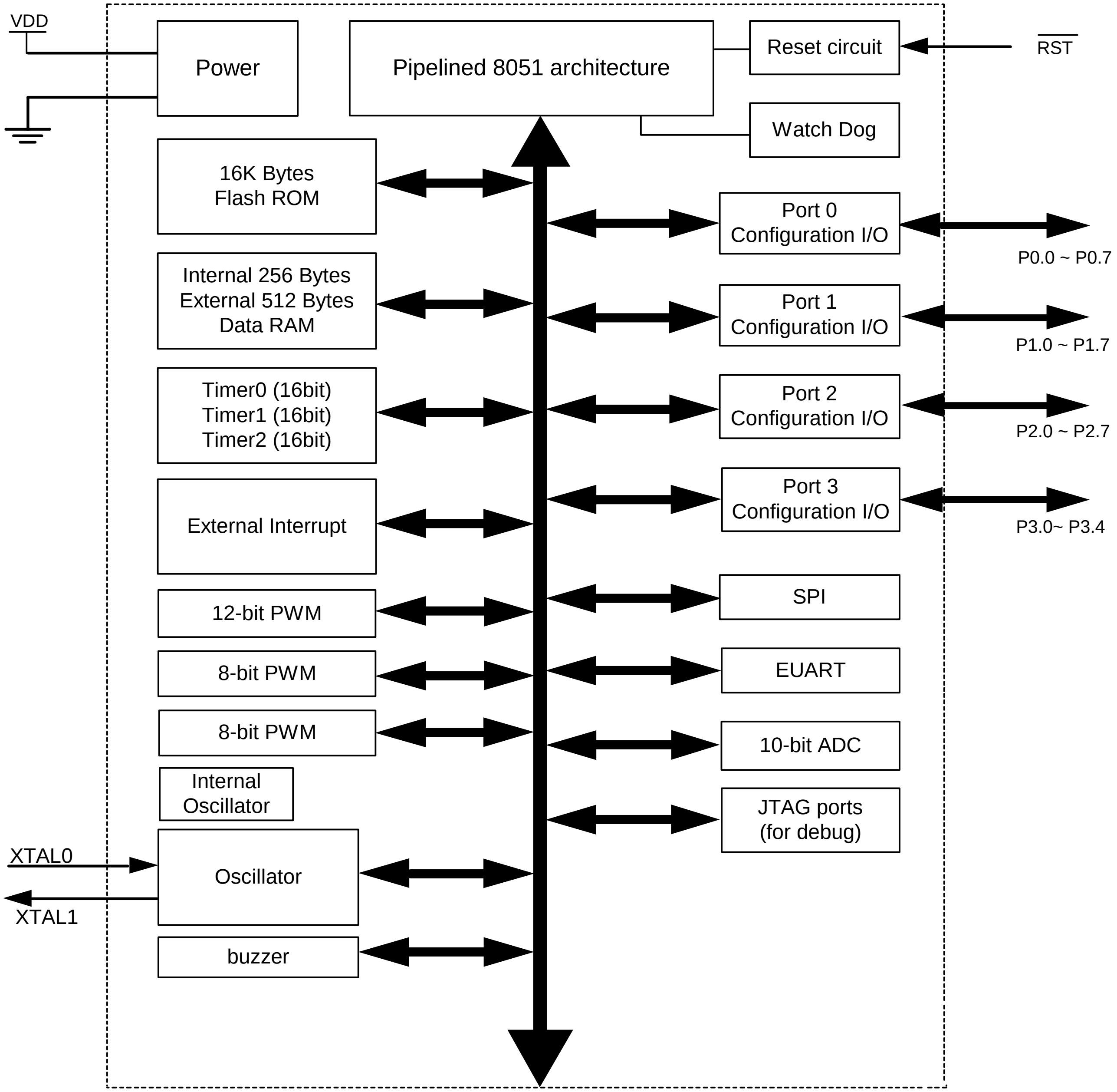
General Description

The ZEN8105 is a fast 8051 compatible micro-controller with a redesigned CPU of no wasted clock and memory cycles. Typically, it will be faster than and exhibit better performance than the traditional 8051 at the same oscillator frequency.

The ZEN8105 retains most features of the standard 8051. These features include internal 256 bytes RAM and two 16-bit timer/counters. In addition, the ZEN8105 provides external 512 bytes RAM and another 16-bit timer/counter compatible with 8052. It contains a 16K Bytes Flash memory block for program and data.

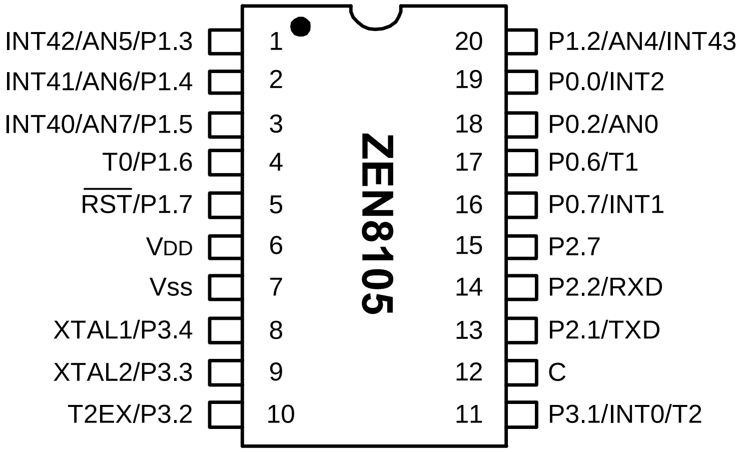
Some standard serial communication modes such as EUART and SPI are supported in ZEN8105. Also the ADC together with built-in digital compare function and PWM timers are incorporated in ZEN8105.
It also provides the following standard features: on-chip watchdog timer, low voltage reset function. It provides two power saving modes to reduce the power consumption.

Block Diagram



PIN Configuration

1) 20 PIN



Pin Functions

Pin No.	Pin Name	Default function
1	INT42 / AN5 / P1.3	P1.3
2	INT41 / AN6 / P1.4	P1.4
3	INT40 /AN7 / P1.5	P1.5
4	T0 / P1.6	P1.6
5	$\overline{\text{RST}}$/P1.7	Reset Pin or P1.7, selected by code option
6	V_{DD}	-----
7	Vss	-----
8	XTAL0 / P3.4	I/O Port or XTAL0 pin, selected by code option
9	XTAL1 / P3.3	I/O Port or XTAL1 pin, selected by code option
10	T2EX / P3.2	P3.2
11	T2 / INT0 / P3.1	P3.1
12	C	-----
13	TXD / P2.1	P2.1
14	RXD / P2.2	P2.2
15	P2.7	P2.7
16	INT1 / P0.7	P0.7
17	T1 / P0.6	P0.6
*18	AN0 / P0.2	P0.2
19	INT3 / P0.1	P0.1
20	INT43/AN4/P1.2	P1.2

*: These pins can be configured as N-channel open-drain, but voltage provided for this pin can't exceed VDD+0.3V.
Total 20 pins.

Note: *The out most pin function has the highest priority, and the inner most pin function has the lowest priority (Refer to* ***Pin Configuration Diagram****). This means when one pin is occupied by ahigher priority function (if enabled) cannot be used as the lower priority functional pin, even when the lower priority function is also enabled. Only until the higher priority function is disabled by software, can the corresponding pin be released for the lower priority function use.*

Pin Description

Pin Name	I/O	Description
PORT		
P0.0-P0.7	I/O	8-bit bi-directional I/O port
P1.0-P1.7	I/O	8-bit bi-directional I/O port
P2.0-P2.7	I/O	8-bit bi-directional I/O port
P3.0-P3.4	I/O	5-bit bi-directional I/O port
Timer		
T0	I/O	Timer0 external input or Timer0 compare output
T1	I/O	Timer1 external input or Timer1 compare output
T2	I/O	Timer2 external input/ Baudrate clock output
T2EX	I	The external clock input pin for the capture timer
EUART		
RXD	I/O	EUART data input
TXD	O	EUART data output.
ADC		
AN0 – AN7	I	ADC input channel
Interrupt & Reset & Clock & Power		
INT0 – INT3	I	External interrupt 0~3
INT40 – INT47	I	External interrupt 40~47
$\overline{\text{RST}}$	I	A low on this pin for 10us longer will reset the device. An internal diffused resistor to VDD permits a power-on reset using only an external capacitor to GND.
XTAL0	I	Oscillator input
XTAL1	O	Oscillator output
Vss	P	Ground
V_{DD}	P	Power supply (3.0 ~ 5.5V)
Capacitor		
C		Capacitance pin for regulating the power supply, at least 47uF recommended.
Programmer		

TDO (P1.2)	O	Debug interface: Test data out
TMS (P1.3)	I	Debug interface: Test mode select
TDI (P1.4)	I	Debug interface: Test data in
TCK (P1.5)	I	Debug interface: Test clock in

Note:

When P1.2-1.5 used as debug interface, other functions of P1.2-1.5 are blocked.

SFR Mapping

The ZEN8105 provides 256 bytes of internal RAM to contain general-purpose data memory and Special Function Register (SFR). The SFR of the ZEN8105 fall into the following categories:

Cpu core registers: ACC, B, PSW, SP, DPL, DPH

Enhanced C51 core registers: C, DPL1, DPH1, INSCON, XPAGE

Power and clock control registers: PCON, SUSLO

Flash registers: IB_CLK0, IB_CLK1, IB_OFFSET, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5

Data Memory registers: XPAGE

Hardware Watchdog Timer registers: RSTSTAT

System Clock control registers: CLKCON

Interrupt system registers: IEN0, IEN1, IENC, IPH0, IPL0, IPH1, IPL1, EXF0, EXF1

I/O port registers: P0, P1, P2, P3, P0CR, P1CR, P2CR, P3CR, P0PCR, P1PCR, P2PCR, P3PCR, P0SS

Timer registers: TCON, TMOD, TH0, TH1, TL0, TL1, T2CON, T2MOD, TH2, TL2, RCAP2L, RCAP2H, TCOME

EUART registers: SCON, SBUF, SADEN, SADDR, PCON

ADC registers: ADCON, ADT, ADCH, ADDL, ADDH

Cpu Core SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6 Bit5 Bit4				Bit3 Bit2		Bit1	Bit0
ACC	E0h	Accumulator	00000000	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
B	F0h	B Register	00000000	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
C	F1h	C Register	00000000	C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
PSW	D0h	Program Status Word	00000000	CY	AC	F0	RS1	RS0	OV	F1	P
SP	81h	Stack Pointer	00000111	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
DPL	82h	Data Pointer1 Low byte	00000000	DPL0.7	DPL0.6	DPL0.5	DPL0.4	DPL0.3	DPL0.2	DPL0.1	DPL0.0
DPH	83h	Data Pointer1 High byte	00000000	DPH0.7	DPH0.6	DPH0.5	DPH0.4	DPH0.3	DPH0.2	DPH0.1	DPH0.0
DPL1	84h	Data Pointer 2 Low byte	00000000	DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0
DPH1	85h	Data Pointer 2 High byte	00000000	DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0
INSCON	86h	Data pointer select	----00-0	-	-	-	-	DIV	MUL	-	DPS

Power and clock control SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6		Bit5 Bit4		Bit3 Bit2		Bit1 Bit0	
PCON	87h	Power Control	00--0000	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
SUSLO	8Eh	Suspend Mode Control	00000000	SUSL.7	SUSL.6	SUSL.5	SUSL.4	SUSL.3	SUSL.2	SUSL.1	SUSL.0

Flash control SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6 Bit5			Bit4	Bit3 Bit2		Bit1 Bit0	
IB_CLK0	F9h	Flash programming clock register 0	00000000	IB_CLK 0.7	IB_CLK 0.6	IB_CLK 0.5	IB_CLK 0.4	IB_CLK 0.3	IB_CLK 0.2	IB_CLK 0.1	IB_CLK 0.0
IB_CLK1	FAh	Flash programming clock register 1	00000000	IB_CLK 1.7	IB_CLK 1.6	IB_CLK 1.5	IB_CLK 1.4	IB_CLK 1.3	IB_CLK 1.2	IB_CLK 1.1	IB_CLK 1.0
IB_OFF SET	FBh	Low byte offset of flash memory for programming	00000000	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
IB_DATA	FCh	Data Register for programming flash memory	00000000	IB_ DATA.7	IB_ DATA.6	IB_ DATA.5	IB_ DATA.4	IB_ DATA.3	IB_ DATA.2	IB_ DATA.1	IB_ DATA.0
IB_CON1	F2h	Flash Memory Control Register 1	00000000	IB_ CON1.7	IB_ CON1.6	IB_ CON1.5	IB_ CON1.4	IB_ CON1.3	IB_ CON1.2	IB_ CON1.1	IB_ CON1.0
IB_CON2	F3h	Flash Memory Control Register 2	---00000	-	-	-	IB_CON 2.4	IB_CON 2.3	IB_CON 2.2	IB_CON 2.1	IB_CON 2.0
IB_CON3	F4h	Flash Memory Control Register 3	----0000	-	-	-	-	IB_CON 3.3	IB_CON 3.2	IB_CON 3.1	IB_CON 3.0
IB_CON4	F5h	Flash Memory Control Register 4	----0000	-	-	-	-	IB_CON 4.3	IB_CON 4.2	IB_CON 4.1	IB_CON 4.0
IB_CON5	F6h	Flash Memory Control Register 5	----0000	-	-	-	-	IB_CON 5.3	IB_CON 5.2	IB_CON 5.1	IB_CON 5.0
XPAGE	F7h	Memory Page	00000000	XPAGE.	XPAGE.	XPAGE.	XPAGE.	XPAGE.	XPAGE.	XPAGE.	XPAGE.

Data MPAGE SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7	Bit6 Bit5 Bit4 Bit3 Bit2 Bit1						Bit0
XPAGE	F7h	Memory Page	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0

WDT SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6 Bit5 Bit4				Bit3 Bit2 Bit1 Bit0			
RSTSTAT	B1h	Watchdog Timer Control	0-000000*	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0

Note: RSTSTAT watchdog reset value is determined by different RESET.

CLKCON SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6		Bit5 Bit4		Bit3	Bit2 Bit1 Bit0		
CLKCON	B2h	System Clock Select	-00-----	-	CLKPS1	CLKPS0	-	-	-	-	-

Interrupt SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6		Bit5 Bit4		Bit3	Bit2 Bit1	Bit0	
IEN0	A8h	External Interrupt Enable Control 1	00000000	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
IEN1	A9h	External Interrupt Enable Control 2	--0-0000	-	-	EPWM	-	EX4	EX3	EX2	ESPI
IENC	BAh	External interrupt channel enable	00000000	EXS47	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
IPH0	B4h	Interrupt Priority Control High 0	-0000000	-	PADCH	PT2H	PUH	PT1H	PX1H	PT0H	PX0H
IPL0	B8h	Interrupt Priority Control Low 0	-0000000	-	PADCL	PT2L	PUL	PT1L	PX1L	PT0L	PX0L
IPH1	B5h	Interrupt Priority Control High 1	--0-0000	-	-	PPWMH	-	PX4H	PX3H	PX2H	PSPIH
IPL1	B9h	Interrupt Priority Control Low 1	--0-0000	-	-	PPWML	-	PX4L	PX3L	PX2L	PSPIL
EXF0	E8h	External Interrupt 0 flag	00000000	IT4.1	IT4.0	IT3.1	IT3.0	IT2.1	IT2.0	IE3	IE2
EXF1	D8h	External Interrupt 1 flag	00000000	IF47	IF46	IF45	IF44	IF43	IF42	IF41	IF40

Port SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6		Bit5 Bit4		Bit3 Bit2		Bit1 Bit0	
P0	80h	8-bit Port 0	00000000	P0.7	P0.6	-	-	-	P0.2	-	P0.0
P1	90h	8-bit Port 1	00000000	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	-	-
P2	A0h	8-bit Port 2	00000000	P2.7	-	-	-	-	P2.2	P2.1	-
P3	B0h	5-bit Port 3	---00000	-	-	-	P3.4	P3.3	P3.2	P3.1	-
P0CR	E1h	Port0 input/output direction control	00000000	P0CR.7	P0CR.6	RES	RES	RES	*P0CR.2	RES	P0CR.0
P1CR	E2h	Port1 input/output direction control	00000000	P1CR.7	P1CR.6	P1CR.5	P1CR.4	P1CR.3	P1CR.2	RES	RES
P2CR	E3h	Port2 input/output direction control	00000000	P2CR.7	RES	RES	RES	RES	P2CR.2	RES	RES
P3CR	E4h	Port3 input/output direction control	---00000	-	-	-	P3CR.4	P3CR.3	P3CR.2	P3CR.1	RES
P0PCR	E9h	Internal pull-high enable for Port0	00000000	P0PCR.7	P0PCR.6	-	-	-	P0PCR.2	-	P0PCR.0
P1PCR	EAh	Internal pull-high enable for Port1	00000000	P1PCR.7	P1PCR.6	P1PCR.5	P1PCR.4	P1PCR.3	P1PCR.2	-	-
P2PCR	EBh	Internal pull-high enable for Port2	00000000	P2PCR.7	P2PCR.6	-	-	-	P2PCR.2	P2PCR.1	-
P3PCR	ECh	Internal pull-high enable for Port3	---00000	-	-	-	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	-
P0SS	EFh	Output mode select	--0000--	-	-	-	-	-	P02OS	-	-

Timer SFRs

Mnem A	dd	Name	POR/WDT/LVR Reset Value	Bit7 Bit6		Bit5	Bit4	Bit3 Bit2		Bit1	Bit0
TCON	88h	Timer/Counter 0 and 1 Control	00000000	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	Timer/Counter 0 and 1 Modes	00000000	GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
TL0	8Ah	Timer/Counter 0 Low Byte	00000000	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	8Ch	Timer/Counter 0 High Byte	00000000	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1	8Bh	Timer/Counter 1 Low Byte	00000000	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.1
TH1	8Dh	Timer/Counter 1 High Byte	00000000	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.1
T2CON	C8h	Timer/Counter 2 Control	00000000	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
T2MOD	C9h	Timer/Counter 2	------00	-	-	-	-	-	-	T2OE	DCEN

		Mode									
RCAP2L	CAh	Timer/Counter 2 Reload/Capture Low Byte	00000000	RCAP 2L.7	RCAP 2L.6	RCAP 2L.5	RCAP 2L.4	RCAP 2L.3	RCAP 2L.2	RCAP 2L.1	RCAP 2L.0
RCAP2H	CBh	Timer/Counter 2 Reload/Capture High Byte	00000000	RCAP 2H.7	RCAP 2H.6	RCAP 2H.5	RCAP 2H.4	RCAP 2H.3	RCAP 2H.2	RCAP 2H.1	RCAP 2H.0
TL2	CCh	Timer/Counter 2 Low Byte	00000000	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.2	TL2.2
TH2	CDh	Timer/Counter 2 High Byte	00000000	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.2	TH2.2
TCOME	CEh	Timer0/1Compare Function Enable	------00	-	-	-	-	-	-	TC1	TC0

UART SFRs

Mnem A	**dd**	**Name**	**POR/WDT/LV R Reset Value**	**Bit7 Bit6**		**Bit5**	**Bit4**	**Bit3 Bit2**		**Bit1 Bit0**	
SCON	98h	Serial Control	00000000	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
SBUF	99h	Serial Data Buffer	00000000	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SADEN	9Bh	Slave Address Mask	00000000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
SADDR	9Ah	Slave Address	00000000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
PCON	87h	Power & serial Control	00—0000	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL

ADC SFRs

Mnem A	**dd**	**Name**	**POR/WDT/LVR Reset Value**	**Bit7 Bit6 Bit5**			**Bit4**	**Bit3 Bit2**		**Bit1**	**Bit0**
ADCON	93h	ADC Control	000-0000	ADON	ADCIF	EC	-	SCH2	SCH1	SCH0	GO/$\overline{\text{DONE}}$
ADT	94h	ADC Time select	000-0000	TADC2	TADC1	TADC0	-	TS3	TS2-	TS1	TS0
ADCH	95h	ADC Configuration	00000000	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADDL	96h	ADC Data Low Byte	------00	-	-	-	-	-	-	A1	A0
ADDH	97h	ADC Data High Byte	00000000	A9	A8	A7	A6	A5	A4	A3	A2

- - - -

SFR mapping figure

	Bit Addressable	**Non Bit Addressable**							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	SPSTA	IB_CLK0	IB_CLK1	IB_OFFSET	IB_DATA				FFh
F0h	B	C	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7h
E8h	EXF0	P0PCR	P1PCR	P2PCR	P3PCR			P0SS	Efh
E0h	ACC	P0CR	P1CR	P2CR	P3CR				E7h
D8h	EXF1								DFh
D0h	PSW								D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TCOME		CFh
C0h									C7h
B8h	IPL0	IPL1	IENC						BFh
B0h	P3	RSTSTAT	CLKCON		IPH0	IPH1			B7h
A8h	IEN0	IEN1							Afh
A0h	P2								A7h
98h	SCON	SBUF	SADDR	SADEN					9Fh
90h	P1			ADCON	ADT	ADCH	ADDL	ADDH	97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	SUSLO		8Fh
80h	P0	SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87h
	0/8 1/9		*2/A*	*3/B*	*4/C 5/D*		*6/E*	*7/F*	

Note:
The unused addresses of SFR are not available. Please don’t operation these address.

Function Description

1 CPU

1.1 Instruction Extension

ZEN8105 has modified 'MUL' and 'DIV' instructions. These instructions support 16bit operand. A new register – the C register is applied to hold the upper part of the operand/result.

After reset, the CPU is in standard mode, which means that the 'MUL' and 'DIV' instructions are operating like the standard 8051 instructions. To enable the 16bit mode operation, the corresponding enable bit in the INSCON register must be set.

	Operation		Result		
			A	B	C
MUL	INSCON.2 = 0; 8 bit mode	(A)*(B)	Low Byte	High Byte	---
	INSCON.2 = 1; 16 bit mode	(C A)* (B)	Low Byte	Middle Byte	High Byte
DIV	INSCON.3 = 0; 8 bit mode	(A) / (B)	Quotient Low Byte	Remainder	---
	INSCON.3 = 1; 16 bit mode	(C A) / (B)	Quotient Low Byte	Remainder	Quotient High Byte

1.2 Dual Data Pointer

Data memory moves can be accelerated by using two data pointers. The standard data pointer is called DPTR and the new data pointer is called DPTR1. The DPS bit in INSTCON register is used to choose the active pointer. The user can switch data pointers by toggling the DPS bit. And all DPTR-related instructions will use the currently selected data pointer.

Data Pointer Select Register

86H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INSCON	-	-	-	-	DIV	MUL	-	DPS
R/W	-	-	-	-	R/W	R/W	-	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	0	0	-	0

Bit Number	Bit Mnemonic	Description
3	**DIV**	**16 bit / 8 bit Divide Selector** 0: 8 Bit Divide 1: 16 Bit Divide
2	**MUL**	**MUL– 16 bit / 8 bit Multiply Selector** 0: 8 Bit Multiply 1: 16 Bit Multiply
0	**DPS**	**DPS – Data Pointer Selector** 0: Data pointer 1: Data pointer1

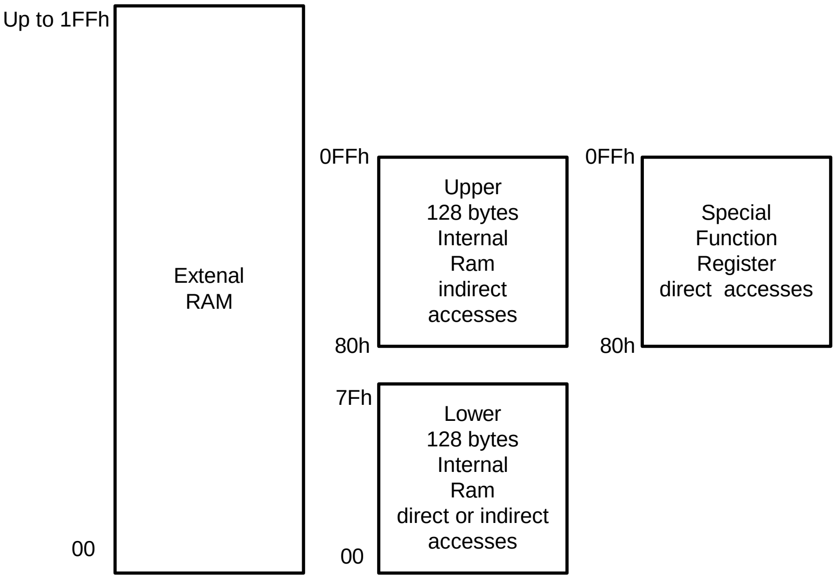
2 RAM

The ZEN8105 provides additional Bytes of RAM space for increased data parameter handling, high level language usage. ZEN8105 devices have expanded RAM in external data space configurable up to 512 bytes.
The ZEN8105has internal data memory that is mapped into four separate segments.

The four segments are:

1.The Lower 128 bytes of RAM (addresses 00h to 7Fh) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80h to FFh) are indirectly addressable only.
3. The Special Function Registers (SFR, addresses 80h to FFh) are directly addressable only.
4. The expanded RAM bytes are indirectly accessed by MOVX instructions.

The Upper 128 bytes occupy the same address space as SFR, but they are physically separate from SFR space. When an instruction accesses an internal location above address 7Fh, the CPU can distinguish whether to access the upper 128 bytes data RAM or to access SFR by different addressing mode of the instruction.
Note the unused address is unavailable in SFR.
The Internal and External RAM configuration is shown as below:


Up to 1FFh
Extenal RAM
00
0FFh
Upper 128 bytes Internal Ram indirect accesses
80h
7Fh
Lower 128 bytes Internal Ram direct or indirect accesses
00
0FFh
Special Function Register direct accesses
80h

Internal and External RAM Address

ZEN8105 provides traditional method for accessing of external RAM. Use *MOVX A, @Ri* or *MOVX @Ri, A;* to access external low 256 bytes RAM; *MOVX A, @DPTR* or *MOVX @DPTR, A* to access external 512 bytes RAM.
Also ZEN8105 can use XPAGE to access external RAM only use *MOVX A, @Ri* or *MOVX @Ri, A* instructions. The user can use XPAGE to represent the high byte address of external 512 bytes RAM.

In flash SSP mode, the XPAGE can also be used as sector selector (Refer to SSP Function)

Data memory page Register (XPAGE)

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	XPAGE [7:0]	RAM Page Selector

3 Flash program memory

The ZEN8105 embeds 16K flash program memory for program code. The flash program memory provides electrical erasure and programming.

NOTE: The last 64byte ($3FC0~$3FFF) is reserved, can't be used as program memory.

In ICP (In-Circuit Programming) mode, the programmer can do all the operations to flash memory, such as erase or write. The read or write operation of flash memory is done by byte, but the erase operation is done by sectors or whole chip.

In ICP mode, the sector erase operation can erase any flash sector except the sector 7. In SSP mode, the sector erase function can erase any flash sector except the sector 7 and the sector that contains SSP code.

The mass-erase operation only support in ICP mode and this operation will erase the entire program memory including sector 7.

3.1 Features

- The program memory consists 8 x 2 KB sectors, total 16KB.
- Programming and erase can be done over the full operation voltage range.
- Write, read and erase operation are all supported by ICP
- Fast mass/sector erase and programming
- Minimum program/erase cycles: 10000
- Minimum years data retention: 10
- Low power consumption

3.2 Flash operation in ICP mode

ICP mode is performed without removing the micro-controller from the system. In ICP mode, the user system must be power-off, and the programmer can refresh the program memory through ICP programming interface. The ICP programming interface consists of 6 wires (VDD, GND, TDO, TDI, TCK, TMS).

At first the four JTAG pins (TDO, TDI, TCK, TMS) are used to enter the programming mode. Only after the four pins are inputted the specified waveform, the CPU will enter the programming mode. For more detailed description please refers to the **FLASH Programmer's user guide**. The ICP mode supports the following operations:

Code-Protect Control mode Programming

ZEN8105 implements code-protect function to offer high safeguard for customer code. Two modes are available for each sector.

Code-protect control mode 0: Used to enable/disable the write/read operation (except mass erase) from any programmer.

Code-protect control mode 1: Used to enable/disable the read operation through MOVC instruction from other sectors; or the sector erase/write operation through SSP Function

To enable the wanted protect mode, the user must use the **Flash Programmer** to set the corresponding protect bit.

Mass Erase

The mass erase operation will erase all the contents of program code, code option, code protect bit and customer code ID, regardless the status of code-protect control mode. (The Flash Programmer supplies customer code ID setting function for customer to distinguish their product.)

Mass erase only available in Flash Programmer.

Sector Erase

The sector erase operation will erase the contents of program code of selected sector except sector 7.This operation can be done by Flash Programmer or the user's program.

If done by the user's program, the code-protect control mode 1 of the selected sector must be disabled.

If done by the Flash Programmer, the code-protect control mode 0 of the selected sector must be disabled.

Note: the last sector has no Sector Erase function.

Write/Read Code

The Write/Read Code operation will write the customer code into the Flash Programming Memory or read the customer code from the Flash Programming Memory. This operation can be done by Flash Programmer or the user's program.

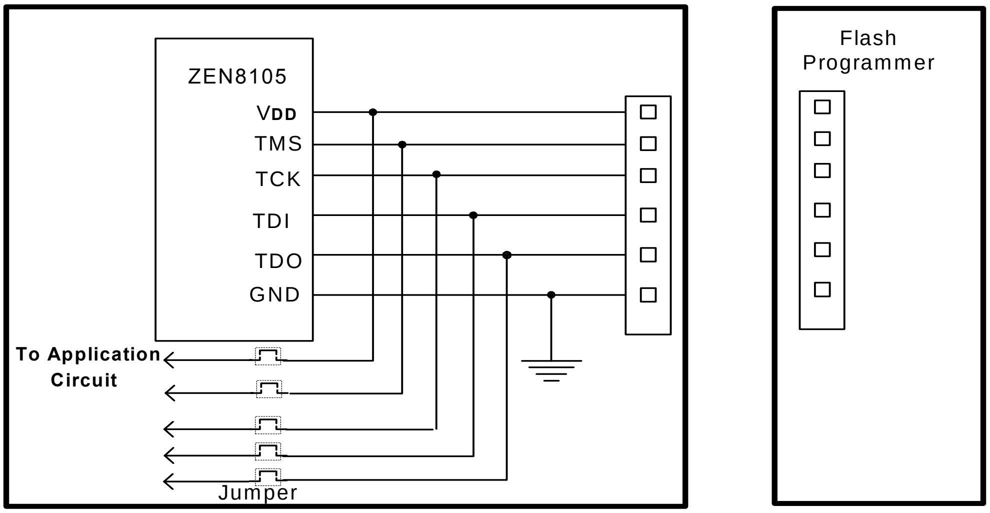
If done by the user's program, the code-protect control mode 1 of the selected sector must be disabled.

If done by the Flash Programmer, the code-protect control mode 0 of the selected sector must be disabled.

Time Control register for Programming		
Operation ICP		**SSP**
Code Protection	Yes	No
Sector Erase	Yes (Without security bit)	Yes (Without security bit)
Mass Erase	Yes	No
Write / Read	Yes (Without security bit)	Yes (Without security bit or its own sector)

In ICP mode，all the flash operations are completed by the programmer through 6-wire interface. Since the program timing of is very sensitive, five jumpers are needed (VDD, TDO, TDI, TCK, TMS) to separate the program pins from the application circuit as the following diagram.

The recommended steps are as following:
The jumpers must be open to separate the programming pins from the application circuit before programming
Connect the programming interface with programmer and begin programming.
Disconnect programmer and short these jumpers after programming is complete.



4 SSP Function

The ZEN8105 provides SSP (Self Sector Programming) function, each sector can be sector erased or programmed by the user's code if the selected sector is not being protected. But once a sector has been programmed, it cannot be reprogrammed before sector erase.

The ZEN8105 build in a complex control flow to prevent the code from carelessly modification. If the dedicated conditions are not met (IB_CON2~5), the SSP will be terminated.

See table **Time Control register for Programming** for more information.

Registers

Time Control register for Programming

F9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CLK0	IB_CLK 0.7	IB_CLK 0.6	IB_CLK 0.5	IB_CLK 0.4	IB_CLK 0.3	IB_CLK 0.2	IB_CLK 0.1	IB_CLK 0.0
FAH	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IB_CLK1	IB_CLK 1.7	IB_CLK 1.6	IB_CLK 1.5	IB_CLK 1.4	IB_CLK 1.3	IB_CLK 1.2	IB_CLK 1.1	IB_CLK 1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_CLKx[1-0] X=0,1**	Flash Programming time select The value in IB_CLK1:IB_CLK0 should be calculated as below: Programming: $65536-\frac{T_{prog}}{8\times T_{sysck}}, \quad 20us \le T_{prog} \le 40us \quad Fsys \ge 1MHz$ $65536-\frac{T_{prog}}{T_{sysck}}, \quad 20us \le T_{prog} \le 40us \quad Fsys < 1MHz$ Typically Tprog = 30us Sector erase: $65536-\frac{T_{prog}}{8\times T_{sysck}}, \quad 50ms \le T_{prog} \le 90ms \quad Fsys \ge 1MHz$ $65536-\frac{T_{prog}}{T_{sysck}}, \quad 50ms \le T_{prog} \le 90ms \quad Fsys < 1MHz$ Typically Tprog = 60ms *Note: When Sector erase function is used, you m ust ensure sysck ≤ 8MHz. If oscillator frequency >8MHz is used, you should set System Clock prescaler (CLKCON) register to get a ≤ 8MHz system clock.*

Offset register for programming

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE. 7	XPAGE. 6	XPAGE. 5	XPAGE. 4	XPAGE. 3	XPAGE. 2	XPAGE. 1	XPAGE. 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-3	**XPAGE[7-3]**	Sector of the flash memory to be programmed, 00000---means sector 0, and so on
2-0	**XPAGE[2-0]**	High Address of Offset of the flash memory sector to be programmed

Offset of flash memory for programming

FBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFFSET	IB_OFFSET.7	IB_OFFSET.6	IB_OFFSET.5	IB_OFFSET.4	IB_OFFSET.3	IB_OFFSET.2	IB_OFFSET.1	IB_OFFSET.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_OFFSET[7-0]**	**Low Address of Offset of the flash memory sector to be programmed**

Data register for programming

FCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_DATA	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_DATA [7:0]**	Data to be programmed

SSP Type select register

F2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON1	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_CON1[7-0]**	**SSP Type select** 0xE6 = Sector Erase 0x6E = Sector Programming

SSP Flow Control Register 1

F3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON2	-	-	-	IB_CON2.4	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
4	**IB_CON2. 4**	**System Clock confirmation** 0 = F_{sys} ≥1 MHz 1 = F_{sys} < 1MHz
3-0	**IB_CON2 [3:0]**	**Must be 05H, else Flash Programming will terminate**

SSP Flow Control Register 2

F4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON3	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	IB_CON3[3:0]	Must be 0AH else Flash Programming will terminate

SSP Flow Control Register 3

F5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON4	-	-	-	-	IB_CON 4.3	IB_CON 4.2	IB_CON 4.1	IB_CON 4.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	0	0	0	0

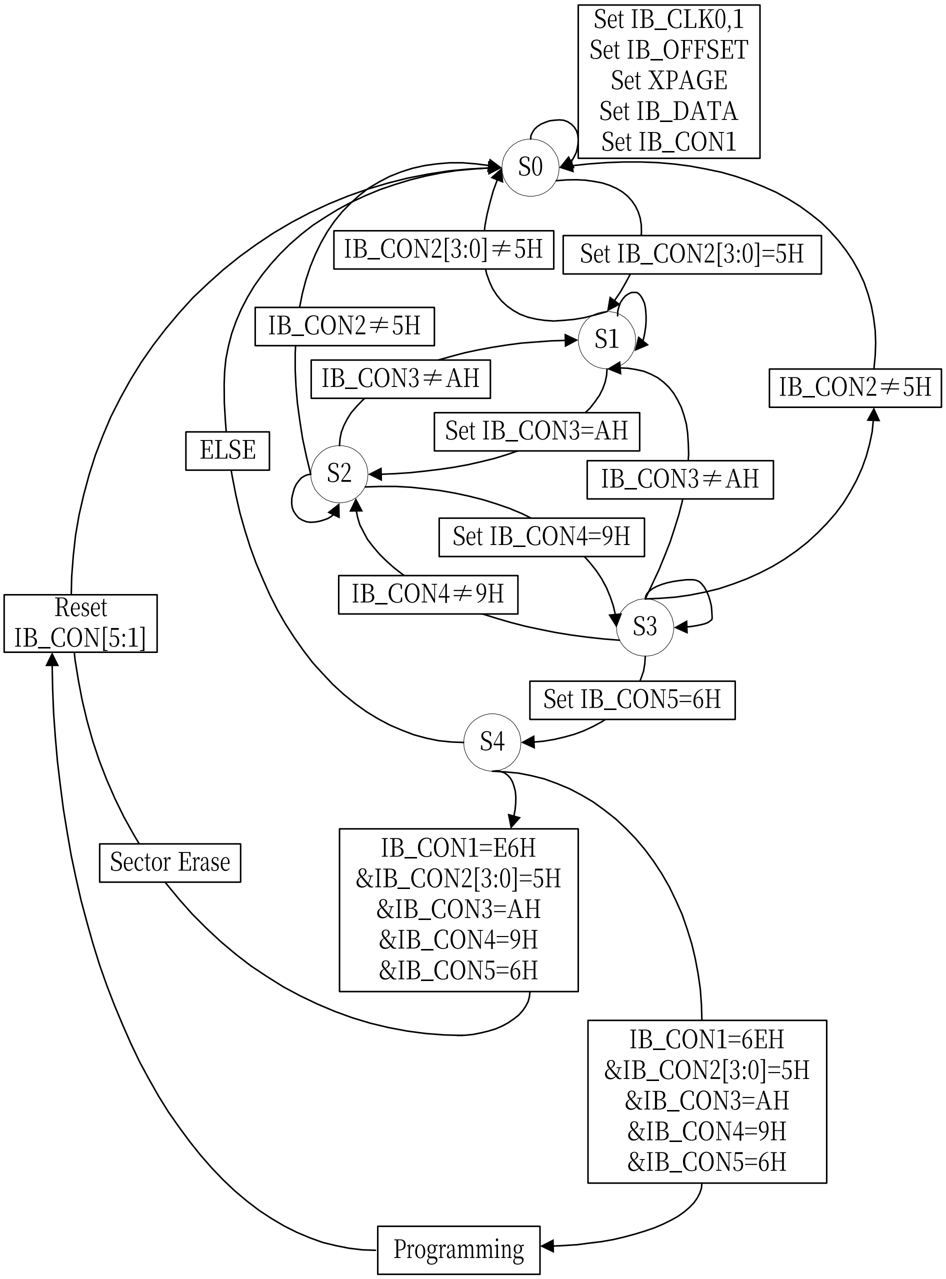
Bit Number	Bit Mnemonic	Description
3-0	IB_CON4[3:0]	Must be 09H, else Flash Programming will terminate

SSP Flow Control Register 4

F6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON5	-	-	-	-	IB_CON 5.3	IB_CON 5.2	IB_CON 5.1	IB_CON 5.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	IB_CON5[3:0]	Must be 06H, else Flash Programming will terminate

4.1 Flash control flow



4.2 SSP Programming Notice:

To successfully complete SSP programming, the user's software must following the steps below:

A.For Code/Data programming:

1)Disable interrupt
2) Fill in IB_CLK1, IB_CLK0
3) Fill in the XPAGE, IB_OFFSET for the corresponding sector
4) Fill in IB_DATA if programming is wanted
5) Fill in IB_CON1~5 sequentially
6) Code / Data programming: CPU will be in IDLE mode
7) Go to Step 3 if more data are to be programmed in the successive address of same sector
It is recommended to add 4 NOP between step 6 and step 7 for more stable operation.

B. Sector Erase:

1)Disable interrupt
2) Fill in IB_CLK1, IB_CLK0
3) Fill in the XPAGE for the corresponding sector
4) Fill in IB_CON1~ 5 sequentially
5) Sector Erase, CPU will be in IDLE mode
6) Go to step 3 for more sectors
It is recommended to add 4 NOP between step 5 and step 6 for more stable operation.

C. For Code Reading:

Just Use "MOVC A, @A+DPTR" or "MOVC A, @A+PC"

5 SystemClock and Oscillator

ZEN8105 has two clock sources. One is ceramic resonator; the other is built-in RC. Clock source is determined by code option. The oscillator generates the basic clock pulses that provide the system clock for the CPU and on-chip peripherals.

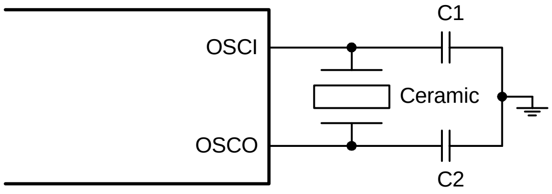
System Clock Control register

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	-	CLKPS1	CLKPS0	-	-	-	-	-
R/W	-	R/W	R/W	-	-	-	-	-
Reset Value (POR/WDT/LVR)	-	0	0	-	-	-	-	-

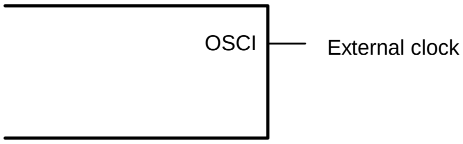
Bit Number	Bit Mnemonic	Description
6-5	**CLKS[1: 0]**	**System Clock prescaler from oscillator** 00 = / 1 (Default) 01 = / 2 10 = / 4 11 = / 12

5.1 Oscillator Type

(1) Ceramic Resonator: 400k – 12MHz



(2) Internal RC Oscillator: 12MHz



5.2 Capacitor selection for oscillator

Ceramic Resonators			Recommend Type	Manufacturer
Frequency C1		C2		
455kHz	47~100pF	47~100pF	ZTB 455KHz	Vectron International
			ZT 455E	Shenzhen DGJB Electronic Co.Ltd.
3.58MHz	15pF	15pF	ZTT 3.580M	Vectron International
			ZT 3.58M*	Shenzhen DGJB Electronic Co.Ltd.
12MHz	15pF	15pF	ZTT 12.000M	Vectron International
			ZT 12M*	Shenzhen DGJB Electronic Co.Ltd.

***- THE SPECIFIED CERAMIC RESONATOR HAS INTERNAL BUILT-IN LOAD CAPACITY**

Notes:

1. ***Capacitor values are used for design guidance only!***
2. *These capacitors were tested with the ceramic listed above for basic start-up and operation. They are* ***not optimized.***
3. *Be careful for the straycapacitance on PCB board, the user should test the performance of the oscillator over the expected V_{DD} and the temperature range for the application.*
4. *Before selecting ceramic, the user should consult the ceramic manufacturer for appropriate value of external component to get best performance, call zenchant for more recommended manufacture.*

6 I/O Port

The ZEN8105 has 29 bi-directional I/O ports. The PORT data is put in Px register. The PORT control register (PxyCR) controls the PORT as input or output. Each I/O port has an internal pull-high resistor, which is controlled by PxyPCR when the PORT is used as input (x=0~3,y=0~7).

For ZEN8105, some I/O pins can share with alternative functions. There exists a priority rule in CPU to avoid these functions be conflict when all the functions are enabled. (See **Port Share** Section)

Port Control Register

E1H- E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0CR (E1H)	P0CR.7	P0CR.6	RES	RES	RES	P0CR.2	RES	P0CR.0
P1CR (E2H)	P1CR.7	P1CR.6	P1CR.5	P1CR.4	P1CR.3	P1CR.2	RES	RES
P2CR (E3H)	P2CR.7	RES	RES	RES	RES	P2CR.2	P2CR.1	RES
P3CR (E4H)	-	-	-	P3CR.4	P3CR.3	P3CR.2	P3CR.1	RES
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PxCRy** **x=0~3, y=0~7**	**Port input/output direction control Register** 0: input mode (default) 1: output mode

Port Pull up Resistor Control Register

E9H- ECH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0PCR (E9H)	P0PCR.7	P0PCR.6	-	-	-	P0PCR.2	-	P0PCR.0
P1PCR (EAH)	P1PCR.7	P1PCR.6	P1PCR.5	P1PCR.4	P1PCR.3	P1PCR.2	-	-
P2PCR (EBH)	P2PCR.7	-	-	-	-	P2PCR.2	P2PCR.1	-
P3PCR (ECH)	**-**	**-**	**-**	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PxPCRy** **X=0-3,y=0-7**	**Input Port internal pull-high resistor enable/disable control** 0: internal pull-high resistor disabled (default) 1: internal pull-high resistor enabled

***Note:** *These ports can be configured as N-channel open drain I/O. but voltage provided for this pin can't exceed VDD+0.3.*

Port Data Register

80H, 90H, A0H, B0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0 (80H)	P0.7	P0.6	-	-	-	*P0.2	-	P0.0
P1 (90H)	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	-	-
P2 (A0H)	P2.7	-	-	-	-	P2.2	P2.1	-
P3 (B0H)	-	-	-	P3.4	P3.3	P3.2	P3.1	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**Px.y** **x=0~3, y=0~7**	**Port Data Register**

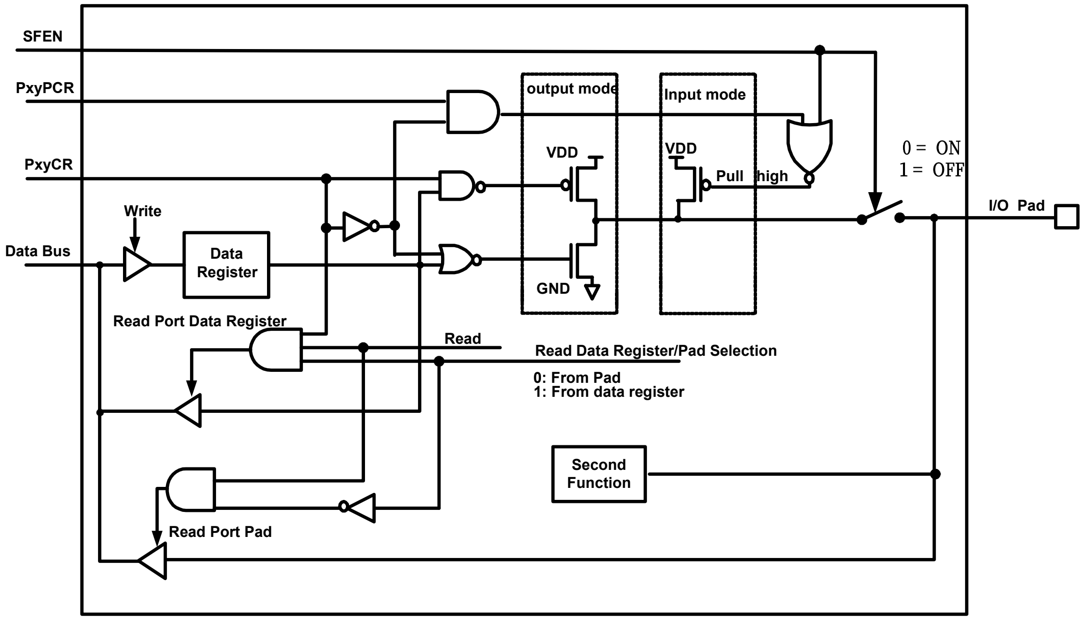
***Note:** *These ports can be configured as N-channel open drain I/O. but voltage provided for this pin can't exceed VDD+0.3.*

Port0 Mode Select Register

EFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0SS	-	-	-	-	-	P02OS	-	-
R/W	-	-	R/W	R/W	R/W	R/W	-	-
Reset Value (POR/WDT/LVR)	-	-	0	0	0	0	-	-

Bit Number	Bit Mnemonic	Description
5-2	**P0xOS** **x=5~2**	**Port 0 output mode select** 0: output mode of the pin is set to N-channel open drain type (default) 1: output mode of the pin is set to CMOS push-pull type

Diagram



Note:

1) The input source of reading input port operation is from input pin directly.

2) The input source of reading output port operation has two paths, one is from the port data Register, and the other is from the output pin directly. The read Instruction distinguishes which path is selected.

3) The read-modify-write instruction reading of the data registers in output mode, and the other instructions are for reading the output pin directly.

4) The destination of writing Input / Output port operation is the data register.

6.1 Port Share

The 29 bi-directional I/O ports can also share second or third special function. But the share priority should obey the **Out Most Inner Least** rule:

The out most pin function in **Pin Configuration** has the highest priority, and the inner most pin function has the lowest priority. This means when one pin is occupied by a higher priority function (if enabled) cannot be used as the lower priority functional pin even the lower priority function is also enabled. Only the higher priority function is closed by software, the corresponding pin can be released for the lower priority function use. Also the function that need pull up resister is also controlled by the same rule.

When port share function is enabled, the user can modify PxyCR, PxyPCR, but these operations will have no effect on the port status until the second function was disabled.

When port share function is enabled, any write to output port will only affect the data register while the port pin keeps unchanged until all the share functions are closed. Any read to output port will return the data register value or pin voltage level according to different instructions.

When port share function is enabled, any write to input port will only affect the data register, any reading to input port will return the pin voltage level only.

If the second function enables analog module such as ADC, the read instruction of pin will only return 0, regardless of the actual pin voltage level or I/O status.

The 29 bi-directional I/O ports also can serve the special features:

Port 0:

-INT2(pin19): Input of external interrupt/2 (P0.0)
-AN0 (pin18): channel 0 of A/D converter analog input (P0.2)
-T1(pin17): Timer1 external input (P0.6)
-INT1 (pin16): Input of external interrupt1 (P0.7)

Port 0 Share Table			
Pin No.	**Priority**	**Function**	**Enable bit**
Pin19	1	INT2	Set EX2 bit in IEN1 register, P0.0 in input mode
	2	P0.0	Above condition is not met
Pin 18	1	AN0	Set ADCON bit in ADC control Register
	2	P0.2	Clear ADCON bit in ADC control Register
Pin 17	1	T1	Set TR1 bit in TCON Register and Set C/T1 bit in TMOD Register, (Auto Pull up)
	2	P0.6	Above condition is not met
Pin 16	2	INT1	Set EX1 bit in IEN0 Register, Port0.7 in input mode
	3	P0.7	Above condition is not met

Note:

Pin18 are configured as N-channel open drain when P0SS=0.

Port 1:

-AN4 (pin20): channel 4 of A/D convert analog input (P1.2)

-RST (pin8): systerm reset pin (P1.7)

-INT40-INT42 (pin3-1), INT43 (pin20): Input of external interrupt (P1.5~P1.2)

-T0 (pin4): Timer0 external input. (P1.6)

Port 1 Share Table			
Pin No.	**Priority**	**Function**	**Enable bit**
Pin 20,1-3	1	INT43-40	Set EX4 bit in IEN1 register and EXS43-40 bit, P1.2-1.5 in input mode
	2	AN4-AN7	Set ADCON bit in ADC control Register
	3	P1.2-P1.5	Above condition is not met
Pin 4	1	T0	Set TR0 bit in TCON Register and Set C/T0 bit in TMOD Register, (Auto Pull up)
	2	P1.6	Above condition is not met
Pin 5		RST	Selected by Code Option
		P1.7	Selected by Code Option

Port 2:

-

-

-

-

-TXD(pin13): data output pin for EUART (P2.1)

-RXD(pin14): data input pin for EUART (P2.2)

-

Port 2 Share Table			
Pin No.	**Priority**	**Function**	**Enable bit**
- -	-		
		-	- -
	-		
-	-		
	-		
Pin 14	1	RXD	Set REN bit in SCON Register, (Auto Pull up)
		-	- -
	2	P2.2	Above condition is not met
Pin 13	1	TXD	When Write to SBUF Register
	-		- -
	-	P2.1	Above condition is not met
Pin17	--		
		-	- -
	-		

Port 3:

-XTAL1 (pin9): External oscillator pin (P3.3)
-XTAL0 (pin8): External oscillator pin (P3.4)
-T2EX (pin13): external clock for capture timer (P3.2)
-T2/ INT0 (pin14): external input for Timer2/ input pin of external interrupt. (P3.1)
-

Port 3 Share Table			
Pin No.	**Priority**	**Function**	**Enable bit**
Pin 8~9		XTAL0/1	Selected by Code Option
		P3.4~P3.3	Selected by Code Option
Pin 10	1	T2EX	Set TR2 bit in T2CON Register and set C/T2 bit and set EXEN2 bit in T2MOD Register, (Auto Pull up)
	2	P3.2	Above condition is not met
Pin 11	1	T2	Set TR2 bit in T2CON Register and set C/T2 bit in T2MOD Register, (Auto Pull up)
	2	INT0	Set EX0 bit in IEN0 Register and Port3.1 in input mode
	3	P3.1	Above condition is not met
-	-		
			- - - - - -
	-		

7 Timer

The ZEN8105 has three independent 16-bit timers, Timer0, Timer1 and Timer 2, which are compatible to traditional 8052. Timer0/1 also have compare function.

7.1 Timer 0 & 1

Each timer is implemented as a 16-bit register accessed as two cascaded Timer x/ Counter x Date Registers: THx & TLx (x = 0, 1). They are controlled by the register TCON and TMOD. The Timer 0 & Timer 1 interrupts can be enabled by setting the ET0 & ET1 bit in the IEN0 register (Refer to section Interrupt).

Timer x / Counter x Control register (x = 0,1)

88H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	**Bit Mnemonic**	**Description**
7,5	**TFx** **x = 0, 1**	**Timer x overflow flag** 0 = Timer x no overflow. 1 = Timer x overflow, set by hardware; set by software will cause a timer interrupt
6,4	**TRx** **x = 0, 1**	**Timer x start, stop control** 0 = Stop timer x 1 = Start timer x

Timer x / Counter x mode register (x = 0,1)

89H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
TMOD	GATE1	C/$\overline{T1}$	M11	M10	GATE0	C/$\overline{T0}$	M01	M00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	**Bit Mnemonic**	**Description**
7,3	**GATEx** **x = 0, 1**	**Timer x Gate Control** 0 = Disable 1 = Enable (if TRx & GATEx =1,INTx will auto set as input state and internal Pull-high is enable automatically)
6,2	**C/$\overline{Tx}$** **x = 0, 1**	**Timer x Timer / Counter mode selected bit.** 0 = Timer Mode, T0 or T1 pin is used as I/O port 1 = Counter Mode (if TRx & **C/**$\overline{Tx}$ =1,T0/T1 auto set as input state and internal Pull-high is enable automatically)
5-4 1-0	**Mx [1:0]** **x = 0, 1**	**Timer x Timer mode selected bit,** 00 = Mode 0, 13-bit up counter / timer, bit7~5 of TLx is ignored. 01 = Mode 1, 16-bit up counter / timer 10 = Mode 2, 8-bit auto-reload up counter/timer 11 = Mode 3 (only for Timer0), two 8-bit up timer.

Timer x / Counter x Data Register (x = 0,1)

8AH, 8CH	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
TL0	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
8BH, 8DH	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
TL1	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TLx.y , THx.y** **x=0~1, y=0~7**	**Timer x Low & High byte counter**

Timer x / Counter x Compare EnableRegister (x = 0,1)

CEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCOME	-	-	-	-	-	-	TC1	TC0
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	-	-	0	0

Bit Number	Bit Mnemonic	Description
1-0	**TCx[1-0]**	**Timer x Compare Enable (x=0,1)** 0: Disable compare function of Timer0/1 1: Enable compare function of Timer0/1

Timer 0 & Timer 1 Mode

Both timers operate in one of four primary modes selected by the Mode Select bits Mx1-Mx0 (x = 0, 1) in the Counter/Timer Mode register (TMOD). Since Timer 1 operates the same as Timer 0, the text below describes only Timer 0. And Timer 1 is identical with Timer 0 in mode 0/1/2, except that it uses its own register bits and uses INT1 when Gate1=1.

Mode 0: 13-bit Counter/Timer

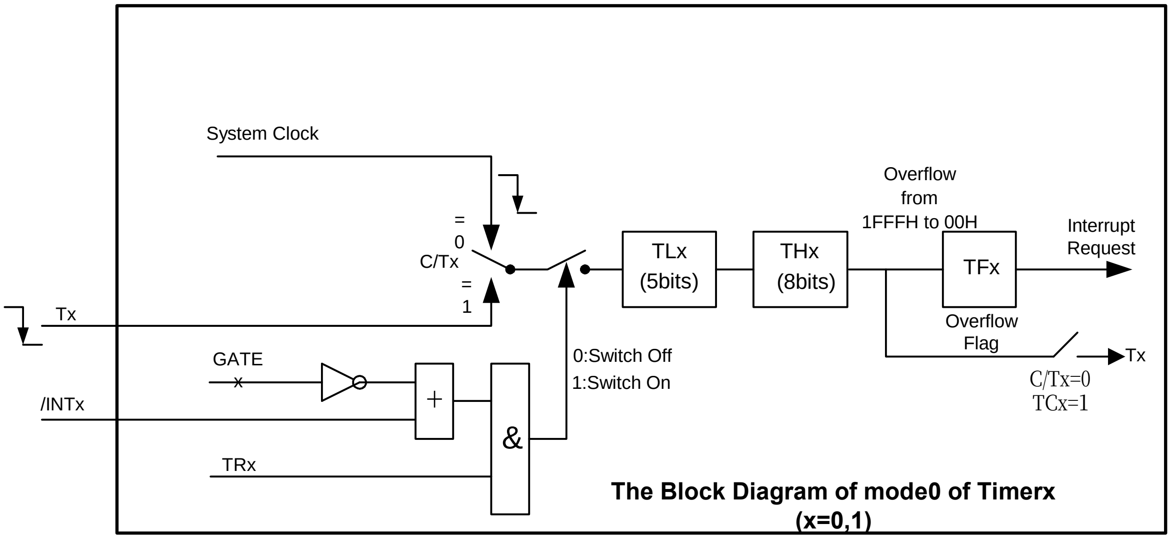
Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The TH0 register holds the high eight bits of the 13-bit counter/timer, TL0 holds the five low bits TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts is enabled. The C/T0 bit (TMOD.2) selects the counter/timer’s clock source.

If C/T0 = 1, high-to-low transitions at the Timer 0 input pin (T0) will increase the timer/Counter 0 Data register. Else if C/T0 = 0, selects the system clock to increase the timer/Counter 0 Data register. The system clock is controlled by the bits CLKS1:0 in register CLKCON.

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3)= 0, or GATA0 = l and the input signal /INT0 is active. Setting GATE0 to ‘1’ allows the timer to be controlled by the external input signal /INT0, facilitating positive pulse width in /INT0 measurements. Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

In Compare mode, the internal counter is constantly countered from TH0/1 and TL0/1 register value to 0x1FFF. When an overflow occurs, the T0/T1 pin will be inverted. The T0/T1 pin is automatically set as output mode by hardware when in compare mode.

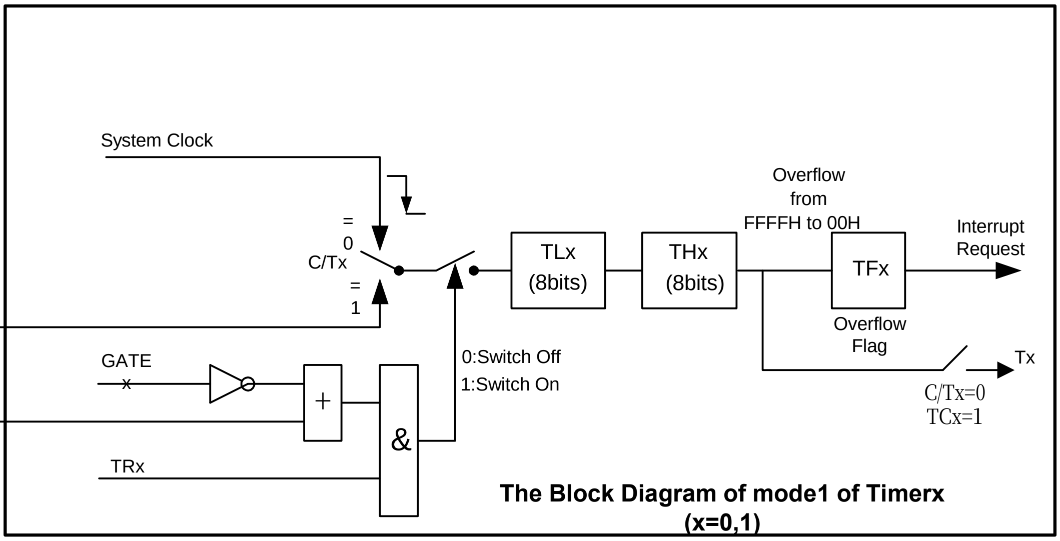
At the same time, interrupt flag bit of time0/1 is set. Timer0/1 must be running in Timer mode (C/Tx=0) when compare function enabled.



The Block Diagram of mode0 of Timerx (x=0,1)

Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.



The Block Diagram of mode1 of Timerx (x=0,1)

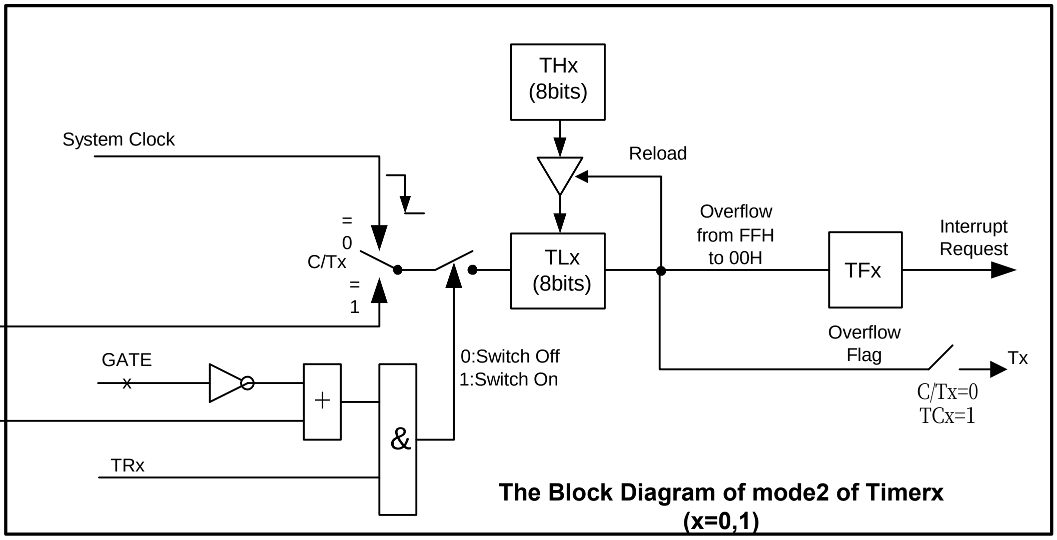
Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as a 8-bit counter/timers with automatic reload the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from 0xFF to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be precise.

Except the Auto-Reload function, both counter / timers are enabled and configured in Mode 2 is the same as in Mode 0 & Mode 1.

In Compare mode, the internal counter is constantly countered from TL0/1 register value to 0xFF. When an overflow occurs, the T0/T1 pin will be inverted. The T0/T1 pin is automatically set as output mode by hardware when in compare mode.

At the same time, interrupt flag bit of time0/1 is set. Timer0/1 must be running in Timer mode when compare function enabled.



The Block Diagram of mode2 of Timerx (x=0,1)

Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

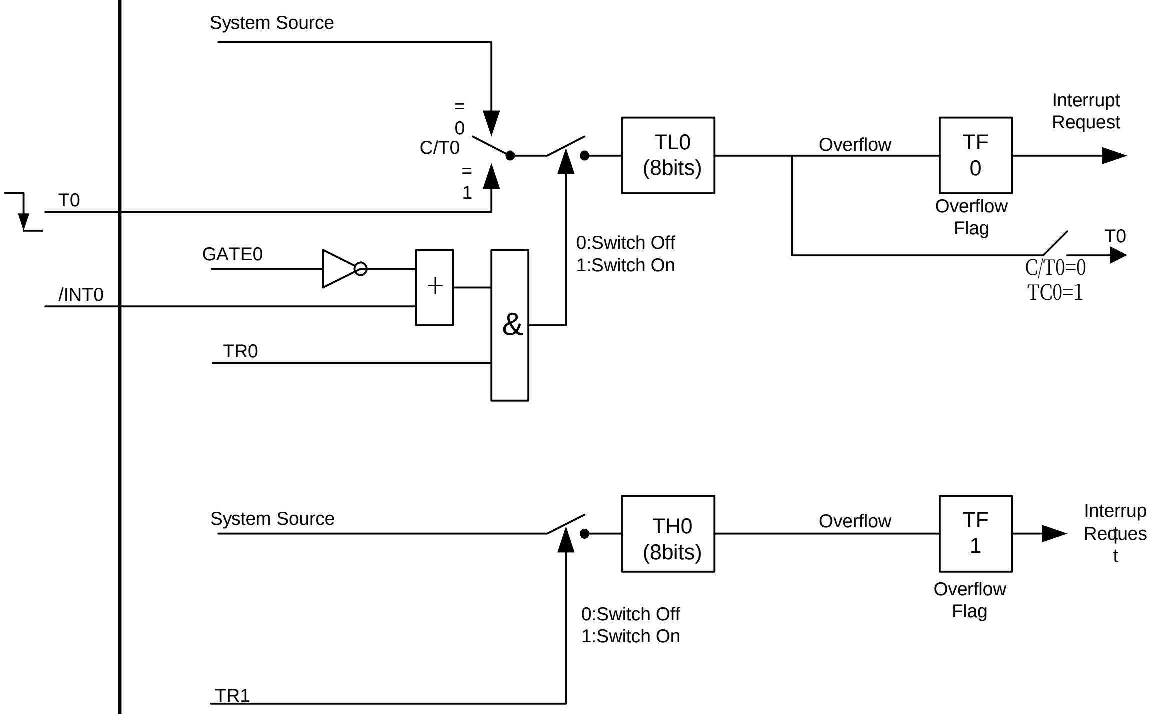
In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0.

TL0 can use either the system clock or an external input signal as its time base. The TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for EUART.

While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings, because TR1 is used by Time 0. And the pull high resistor of T1 input pin is also disabled.

To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.



The Block Diagram of mode3 of Timer0

Note: when timer0/1 used in counter mode, the all input signal is synchronized by system clock, so T0/T1 must be lower than half of system clock, /INT0 or /INT1 must be lower than quarter of system clock.

7.2 Timer 2

The Timer 2 in ZEN8105 is implemented as a 16-bit register accessed as two cascaded Data Registers: TH2 and TL2. It is controlled by the register T2CON and T2MOD. Timer2 interrupt can be enabled by setting ET2 bit in IEN0 register (**see Interrupt section**).

Timer2 operation is similar to Timer 0 and Timer 1. C/T2 selects system clock (timer operation) or external pin T2 (counter mode) as the timer clock input. Setting TR2 allows Timer 2/Counter 2 Data Register to increment by the selected input.

Timer 2 Control register

C8H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TF2**	**Timer2 overflow flag bit** 0 = No overflow (Must be cleared by software) 1 = Overflow (Set by hardware if RCLK = 0 & TCLK = 0)
6	**EXF2**	**External event input (falling edge) from T2EX pin detected flag bit.** 0 = No external event input (Must be cleared by software) 1 = Detected external event input (Set by hardware if EXEN2 = 1)
5	**RCLK**	**EUART Receive Clock control bit** 0 = Timer 1 1 = Timer 2
4	**TCLK**	**EUART Transmit Clock control bit** 0 = Timer 1 1 = Timer 2
3	**EXEN2**	**External ev ent input (falling edge) from T2EX pin used as Reload/Capture trigger enable/disable control bit** 0 = Ignore events on T2EX pin for Timer 2 operation 1 = Cause a capture or reload when a negative edge on T2EX pin is detected, when Timer 2 is not used to clock the EUART (T2EX always has a pull up resistor)
2	**TR2**	**Timer2 start/stop control bi** 0 = Stop Timer 2 1 = Start Timer 2
1	**C/$\overline{T2}$**	**Timer 2 Timer / Counter mode selected bit** 0 = Timer Mode, T2 pin is used as I/O port 1 = Counter Mode, the internal pull-up resister is turned on
0	**CP/$\overline{RL2}$**	**Capture/Reload mode selected bit** 0: 16 bits timer/counter with reload function 1: 16 bits timer/counter with capture function

Timer 2 Mode Control register

C9H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
T2MOD	-	-	-	-	-	-	T2OE	DCEN
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	-	-	0	0

Bit Number	Bit Mnemonic	Description
1	**T2OE**	**Timer 2 Output Enable bit** 0 = Set P3.1/T2 as clock input or I/O port. 1 = Set P3.1/T2 as clock output (Baud-Rate generator mode)
0	**DCEN**	**Down Counter Enable bit** 0 = Disable Timer 2 as up/down counter, timer 2 is an up counter. 1 = Enable Timer 2 as up/down counter.

Timer 2 Reload/Capture & Data Registers

CAH- CBH	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
RCAP2L	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	**Bit Mnemonic**	**Description**
7-0	**RCAP2L.x**	**Timer 2 Reload/ Capturer Data, x=0~7**
	RCAP2H.x	

Timer 2 Low & High byte counter Registers

CCH- CDH	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
TL2	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	**Bit Mnemonic**	**Description**
7-0	**TL2.x**	**Timer 2 Low & High byte counter, x=0~7**
	TH2.x	

Timer 2 Modes

Timer 2 has 4 operating modes: Capture/Reload, Auto-reload mode with up or down counter, Baud Rate Generator and Programmable clock-output. These modes are selected by the combination of RCLK, TCLK and CP/RL2.

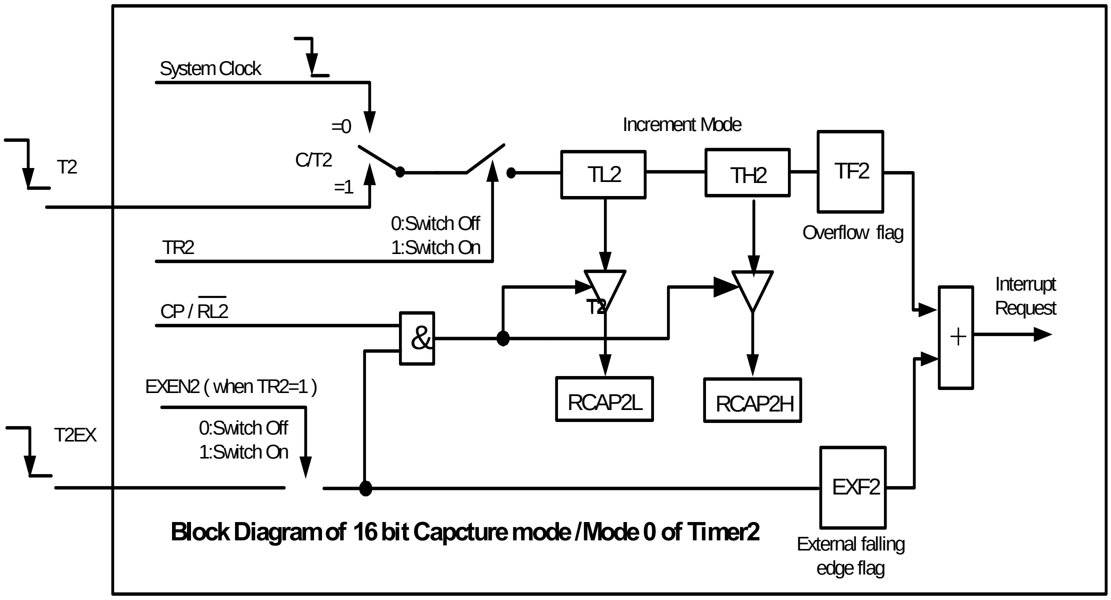
Timer 2 Mode								
C/T2	T2OE	DCEN	TR2	CP/RL2	RCLK	TCLK	Mode	
X	0	X	1	1	0	0	0	16 bit capture
X	0	0	1	0	0	0	1	16 bit auto-reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud-Rate generator
					X	1		
0	1	X	1	X	0	0	3	Programmable clock-output only
					1	X	3	Programmable clock-output, with Baud-rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer2 stop

Mode 0: 16 bit Capture

In the capture mode, two options are selected by bit EXEN2 in T2CON.

If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which will set TF2 on overflow to generate an interrupt if IET2 is enabled.

If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L respectively, In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can also generate an interrupt if IET2 is enabled.



Block Diagram of 16 bit Capcture mode /Mode 0 of Timer2

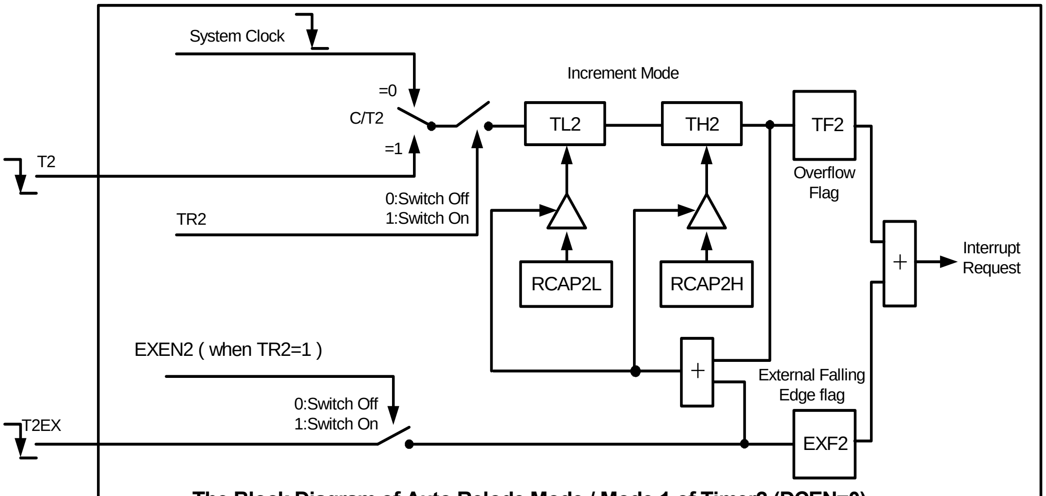
Mode 1: 16 bit auto-reload timer

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit in T2MOD. Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

When DCEN=0, two options are selected by bit EXEN2 in T2CON.

If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L, which are pressed by software.

If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled. Setting the DCEN bit enables Timer 2 to count up or down



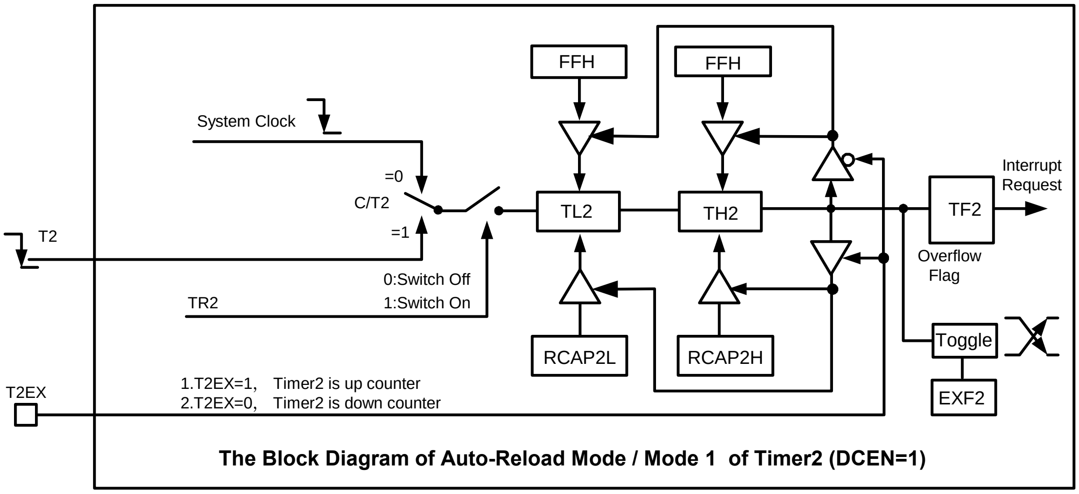
The Block Diagram of Auto Relode Mode / Mode 1 of Timer2 (DCEN=0)

When DCEN = 1, the T2EX pin controls the direction of the count, and EXEN2's control is invalid.

A logical "1" at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logical "0" at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17^{th} bit of resolution. In this operating mode, EXF2 does not flag an interrupt.



The Block Diagram of Auto-Reload Mode / Mode 1 of Timer2 (DCEN=1)

Mode 2: Baud-Rate generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other.

Setting RCLK and/or TCLK will put Timer 2 into its baud rate generator mode, which is similar to the auto-reload mode.

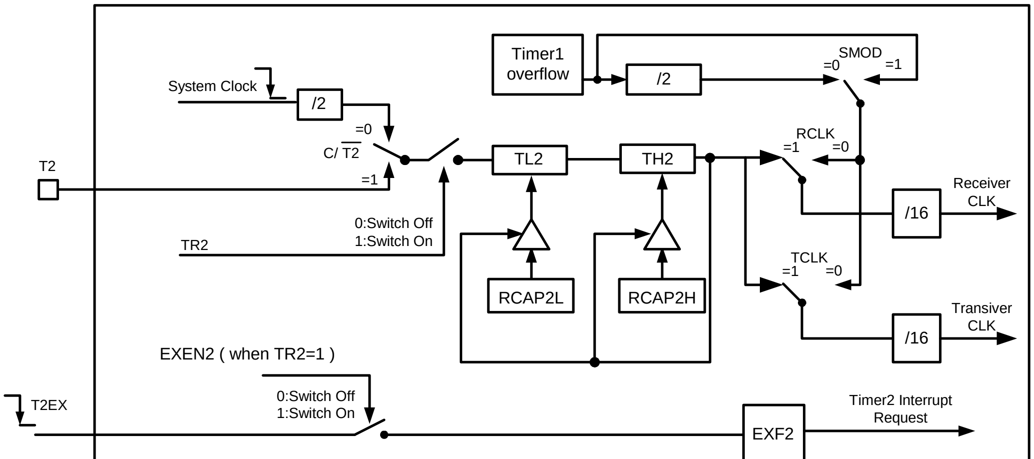
Over flow of Timer 2 will causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L that preset by software. But this will not generate an interrupt.

If EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

The baud rates in EUART Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$BaudRate = \frac{1}{2\times 16}\times\frac{System\quad Clock}{65536-[RCAP2H,RCAP2L]}\ ;\qquad C/\overline{T2} = 1$$

$$BaudRate = \frac{1}{16}\times\frac{T2\quad frequency}{65536-[RCAP2H,RCAP2L]}\ ;\qquad C/\overline{T2} = 0$$



The Block Diagram of Baud-Rate Generator / Mode 2 of Timer2

Note: *When Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. The reason is stated as below:*

Timer is incremented every state time, and the results of a read or write may not be accurate.

The RCAP2 registers may be read but should not be written to; because a write might overlap a reload and cause write and/or reload errors. So, the timer 2 must be turned off (clear TR2) before accessing the TH2 / TL2 or RCAP2H / RCAP2L / registers.

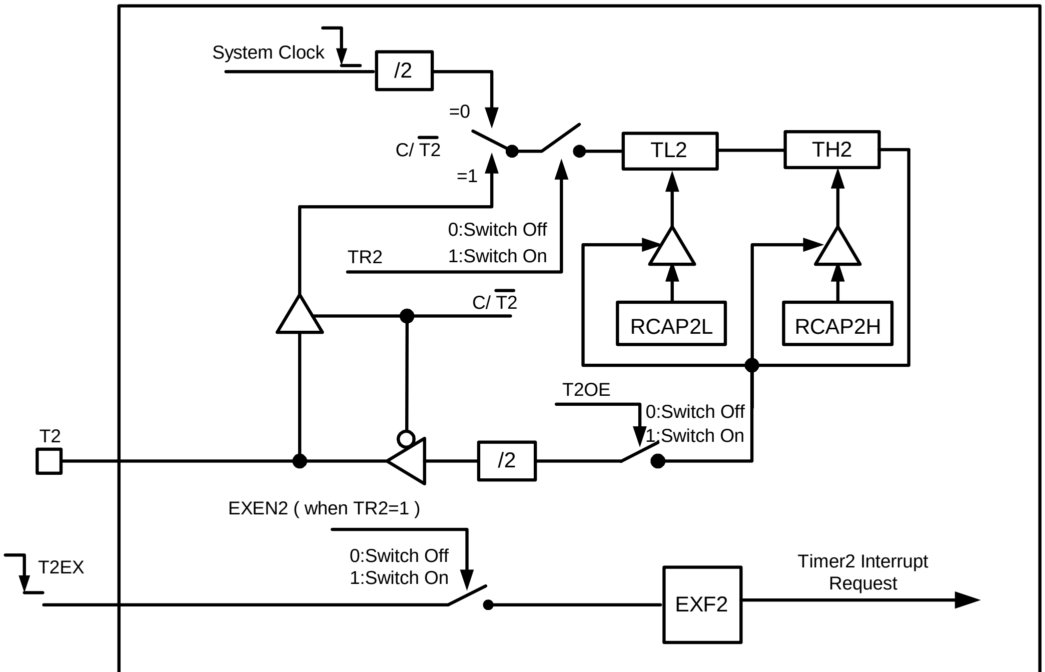
Mode 3: Programmable Clock output

A 50% duty cycle clock can be programmed to come out on P3.1. To configure the Timer2 as a clock generator, bit C/$\overline{\text{T2}}$ must be cleared and bit T2OE must be set. Bit TR2 starts and stops the timer.

The clock-out frequency depends on the system clock and the reload value of Timer2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$Clock \quad Out \quad Frequency = \frac{1}{2 \times 2} \times \frac{System \quad Clock}{65536 - [RCAP2H, RCAP2L]}$$

In the clock-out mode, Timer 2 overflow will not generate an interrupt, so it is possible to use Timer 2 as a baud-rate generator and a clock output simultaneously with the same frequency.



The Block Diagram of Programmable Clock output / Mode 3 of Timer2

Additional Notes on Timer:

1. Both TF2 and EXF2 can cause timer2 interrupt request, and they have the same vector address.
2. TF2 and EXF2 are set as 1 by hardware while event occurs. But they can also be set by software at any time. Only the software and the hardware reset will be able to clear TF2 & EXF2 to 0.
3. When EA =1 & ET2=1, setting TF2 or EXF2 as 1 by software will cause a timer2 interrupt.
4. While Timer1/2 used as baud rate generator, Reading or writing TH1/TL1, TH2/TL2, writing RCAPH2/RCAPL2 will affect the accuracy of baud rate, thus might make cause communication error.

8 Enhanced Universal Asynchronous Receiver-Transmitter (EUART)

The ZEN8105 has one enhanced UART (EUART), which is compatible with the conventional 8051 UART. The serial ports are capable of synchronous as well as asynchronous communication. In the Synchronous mode the ZEN8105 generates the clock and operates in a half duplex mode. In the asynchronous mode, full duplex operation is available. This means that it can simultaneously transmit and receive data. The transmit register and the receive buffer are both addressed as SBUF Special Function Register. However any write to SBUF will be to the transmit register, while a read from SBUF will be from the receiver buffer register.

The baud rate can be selected from the oscillator (divided by a constant), Timer 1/2 overflow. In addition to the baud rate generation, enhancements over the standard 8051 UART include Framing Error detection, break detect, automatic address recognition.

NOTE: TXD Pin is shared with P2.1, this pin only used as TXD pin in data transmitting, after transmit, it auto turn to I/O state, so, In the customer's UART initial program, P2.1 must set as output high.

8.1 EUART Mode

Mode Description

The EUART can be operated in 4 modes. The user must first initialize the SCON before any communication can take place. This involves selection of the Mode and baud rate. The Timer 1 or Timer 2 should also be initialized if modes 1 or mode 3 is used.

In all of the four modes, transmission is started by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. This will generate a clock on the TXD pin and shift in 8 bits on the RXD pin. Reception is initiated in the other modes by the incoming start bit if REN = 1. The external transmitter will start the communication by transmitting the start bit.

EUART Mode Summary

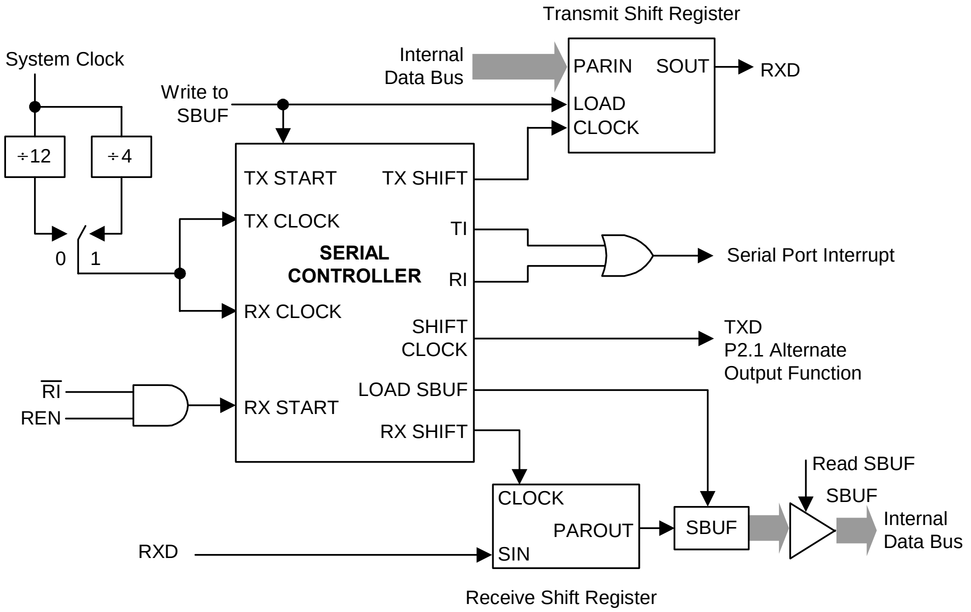
SM0	SM1	Mode	Type	Baud Clock	Frame Size	Start Bit	Stop Bit	9th bit
0	0	0	Sych	4 or 12 SysClk	8 bits	NO	NO	None
0	1	1	Ansych	Timer 1 or 2	10 bits	1	1	None
1	0	2	Ansych	32 or 64 SysClk	11 bits	1	1	0, 1
1	1	3	Ansych	Timer 1 or 2	11 bits	1	1	0, 1

Mode 0: Synchronous Mode, Half duplex

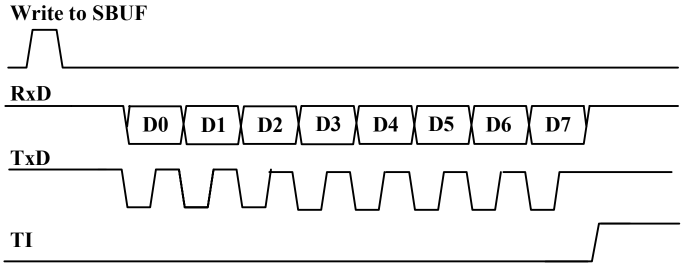
This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RXD line. TXD is used to transmit the shift clock. The TxD clock is provided by the ZEN8105 whether the device is transmitting or receiving. This mode is therefore a half duplex mode of serial communication. In this mode, 8 bits are transmitted or received per frame. The LSB is transmitted/received first.

The baud rate is fixed at 1/12 or 1/4 of the system clock. This baud rate is determined in the SM2 bit (SCON.5). When this bit is set to 0, the serial port runs at 1/12 of the clock. When set to 1, the serial port runs at 1/4 of the clock. This additional facility of programmable baud rate in mode 0 is the only difference between the standard 8051 and the ZEN8105.

The functional block diagram is shown below. Data enters and leaves the serial port on the RxD line. The TxD line is used to output the SHIFT CLOCK. The SHIFT CLOCK is used to shift data into and out of the ZEN8105.

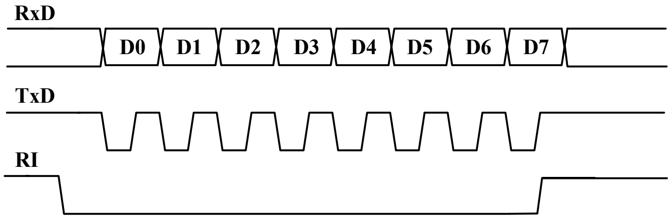


Any instruction that uses SBUF as a destination register ("write to SBUF" signal) will start the transmission. The next machine cycle tells the Tx control block to commence a transmission. The data shift occurs at the falling edge of the SHIFT CLOCK, and the contents of the transmit shift register is shifted one position to the right. As data bits shift to the right, zeros come in from the left. After transmission of all 8 bits in the transmit shift register, the Tx control block will deactivates SEND and sets TI (SCON.1) at the rising edge of the next machine cycle, and RxD keeps High.



Send Timing of Mode 0

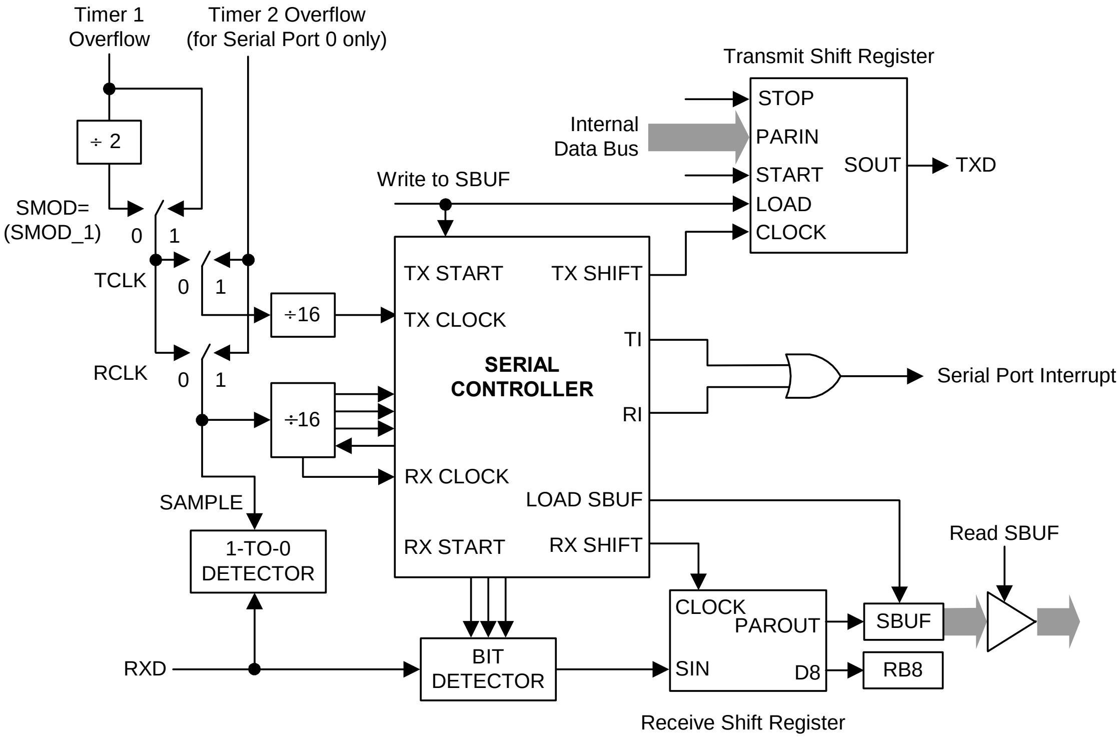
Reception is initiated by the condition REN (SCON.4)= 1 and RI (SCON.0) = 0. The next machine cycle activates RECEIVE. The data latch occurs at the rising edge of the SHIFT CLOCK, and the contents of the receive shift register are shifted one position to the left. After the receiving of all 8 bits into the receive shift register, the RX control block will deactivates RECEIVE and sets RI at the rising edge of the next machine cycle, and the reception will not be enabled till the RI is cleared by software.



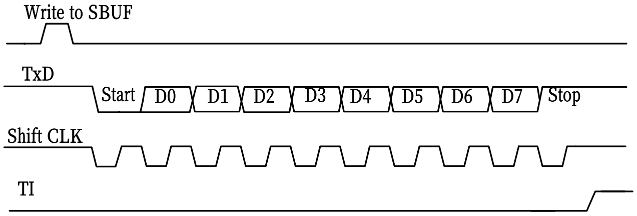
Receive Timing of Mode 0

Mode 1: 8-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

The full duplex asynchronous mode is used in this mode. Serial communication frames are made up of 10 bits transmitted on TxD and received on RxD. The 10 bits consist of a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). On receiving, the eight data bits are stored in SBUF and the stop bit goes into RB8 (SCON.2). The baud rate in this mode is variable. The serial receive and transmit baud rate can be programmed to be 1/16 or 1/32 of the Timer 1 overflow or 1/16 of Timer 2 overflow (see "**Baud Rate**" section). The functional block diagram is shown below.



Transmission begins with a “write to SBUF” signal, and it actually commences at the next machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SUBF” signal). The start bit is firstly put out on TxD pin, then are 8 bits of data. After all 8 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time, this will be at the 10th rollover of the divide-by-16 counter after a write to SBUF.



Send Timing of Mode 1

Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter. The 16 states of the counter divide each bit time into 16 counter states. The bit detector samples the value of RxD at the 7th, 8th and 9th counter state of each bit time.

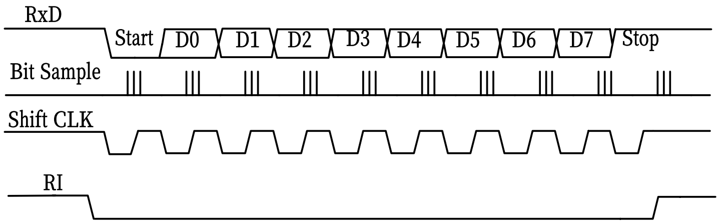
The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the first bit after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again waiting for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF. After shifting in 8 data bits and the stop bit, the SBUF and RB8 are loaded and RI are set if the following conditions are met:

RI must be 0,

Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

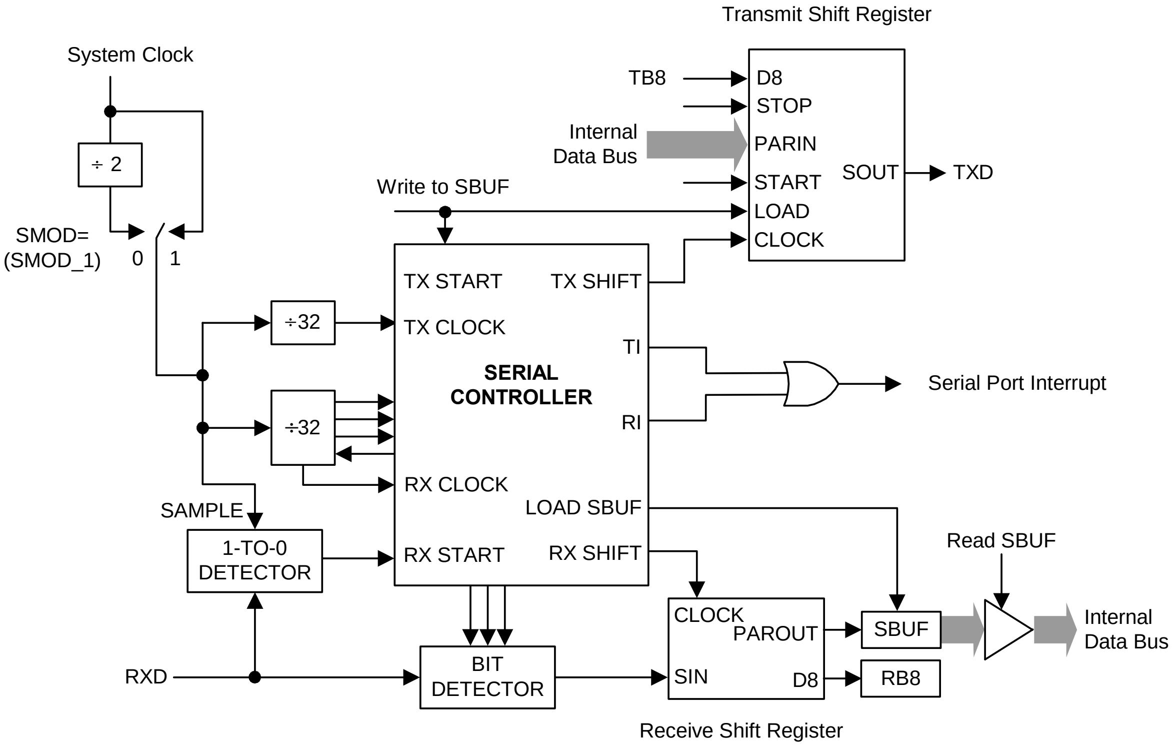
After the middle of the stop bit, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



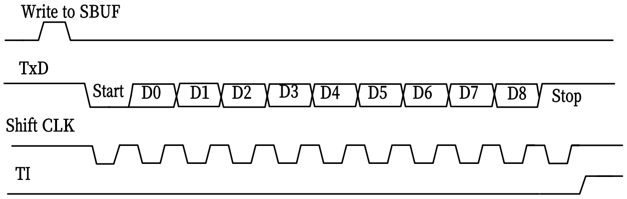
Receive Timing of Mode 1

Mode 2: 9-Bit EUART, Fixed Baud Rate, Asynchronous Full-Duplex

This mode uses a total of 11 bits in asynchronous full-duplex communication. The functional block diagram is shown below. The frame consists of one start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). Mode 2 supports multiprocessor communications and hardware address recognition **(Multiprocessor Communication Sector)**. When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1, for example, the parity bit P in the PSW or used as data/address flag in multiprocessor communications. When data is received, the 9th data bit goes into RB8 and the stop bit is not saved. The baud rate is programmable to either 1/32 or 1/64 of the system working frequency, as determined by the SMOD bit in PCON.



Transmission begins with a “write to SBUF” signal, the “write to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register. Transmission actually commences at the next machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SUBF” signal). The start bit is firstly put out on TxD pin, then are the 9 bits of data. After all 9 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time, this will be at the 11th rollover of the divide-by-16 counter after a write to SBUF.



Send Timing of Mode 2

Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter. The 16 states of the counter divide each bit time into 16 counter states. The bit detector samples the value of RxD at the 7th, 8th and 9th counter state of each bit time. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the first bit detected after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF.

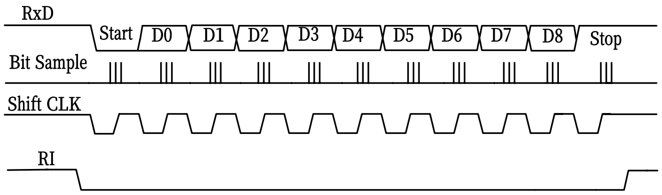
After shifting in 9 data bits and the stop bit, the SBUF and RB8 are loaded and RI are set if the following conditions are met:

RI must be 0

Either SM2 = 0, or the received 9th bit = 1, and the received byte matches the EUART address

If these conditions are met, then the 9th bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

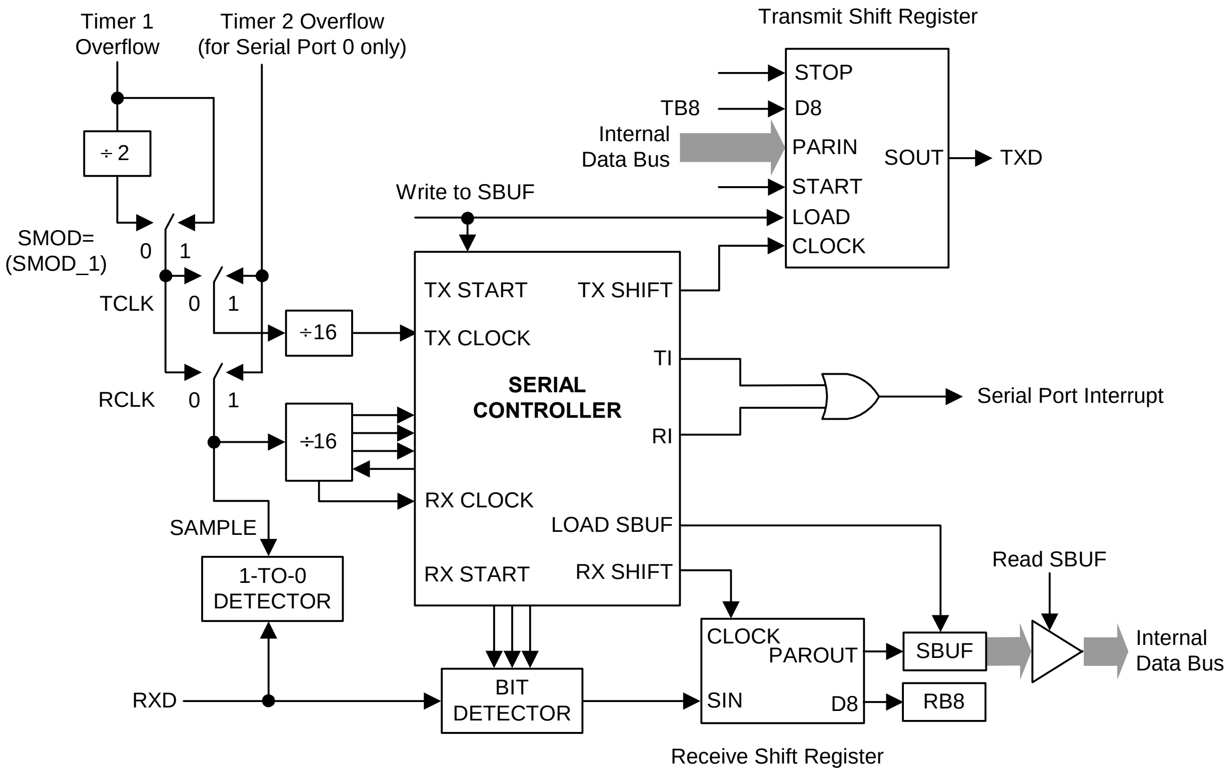
After the middle of the stop bit, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



Receive Timing of Mode 2

Mode 3: 9-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

Mode 3 uses the Mode 2 transmission protocol with the Mode 1 baud rate generation. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable 9th data bit, and a stop bit. The baud rate is derived from Timer 1 or Timer 2 overflows. Multiprocessor communications and hardware address recognition are supported.



8.2 Baud rate Generate

In **Mode 0**, the baud rate is fixed at 1/12 or 1/4 of the oscillator frequency. This baud rate is determined by SM2 bit. When set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock.

In **Mode 1 & Mode 3**, the baud rate generated is a function of timer overflow, as shown in Equation below. EUART can use both Timer 1 operating in Mode 2 (8-bit Auto-Reload Counter/Timer), and Timer 2 operating in Mode 2 (Baud Rate Generator Mode) to generate the baud rate (note that the Tx and Rx clocks are selected separately). Each times the timer increments from its maximum count (FFH for Timer 1 or FFFFH for Timer 2), a clock is sent to the baud rate logic. Also Timer 1 mode 0 & mode 1 can used as baud-rate generator, but this is no so convenient for use, and is not recommended.

Timer 2 is selected as TX and/or RX baud clock source by setting the TCLK (T2CON.4) and/or RCLK (T2CON.5) bits, respectively (See **Timer** section). When either TCLK or RCLK is set to logic 1, Timer 2 is forced into Baud Rate Generator Mode. If TCLK and/or RCLK is logic 0, Timer 1 acts as the baud clock source for the Tx and/or Rx circuits, respectively.

The Mode 1 & 3 baud rate equations are shown below, where TH1 is the 8-bit reload register for Timer 1, SMOD is the EUART baud rate double (PCON.7) and [RCAP2H, RCAP2L] is the 16-bit reload register for Timer 2. *T1CLK* is the clock source of Timer 1, and *T2CLK* is the clock source of Timer2.

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{T1CLK}{256 - TH1}$$, Baud Rate using Timer 1, Mode 2

$$BaudRate = \frac{1}{2\times 16} \times \frac{T2CLK}{65536 - [RCAP2H, RCAP2L]}$$, Baud Rate using Timer 2, with system clock

$$BaudRate = \frac{1}{16} \times \frac{T2CLK}{65536 - [RCAP2H, RCAP2L]}$$, Baud Rate using Timer 2, with T2 clock input

In **Mode 2**, the baud rate is fixed at 1/32 or 1/64 of the system clock. This baud rate is determined in the SMOD bit (PCON.7). When this bit is set to 0, the serial port runs at 1/64 of the clock. When set to 1, the serial port runs at 1/32 of the clock.

$$BaudRate = 2^{SMOD} \times (\frac{SYSCLK}{64})$$

8.3 Multi-processor communication.

Software address recognition

Modes 2 and 3 of the EUART have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB8. Then a stop bit follows. The EUART can be programmed such that when the stop bit is received, the EUART interrupt will be activated (i.e. the request flag RI is set) only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON register.

A way to use this feature in multiprocessor communications is as follows. If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte.

With SM2 = 1, no other slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM2 again. The slaves that were not addressed leave their SM2 set and go on with their business, ignoring the incoming data bytes.

Note that in mode 0, SM2 selects baud rate. In mode 1, SM2 can be used to check the validity of the stop bit. If SM2 = 1 in mode 1, the receive interrupt will not be activated unless a valid stop bit is received.

Automatic (Hardware) address recognition

In Mode 2 & 3, setting the SM2 bit will configure EUART act as following: when a stop bit is received, EUART will generate an interrupt only if the 9th bit that goes into RB8 is logic 1 (address byte) and the received data byte matches the EUART slave address. Following the received address interrupt, the slave should clear its SM2 bit to enable interrupts on the reception of the following data byte(s).

The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not a data. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte, which ensures that they will be interrupted only by the reception of an address byte. The Automatic address recognition feature further ensures that only the addressed slave will be interrupted. The address comparison is done by hardware not software.

After being interrupted, the addressed slave clears the SM2 bit to receive data bytes. The un-addressed slaves will be unaffected, as they will be still waiting for their address. Once the entire message is received, the addressed slave should set its SM2 bit to ignore all transmissions until it receives the next address byte.

The Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given Address. All of the slaves may be contacted by using the Broadcast address.

Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. The slave address is an 8-bit value specified in the SADDR register. The SADEN register is actually a mask for the byte value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the slave address in SADDR. Use of the Given Address allows multiple slaves to be recognized while excluding others.

Mnemonic	Slave 1	Slave 2
SADDR	10100100	10100111
SADEN (0 mask)	11111010	11111001
Given Address	10100x0x	10100xx1
Broadcast Address (OR)	1111111x	11111111

The Given address for slave 1 and 2 differ in the LSB. For slave 1, it dosen't care, while for slave 2 it is 1. Thus to communicate only with slave 1, the master must send an address with LSB = 0 (10100000). Similarly the bit 1 is 0 for slave 1 and don't care for slave 2. Hence to communicate only with slave 2 the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. The bit 3 position is don't care for both the slaves. This allows two different addresses to select both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously with the Broadcast Address. This address is formed from the logical OR of the SADDR and SADEN. The zeros in the result are defined as don't cares. In most cases, the Broadcast Address is 0xFFh, this address will be acknowledged by all slaves.

On reset, the SADDR and SADEN are initialized to 00h. This results in Given Address and Broadcast Address being set as XXXXXXXX (all bits don't care). This effectively removes the multiprocessor communications feature, since any selectivity is disabled. This ensures that the EUART will reply to any address, which it is backwards compatible with the 80C51 micro controller that does not support automatic address recognition. So the user may implement multiprocessor by software address recognition mentioned above.

Frame Error Detection

Frame error detection is available in the following modes when the SSTAT bit in register PCON is set to logic 1.

All the 3 bits should be cleared by software after they are set, even when the following frames received without any error will not be cleared automatically.

Note:
The SSTAT bit must be logic 1 to access any of the status bits (FE, RXOVR, and TXCOL). The SSTAT bit must be logic 0 to access the Mode Select bits (SM0, SM1, and SM2).

Transmit collision

The Transmit Collision bit (TXCOL bit in register SCON) reads ‘1’ if user software writes data to the SBUF register while a transmission is still in progress. If this occurs, the new data will be ignored and the transmit buffer will not be written.

Receive Overrun

The Receive Overrun bit (RXOVR in register SCON) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. The previous data is lost when this happen.

The Frame Error bit (FE in register SCON) reads ‘1’ if an invalid (low) STOP bit is detected.

Break Detection

A break is detected when any 11 consecutive bits are sensed low. Since a break condition also satisfies the requirements for a framing error, a break condition will also result in reporting a framing error. Once a break condition has been detected, the EUART will go into an idle state and remain in this idle state until a valid stop bit (rising edge on RxD line) has been received.

8.4 Register

The control and status bits of the UART in special function register SCON are illustrated Table follow.

98H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**SM [0:1]**	SSTAT = 0, the combination of SM0 & SM1 means as below: 00 Serial mode 0: Synchronous Mode, fixed baud rate 01 Serial mode 1: 8 bit Asynchronous Mode, variable baud rate 10 Serial mode 2: 9 bit Asynchronous Mode, fixed baud rate 11 Serial mode 3: 9 bit Asynchronous Mode, variable baud rate
7	**FE**	SSTAT = 1, this bit means Frame Error, ‘1’ means Frame Error occurs. Only clear by software
6	**RXOV**	SSTAT = 1, this bit means Receive run-over, ‘1’ means Receive run-over occurs. Only clear by software
5	**SM2/TXCOL**	SSTAT = 0, this bit Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3 and SM2 being set to 1, RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1 and SM2 = 1, RI will not be activated until a valid stop bit has been received. In mode 0, the serial port runs at 1/12 of the clock when SM2 = 0, the serial port runs at 1/4 of the clock when SM2 = 1 SSTAT = 1, this bit is Transmit Collision index, ‘1’ means Transmit collision occurs Only clear by software
4	**REN**	Receiver enable: Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
3	**TB8**	Transmitter bit 8. Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
2	**RB8**	Receiver bit 8. In modes 2 and 3 it is the 9th bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
1	**TI**	Transmitter interrupt flag: Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
0	**RI**	Receiver interrupt flag. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software.

The data transmitted and received is stored in special function register SBUF. Receive and Writing to SBUF loads the transmit register and initiates transmission. Reading out SBUF accesses a physically separate receive register. Table follow is the data buffer of Serial port

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	SBUF[7-0]	This SFR accesses two registers; a transmit shift register and a receive latch register. A write of SBUF will send the byte to the transmit shift register and then initiate a transmission A read of SBUF returns the contents of the receive latch.

The SMOD of SFR PCON.7 control the double of the baud rate of serial port in Mode 1/2/3, if it is set, the baud rate is doubled.

Baud rate double

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	-	-	R/W	R/w	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	**Baud rate doubler** If set in mode 1 & 3, the baud-rate of EUART is doubled if using timer 1 as baud-rate generator, If set in mode 2, the baud-rate of EUART is doubled
6	**SSTAT**	**SCON [7:5] function selection** 0 = the SCON [7:5] operates as SM0, SM1, SM2 1 = the SCON [7:5] operates as FE, RXOV, TXCOL
3-0	-	**Other bits – see Power Section**

The automatic address recognition feature uses two SFRs as slave’s address, SADDR, and the address mask, SADEN. The actual slave address is defined by the Mask result of SADEN to SADDR. If SADEN is ‘0’, the corresponding bit in SADDR is ignored in actual slave Address.

EUART0 Slave Address & Address Mask register

9AH-9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR	SADD7	SADD6	SADD5	SADD4	SADD3	SADD2	SADD1	SADD0
SADEN	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	SADDR.7-0	UART Slave Address The contents of this register are used to define the UART slave address. Register SADEN is a bit mask to determine which bits of SADDR are checked against a received address: corresponding bits set to logic 1 in SADEN are checked; corresponding bits set to logic 0 are “don’t cares
7-0	SADEN.7-0	SFR SADEN is a bit mask to determine which bits of SADDR are checked against a received address: 0 = Corresponding bit in SADDR0 is a “don’t care”. 1 = Corresponding bit in SADDR 0is checked against a received address.

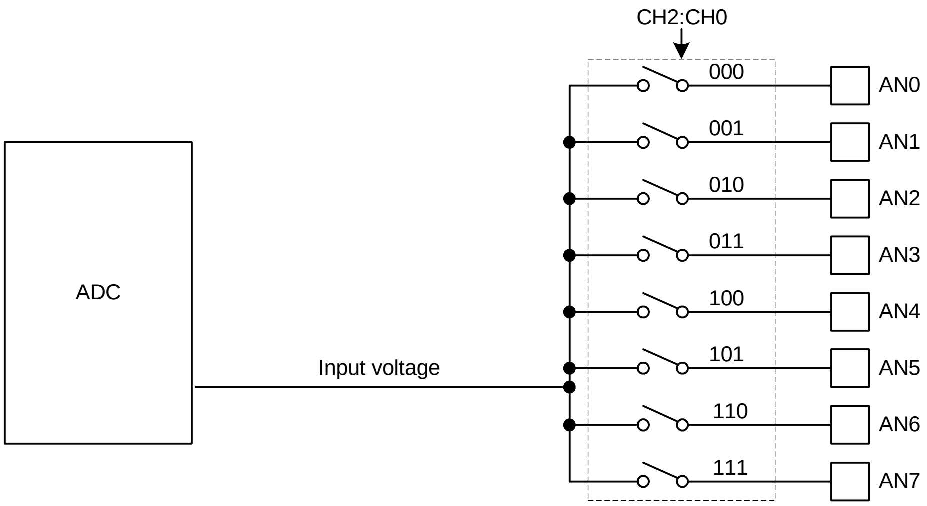
9 Analog Digital Converter (ADC)

ZEN8105 include a single ended, 10-bit SAR Analog to Digital Converter (ADC) with Vref directly connected to Vdd. 8 ADC channels are shared with one ADC module, each channel can be programmed to connect with the analog input individually. Only one channel can be available at one time. GO/$\overline{\text{DONE}}$ signal is available to start convert, and indicate end of convert. When conversion is completed, the data in AD convert data register will be updated and ADCIF bit in ADCON register will be set to generate an interrupt if ADC Interrupt is enabled.

The ADC integrated a digital compare function to compare the value of analog input with the digital value in the AD converter. If this function is enabled (EC =1 in ADCON register) when ADC module is enabled (ADON = 1 in ADCCON register). When the corresponding digital value of analog input is larger than the value in compare value register (ADDH/L), the ADC interrupt will occur, otherwise no interrupt will be generated. The digital comparator can work continuously when GO/$\overline{\text{DONE}}$ bit is set until software clear, which behaviors different with the ADC module.

The ADC module including digital compare module can wok in IDLE mode and be waked up by ADC interrupt, but is disabled in POWER-DOWN mode.

9.1 ADC Diagram



AD Converter Block Diagram

ADC Control Register:

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON	ADON	ADCIF	EC	-	SCH2	SCH1	SCH0	GO/$\overline{\text{DONE}}$
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**ADCON**	**ADC Enable** 0 = Disable the ADC module 1 = Enable the ADC module
6	**ADCIF**	**ADC Interrupt Flag** 0 = No ADC interrupt, cleared by software. 1 = Set by hardware to indicate that the AD Convert has been completed, or analog input is larger than ADDATH/ADDATL if compare is enabled
5	**EC**	**Compare Function Enable** 0 = compare function disabled (default) 1 = compare function enabled.
3-1	**SCH [2:0]**	**ADC channel Select** 000: ADC channel AN0 001: ADC channel AN1 010: ADC channel AN2 011: ADC channel AN3 100: ADC channel AN4 101: ADC channel AN5 110: ADC channel AN6 111: ADC channel AN7
0	**GO/$\overline{\text{DONE}}$**	**ADC status flag** 0 = Automatically cleared by hardware when AD convert is completed. Clearing this bit during converting time will stop current conversion. If Compare function is enabled, this bit will not be cleared by hardware until software clear. 1 = Set to start AD convert.

ADC Channel Configuration Register:

95H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCH	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**CH [7:0]**	**Channel Configuration** 1 = use P0x (x=2~5) or P1x (x=2~5) as ADC input (P0x or P1x haven't been set as other function port). 0 = use P0x (x=2~5) or P1x (x=2~5) as standard I/O port or other function port.

ADC Time Select Register

94H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADT	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-5	**TADC [2:0]**	**TADC [2:0] – ADC Clock Period Select** 000 = ADC Clock Period t_{AD} = 2 t_{SYS} 001 = ADC Clock Period t_{AD} = 4 t_{SYS} 010 = ADC Clock Period t_{AD} = 6 t_{SYS} 011 = ADC Clock Period t_{AD} = 8 t_{SYS} 100 = ADC Clock Period t_{AD} = 12 t_{SYS} 101 = ADC Clock Period t_{AD} = 16 t_{SYS} 110 = ADC Clock Period t_{AD} = 24 t_{SYS} 111 = ADC Clock Period t_{AD} = 32 t_{SYS}
3-0	**TS [3:0]**	**Sample time select** Sample time = (TS [3:0]+1) * T_{AD} < 16 T_{AD}

Note:

1. Make sure that t_{AD} ≧1us

2. The minimal Sample Time is 2T_{AD}, even TS[3:0] = 000;
The maxim Sample Time is 15 T_{AD} , even TS[3:0] = 111;

3. Evaluate the series resistance connected with ADC input pin before set TS[3:0]

*4. Be sure that the series resistance connected with ADC input pin is no more than 10KΩ. When 2*T_{AD} sample time, with VDD between 4.5V-5.5V*

5. Total conversion time is: conversion time (12T_{AD}) + sample time

ADC Data Register (Low Byte):

96H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDL	-	-	-	-	-	-	A1	A0
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	-	-	-	-	0	0
97H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
ADDH	A9	A8	A7	A6	A5	A4	A3	A2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
1-0	**A9-A0**	Digital Value of sampled analog voltage, updated when conversion is completed If ADC Digital Compare function is enabled (EC = 1), this is the value to be compared with the analog input
7-0		

10 Interrupt

10.1 Features

The 88F2049 provides total 13 interrupt sources: one OVL NMI interrupt, five external interrupts (INT0/1/2/3/4; INT4 including INT40-47, which share the same vector address), three timer interrupts (timer 0, 1 and 2), one EUART interrupt, ADC Interrupt, SPI Interrupt, 3 PWM timer interrupts which share the same entrance vector address. The ZEN8105 uses a four-priority level interrupt structure. This allows great flexibility in controlling the handling of the internal interrupt sources.

Also, ZEN8105 provides 4 ways to trigger external interrupt 2, 3, 4, which can be selected by register.

10.2 Program Over Range interrupt (OVL)

The ZEN8105 also has a NMI interrupt source (OVL), whose vector is located in 0x007B, this NMI is used to prevent CPU run out of valid program range. To enable this feature, the user should fill in the unused flash ROM with constant byte 0xA5.

If PC exceeds the valid program range, the operation code will be 0xA5, which is not exist in 8051 instruction set, so the CPU will know the PC is out of valid program range, and the OVL NMI will generate. Also if PC exceeds 16K ROM range, the OVL NMI will also be generated.

The OVL NMI has the highest priority (except RESET), and cannot be interrupted by other interrupt source. Also the OVL NMI can be nested by itself, but the stack will not increase since it is useless to push the stack when PC is invalid. When OVL NMI happened, the other interrupt are still enabled, and their flag will be set if required condition is met.

The user must process this interrupt to protect their system from unwanted execution result. They can modify the top of stack (since this stack top address is a useless one), with a RETI instruction at the end of NMI Interrupt vector service. This two operation will make the program jump to the code the user want to be processed, such as reset entry or protection process entry.

```
OVL_NMI_SERVICE:
   MOV DPTR,  #PROTECT_CODE_ENTRANCE
      POP  A
POP        A
      PUSH DPL
      PUSH DPH
RETI
```

10.3 Interrupt Enable control

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable registers IEN0 or IEN1. The IEN0 register also contains a global interrupt enable bit, EA, which can enable/disable all interrupts at once. Generally, after reset, all interrupt enable bits are set to 0, which means that the corresponding interrupts are disabled.

Primary Interrupt Enable Register0

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	EADC	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**EA**	**All interrupt enable bit** 0 = Disable all interrupt 1 = Enable all interrupt
6	**EADC**	**ADC interrupt enable bit** 0 = Disable ADC interrupt 1 = Enable ADC interrupt
5	**ET2**	**Timer2 overflow interrupt enable bit** 0 = Disable timer 2 overflow interrupt 1 = Enable timer 2 overflow interrupt
4	**ES0**	**EUART0 interrupt enable bit** 0 = Disable EUART interrupt 1 = Enable EUART interrupt
3	**ET1**	**Timer1 overflow interrupt enable bit** 0 = Disable timer 1 overflow interrupt 1 = Enable timer 1 overflow interrupt
2	**EX1**	**External interrupt 1 enable bit** 0 = Disable external interrupt 1 1 = Enable external interrupt 1
1	**ET0**	**Timer0 overflow interrupt enable bit** 0 = Disable timer 0 overflow interrupt 1 = Enable timer 0 overflow interrupt
0	**EX0**	**External interrupt 0 enable bit** 0 = Disable external interrupt 0 1 = Enable external interrupt 0

Primary Interrupt Enable Register1

A9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	-	-	EPWM	-	EX4	EX3	EX2	ESPI
R/W	-	-	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
5	**EPWM**	**PWM interrupt enable bit** 0 = Disable PWM interrupt 1 = Enable PWM interrupt
3	**EX4**	**External interrupt4 enable bits.** 0: Disable external interrupt 4 1: Enable external interrupt 4
2	**EX3**	**External interrupt3 enable bits.** 0: Disable external interrupt 3 1: Enable external interrupt 3
1	**EX2**	**External interrupt2 enable bits.** 0: Disable external interrupt 2 1: Enable external interrupt 2
0	**ESPI**	**SPI interrupt enable bit** 0 = Disable SPI interrupt 1 = Enable SPI interrupt

NOTE:

To enable External interrupt0/1/4, the corresponding port must be set to input mode before using it.
To enable PWM timer interrupt, the EPWM bit here should be set. Also, the PWMxIE bit in PWM control register should be set. (x=0,1,2)

Interrupt channel Enable Register

BAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IENC	EXS47	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	EXS4x. (x=0-7)	**External interrupt4 channel select bit.(x=0-7)** 0: Disable external interrupt 4x 1: Enable external interrupt 4x

10.4 Interrupt Flag

Each Interrupt source has its own interrupt flag, when interrupt occurs, corresponding flag will be set by hardware, the interrupt flag bits are listed in Table bellow

For **external interrupt (INT0/1/2/3/4)**, when an external interrupt0/1/2/3 is generated, if the interrupt was edge trigged, the flag (IE0/1in TCON and IE2/3 in IXF0 register) that generated this interrupt is cleared by hardware when the service routine is vectored. If the interrupt was level trigged, then the requesting external source directly controls the request flag, rather than the on-chip hardware.

When an external interrupt4 is generated, the flag (IF4x in IXF1 register) that generated this interrupt should be cleared by user's program because the same vector entrance was used in INT4. But if INT4 is setup as level trigged, the flag can't be cleared by user's program, it only be controlled by peripheral signal level that connect to INT source pin.

Note that external interrupt flags are always changed according to the state of external interrupt inputs even if the external interrupts are disabled, unless the corresponding pin is shared as other function.

For **timer 0 /1 interrupt,** when timer interrupt is generated, the flag that generated is cleared by the on-chip hardware when the service routine is vectored.

The **timer 2 interrupt** is generated by the logical OR of flag TF2 and bit EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the flag will have to be cleared by software.

The **EUART interrupt** is generated by the logical OR of flag RI and TI in register SCON. Neither of these flags is cleared by hardware when the service routine is vectored. In fact, the service routine will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, and the flag can be cleared by software.

The **ADC interrupt** is generated by ADCIF bit in ADCON. If an interrupt is generated, the converted result in ADCDH/ADCDL will be valid. If continuous conversion is established, ADCIF is set at each conversion. If an ADC interrupt is generated, the flag can be cleared by software.

The **SPI interrupt** is generated by SPIF or MODF (when SSDIS bit is 0) in SPSTA. The flag can be cleared by software.

The **PWM interrupts** are generated by PWMIF0-2. The flags can be cleared by software. The PWM interrupt flags are listed in section PWM timer.

External Interrupt Flag Register

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
3,1	IEx	**External interrupt x request flag (x=0,1)** 0: No interrupt pending 1: Interrupt is pending
2,0	ITx	**External interrupt x trigger mode (x=0,1)** 0: Low level trigger 1: Falling edge trigger

External Interrupt2, 3, 4 Register

E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF0	IT4.1	IT4.0	IT3.1	IT3.0	IT2.1	IT2.0	IE3	IE2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	IT4 [1:0]	**External interrupt 4 trigger mode** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge IT4 [1:0] is effect on external interrupt 4x at the same mode

5-4	IT3 [1:0]	External interrupt 3 trigger mode 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge
3-2	IT2 [1:0]	External interrupt 2 trigger mode 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge
1-0	IEx (x=2,3)	External interrupt x request flag (x=2,3) 0: No interrupt pending 1: Interrupt is pending

External Interrupt4 flag Register

D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF1	IF47	IF46	IF45	IF44	IF43	IF42	IF41	IF40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	IF4x (x=0-7)	**External interrupt4 request flag** 0: No interrupt pending 1: Interrupt is pending IF4x is cleared by software.

10.5 Interrupt Vector

When an interrupt occurs, the program counter is pushed onto the stack and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are listed in Interrupt Summary table.

10.6 Interrupt Priority

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority control registers IPL0, IPH0, IPL1, and IPH1. The interrupt priority service is described below:

An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority.

The highest priority interrupt service cannot be interrupted by any other interrupt source. If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are pending at the start of an instruction cycle, an internal polling sequence determines which request is serviced.

Interrupt Priority Level		
Priority bits		Interrupt Level Priority
IPHx	IPLx	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

Interrupt priority control registers

B8H, B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IPL0	-	PADCL	PT2L	PUL	PT1L	PX1L	PT0L	PX0L
IPH0	-	PADCH	PT2H	PUH	PT1H	PX1H	PT0H	PX0H
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	0	0	0	0	0	0	0
B9H, B5H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IPL1	-	-	PPWML	-	PX4L	PX3L	PX2L	PSPIL
IPH1	-	-	PPWMH	-	PX4H	PX3H	PX2H	PSPIH
R/W	-	-	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	-	-	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
-	PxxxL/H	Corresponding interrupt source xxx's priority level select

10.7 Interrupt Handling

The interrupt flags are sampled and polled at the fetch cycle of each machine cycle. All interrupts are sampled at the rising edge of the clock. If one of the flags was set, the CPU will find it and the interrupt system will generate a LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:
An interrupt of equal or higher priority is already in progress.

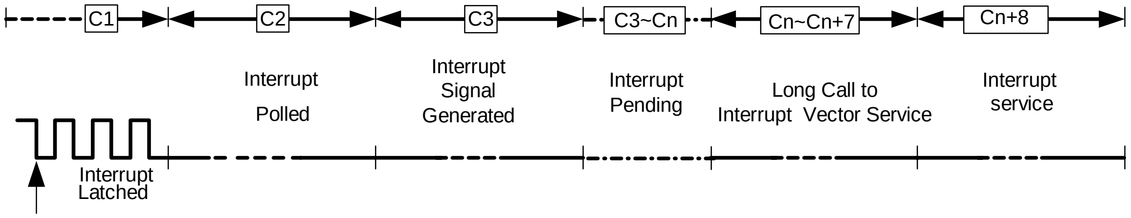
The current cycle is not in the final cycle of the instruction in progress. This ensures that the instruction in progress is completed before vectoring to any service routine.

The instruction in progress is RETI. This ensures that if the instruction in progress is RETI then at least one more instruction except RETI will be executed before any interrupt is vectored to; this delay guarantees that the CPU can observe the changes of the interrupt status.

***Note:** Since priority change normally needs 2 instructions, it is recommended to disable corresponding Interrupt Enable flag to avoid interrupt between the these 2 instructions during the change of priority.*

Note that if the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. Every polling cycle interrogates only the valid interrupt requests.

The polling cycle/LCALL sequence is illustrated below



Interrupt Response Timing

The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with corresponding address that depends on the source of the interrupt being vectored too, as shown in Interrupt Summary table.

Interrupt service execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt service. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt with this priority was still in progress. In this case, no interrupt of the same or lower priority level would be acknowledged.

10.8 Interrupt Response Time

If an interrupt is recognized, its request flag is set in every machine cycle after recognize. The value will be polled by the circuitry until the next machine cycle. The CPU will generate an interrupt at the third machine cycle. If the request is active and conditions are right for it to be acknowledged, a hardware-generated LCALL to the requested service routine will be the next instruction to be executed. Else the interrupt will pending. The call itself takes 7 machine cycles. Thus a minimum of 3+7 complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

A longer response time would be obtained if the request was blocked by one of the above three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine.

If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 21 machine cycles since the longest instructions (DIV & MUL) are only 20 machine cycles long for 16 bit operation or (11 machine cycles for 8 bit operation); and, if the instruction in progress is RETI or the additional wait time cannot be more than 8+21 machine cycles (a maximum of one more cycle to complete the instruction in progress, plus maximal 20 machine cycles to complete the next instruction, if the instruction is 16bit DIV or MUL). Thus a single interrupt system, the response time is always more than 10 machine cycles and less than 8+21 machine cycles.

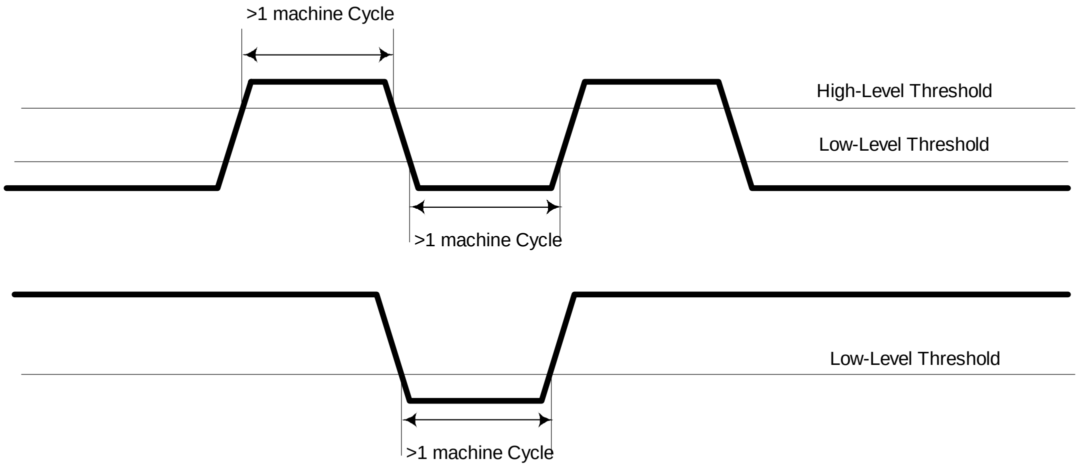
10.9 External Interrupt inputs

The ZEN8105 has 5 external interrupt inputs. External interrupt0-3 each has one vector address. External interrupt 4 has 8 inputs; all of them share one vector address. These external interrupts can be programmed to be level-triggered or edge-triggered by clearing or setting bit IT1 or IT0 in register TCON and register EXF0. If ITn = 0(n=0~1), external interrupt 0/1 is triggered by a low level detected at the INT0/1 pin. If ITn =1(n=0~1), external interrupt 0/1 is edge triggered. In this mode if consecutive samples of the INT0/1 pin show a high level in one cycle and a low level in the next cycle, interrupt request flag in register EXF0 or EXF1 is set, causing an interrupt request. Since the external interrupt pins are sampled once each machine cycle, an input high or low level should be held for at least one machine cycle to ensure proper sampling.

If the external interrupt is edge-triggered, the external source has to hold the request pin high for at least one machine cycle, and then hold it low for at least one machine cycle. This is to ensure that the transition is detected and that interrupt request flag is set. Notice that IE0-3 is automatically cleared by CPU when the service routine is called while IF40-47 should be cleared by software. External interrupt2-4 operates in the similar ways except have different registers and have more selection of trigger.

If the external interrupt is level-triggered, the external source must hold the request active until the requested interrupt is generated, which will take 2 machine cycles. If the external interrupt is still asserted when the interrupt service routine is completed, another interrupt will be generated. It is not necessary to clear the interrupt flag IEx (x=0,1,2,3,4) when the interrupt is level sensitive, it simply tracks the input pin level.

If an external interrupt is enabled when the ZEN8105 is put into Power down or Idle mode, the interrupt event will cause the processor to wake up and resume operation.



External Interrupt Detection

10.10 Interrupt Summary

Interrupt Summary				
Source	**Vector Address**	**Enable bits**	**Flag bits**	**Polling Priority**
Reset	0000h			0 (highest)
INT0	0003h	EX0	IE0	2
Timer0	000Bh	ET0	TF0	3
INT1	0013h	EX1	IE1	4
Timer1	001Bh	ET1	TF1	5
EUART	0023h	ES	RI+TI	6
Timer2	002Bh	ET2	TF2+EXF2	7
ADC	0033h	EADC	ADCIF	8
SPI	003Bh	ESPI	SPIF	9
INT2	0043h	EX2	IE2	10
INT3	004Bh	EX3	IE3	11
INT4	0053h	EX4+IENC	IF47-40	12
PWM	0063h	EPWM+PWM0/1/2IE	PWM0/1/2IF	13
OVL NMI	007Bh	-	-	1

11 Low Voltage Reset (LVR)

The LVR function is to monitor the supply voltage and generate an internal reset in the device when the voltage below the detect value. The LVR de-bounce Timer is about 30~60 us

The LVR function is enabled by the code option.

The LVR circuit has the following functions when the LVR function is enabled:

- Generates a system reset when $V_{DD} \leq V_{LVR}$ and $t \geq T_{LVR}$.
- Cancels the system reset when $V_{DD} > V_{LVR}$ or $V_{DD} < V_{LVR}$ but $t < T_{LVR}$.

It is typically used in AC line or large battery supplier applications, where heavy loads may be switched on and cause the MCU supply-voltage temporarily fall below the minimum specified operating voltage. This feature is can protect system from working under bad power supply environment.

12 Watchdog Timer (WDT) and reset state

The watchdog timer is a countdown counter, and its clock source is an independent built-in RC oscillator, so it will always run even in the Power-Down mode. The watchdog timer automatically generates a device reset when it overflows. It can be enabled or disabled permanently by using the code option.

The watchdog timer control bits (WDT.2-0) are used to select different overflow frequency. The watchdog timer overflow flag (WDOF) will be automatically set to “1” by hardware when the watchdog timer overflows. By reading or writing the WDT register RSTSTAT, the watchdog timer should re-count before the overflow happens.

There are also some reset flags in this register as listed below.

B1H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
RSTSTAT	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	-	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	WDOF	**WDT Overflow Flag** Set when the down counter overflows, cleared by software or Power On Reset.
5	PORF	Set by Power On Reset, cleared only by software
4	LVRF	Set by LVR Reset, cleared by software or Power On Reset
3	CLRF	Set by pin reset, cleared by software or Power On Reset
2-0	WDT.2-0	**WDT.2-0** WDT timer-out period control 000: 4096ms 001: 1024ms 010: 256ms 011: 128ms 100: 64ms 101: 16ms 110: 4ms 111: 1ms

13 Power Management

Two power reduction modes are implemented in the ZEN8105: the Idle mode and the Power-Down mode. These two modes are controlled by PCON & SUSLO register.

13.1 Idle mode

Idle mode is a power reduction mode that reduces the power consumption. In this mode, program execution halts. Idle mode freezes the clock to the CPU at known states while the peripherals continue to be clocked. The CPU status before entering Idle mode is preserved, i.e., the program counter and program status word register retain their data for the duration of Idle mode. The contents of the SFRs and RAM are also retained.

By two consecutive instructions: setting SUSLO register as 55H, and immediately followed by setting the IDL bit in PCON register, will make ZEN8105 enter idle mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or IDL bit in the next machine cycle. And the CPU will not enter IDLE mode.

The setting of IDL bit will be the last instruction that CPU executed.

There are two ways to exit idle mode:

1) An interrupt generated. The clock to the CPU will be restored, and the hardware will clear SUSLO register and IDL bit in PCON register. Then the program will execute the interrupt service routine first, and then jumps to the instruction immediately following the instruction that activated idle mode.
2) Reset signal (logic LOW on the $\overline{\text{RST}}$ pin, WDT RESET if enabled, LVR RESET if enabled), this will restore the clock to the CPU, the SUSLO register and the IDL bit in PCON register will be cleared by hardware, finally the ZEN8105 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

13.2 Pow er-Down mode

The Power-Down mode places the ZEN8105 in a very low power state. Power-Down mode will stop all the clocks including CPU and peripherals. If WDT is enabled, WDT block will keep on working. When entering Power-Down mode, all the CPU status before entering will be preserved. Such as: PSW, PC, SFR & RAM are all be retained

By two consecutive instructions: setting SUSLO register as 55H, and immediately followed by setting the PD bit in PCON register, will make ZEN8105 enter Power-Down mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or PD bit in the next machine cycle. And the CPU will not enter Power-Down mode.

The setting of PD bit will be the last instruction that CPU executed.

Note:

If IDL bit and PD bit are set simultaneously, the SH88F2049 enters Power-Down mode. The CPU will not go in idle mode when exiting from Power-Down mode, and the hardware will clear both IDL & PD bit after exit from Power-Down mode.

There are two ways to exit the Power-Down mode:

1) An active external Interrupt such as INT0, INT1.etc will make ZEN8105 exit Power-Down mode. The oscillator will start after interrupt happens, after warm-up time, the clocks to the CPU and peripheral will be restored, the SUSLO register and PD bit in PCON register will be cleared by hardware. Program execution resumes with the interrupt service routine. After completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.
2) Reset signal (logic LOW on the $\overline{\text{RST}}$ pin, WDT RESET if enabled, LVR RESET if enabled). This will restore the clock to the CPU after warm-up time, the SUSLO register and the PD bit in PCON register will be cleared by hardware, finally the ZEN8105 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

In order to entering IDLE/POWER-DOWN, it is necessary to add 3 NOP after setting IDL/PD bit in PCON.

Register Power Control Register

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	SMOD	**Serial port Mode** Set to select double baud rate in mode 1, 2 or 3.
6	SSTAT	**SSTAT** 0: the SCON [7:5] operates as SM0, SM1, SM2 1: the SCON [7:5] operates as FE, RXOV, TXCOL
3	GF1	These are General purpose Flags for use under software control.
2	GF0	
1	PD	**Power-Down Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Power-Down mode. If IDL and PD are both set, PD takes precedence.
0	IDL	**Idle Mode bit** Cleared by hardware when an interrupt or reset occurs. Set to activate the Idle mode. If IDL and PD are both set, PD takes precedence

Suspend Mode Control Register

8EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SUSLO	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	SUSLO.7-0	**Description:** This register is used to control the CPU enter suspend mode (IDLE or Power-Down). Only consecutive instructions like below will make CPU enter suspend mode. Otherwise the either SUSLO, IDL or PD bit will be cleared by hardware in the next machine cycle. **Enter IDLE mode:** *MOV SUSLO, #55H* *ORL PCON, #01H* **Enter Power-Down mode:** *MOV SUSLO, #55H* *ORL PCON, #02H*

Note: In Idle mode, only the sysck to cpu isstopped, the peripheral block such as timer, ADC etc still have clock source provided (even though it's clock source is sysck), so it's possible to use timer, ADC etc to wake up Idle mode. But in power down mode, all clocks are stopped.

14 Warm-up Timer

The device has a built-in warm-up timer, it is designed to eliminate unstable state when oscillator starts oscillating in the following conditions: Power-on reset, Pin reset, LVR reset Watchdog Reset and Wake up from Power-down mode

14.1 Warm-up time interval:

In RC oscillator mode, the warm-up counter prescaler is divided by 2^7 (128).
In Ceramic resonator mode, the warm-up counter prescaler is divided by 2^{12} (4096).
Note: When power-on starts, it will spend about 6ms on the internal regulator stabilization at first. And the following is the warm-up that can eliminate unstable state of initial oscillation.

OSC type	**Power On Reset / Low Voltage Reset**		**Pin Reset**		**WDT reset**		**Wakeup from STOP**	
	T**PWRT ***	Warm up	T**PWRT ***	Warm up	T**PWRT ***	Warm up	T**PWRT ***	Warm up
Ceramic	6ms	4096	6ms	4096	2ms	0	2ms	4096
Internal RC	6ms	128	6ms	128	2ms	0	2ms	128

15 Code Option

OP_RST [5]:
- 0: Pin8 used as RST pin (default)
- 1: Pin8 used as I/O pin

OP_LVREN [4]:
- 0: Disable LVR function (default)
- 1: Enable LVR function

OP_WDT [3]:
- 0: Disable WDT function(default).
- 1: Enable WDT function

OP_OSC [2:0]:
- 000: Internal RC oscillator (default)
- 101: Ceramic resonator
- 110: Ceramic resonator
- Others: Internal RC oscillator

16 Instruction Set

ARITHMETIC OPERATIONS				
Opcode	Description	Code	Byte	Cycle
ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A,direct	Add direct byte to accumulator	0x25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A,#data	Add immediate data to accumulator	0x24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A,#data	Add immediate data to A with carry flag	0x34	2	2
SUBB A,Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment direct byte	0x05	2	3
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement direct byte	0x15	1	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	2	3
INC DPTR	Increment data pointer	0xA3	1	4
MUL AB 8 x 8 16 x 8	Multiply A and B	0xA4	1	11 20
DIV AB 8 / 8 16 / 8	Divide A by B	0x84	1	11 20
DA A	Decimal adjust accumulator	0xD4	1	1

LOGIC OPERATIONS				
Opcode	Description	**Code**	**Byte Cy**	**cle**
ANL A,Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A,direct	AND direct byte to accumulator	0x55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A,#data	AND immediate data to accumulator	0x54	2	2
ANL direct,A	AND accumulator to direct byte	0x52	2	3
ANL direct,#data	AND immediate data to direct byte	0x53	3	3
ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A,direct	OR direct byte to accumulator	0x45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A,#data	OR immediate data to accumulator	0x44	2	2
ORL direct,A	OR accumulator to direct byte	0x42	2	3
ORL direct,#data	OR immediate data to direct byte	0x43	3	3
XRL A,Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	0x65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	0x66-0x67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	0x62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	4

BOOLEAN MANIPULATION				
Opcode	Description	**Code**	**Byte Cy**	**cle**
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear direct bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement direct bit	0xB2	2	3
ANL C,bit	AND direct bit to carry flag	0x82	2	2
ANL C,/bit	AND complement of direct bit to carry	0xB0	2	2
ORL C,bit	OR direct bit to carry flag	0x72	2	2
ORL C,/bit	OR complement of direct bit to carry	0xA0	2	2
MOV C,bit	Move direct bit to carry flag	0xA2	2	2
MOV bit,C	Move carry flag to direct bit	0x92	2	3

DATA TRANSFERS				
Opcode	Description	**Code**	**Byte Cy**	**cle**
MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A,direct	Move direct byte to accumulator	0xE5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2
MOV A,#data	Move immediate data to accumulator	0x74	2	2
MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	2
MOV Rn,direct	Move direct byte to register	0xA8-0xAF	2	3
MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
MOV direct,A	Move accumulator to direct byte	0xF5	2	2
MOV direct,Rn	Move register to direct byte	0x88-0x8F	2	2
MOV direct1,direct2	Move direct byte to direct byte	0x85	3	3
MOV direct,@Ri	Move indirect RAM to direct byte	0x86-0x87	2	3
MOV direct,#data	Move immediate data to direct byte	0x75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	0xF6-0xF7	1	2
MOV @Ri,direct	Move direct byte to indirect RAM	0xA6-0xA7	2	3
MOV @Ri,#data	Move immediate data to indirect RAM	0x76-0x77	2	2
MOV DPTR,#data16	Load data pointer with a 16-bit constant	0x90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to A	0x93	1	7
MOVC A,@A+PC	Move code byte relative to PC to A	0x83	1	8
MOVX A,@Ri	Move external RAM (8-bit address) to A	0xE2-0xE3	1	5
MOVX A,@DPTR	Move external RAM (16-bit address) to A	0xE0	1	6
MOVX @Ri,A	Move A to external RAM (8-bit address)	0xF2-F3	1	4
MOVX @DPTR,A	Move A to external RAM (16-bit address)	0xF0	1	5
PUSH direct	Push direct byte onto stack	0xC0	2	5
POP direct	Pop direct byte from stack	0xD0	2	4
XCH A,Rn	Exchange register with accumulator	0xC8-0xCF	1	3
XCH A,direct	Exchange direct byte with accumulator	0xC5	2	4
XCH A,@Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	4
XCHD A,@Ri	Exchange low-order nibble indirect RAM with A	0xD6-0xD7	1	4

PROGRAM BRANCHES				
Opcode	Description	**Code**	**Byte Cy**	**cle**
ACALL addr11	Absolute subroutine call	0x11-0xF1	2	7
LCALL addr16	Long subroutine call	0x12	3	7
RET	Return from subroutine	0x22	1	8
RETI	Return from interrupt	0x32	1	8
AJMP addr11	Absolute jump	0x01-0xE1	2	4
LJMP addr16	Long jump	0x02	3	5
SJMP rel	Short jump (relative address)	0x80	2	4
JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	6
JZ rel (not taken) (taken)	Jump if accumulator is zero	0x60	2	3 5
JNZ rel (not taken)	Jump if accumulator is not zero	0x70	2	3

(taken)				5
JC rel (not taken) (taken)	Jump if carry flag is set	0x40	2	2 4
JNC rel (not taken) (taken)	Jump if carry flag is not set	0x50	2	2 4
JB bit,rel (not taken) (taken)	Jump if direct bit is set	0x20	3	4 6
JNB bit,rel (not taken) (taken)	Jump if direct bit is not set	0x30	3	4 6
JBC bit, rel (not taken) (taken)	Jump if direct bit is set and clear bit	0x10	3	4 6
CJNE A,direct rel (not taken) (taken)	Compare direct byte to A and jump if not equal	0xB5	3	4 6
CJNE A,#data rel (not taken) (taken)	Compare immediate to A and jump if not equal	0xB4	3	4 6
CJNE Rn,#data,rel (not taken) (taken)	Compare immediate to reg. And jump if not equal	0xB8-0xBF	3	4 6
CJNE @Ri,#data, rel (not taken) (taken)	Compare immediate to Ri and jump if not equal	0xB6-0xB7	3	4 6
DJNZ Rn,rel (not aken) (taken)	Decrement register and jump if not zero	0xD8-0xDF	2	3 5
DJNZ direct,rel (not taken) (taken)	Decrement direct byte and jump if not zero	0xD5	3	4 6
NOP	No operation	0	1	1

Electrical Characteristics

Absolute Maximum Ratings*

DC Supply Voltage . . . . –0.3V to +7.0V

Input / Output Voltage . . . . .GND-0.3V to VDD+0.3V

Operating Ambient Temperature . . . .-40℃ to +85℃
(exclude Flash SSP operation)

Flash Program Temperature………..-0℃ to +85℃

Storage Temperature . . . . –55℃ to +125℃

***Comments**

Stresses exceed those listed under “**Absolute Maximum Ratings**” may cause permanent damage to this device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied or intended. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability

DC Electrical Characteristics (VDD=4.0 – 5.5V, GND = 0V, TA = -40°C to +85°C, unless otherwise specified)

Parameter Sy	mbol	Min.	Typ. ∗	Max.	Unit	Condition
Operating Voltage	V_{DD}	4.0	5.0	5.5	V	400K ≤ f_{OSC} ≤ 12MHz
Operating Current	I_{OP}	-	8	15	mA	f_{OSC} = 12MHz, V_{DD}=5.0V All output pins unload (including all digital input pins unfloating) CPU on (execute NOP instruction), WDT on, all other function block on
Stand by Current (IDLE)	I_{SB1}	-	3	5	mA	f_{OSC} = 12MHz , V_{DD}=5.0V All output pins unload, CPU off (IDLE), WDT off, LVR on, all other function block off
Stand by Current (Power-Down)	I_{SB2}	-	5	10	uA	Osc off, V_{DD}=5.0V All output pins unload, CPU off (Power-Down), WDT off, LVR on, all other function block off
WDT Current	I_{WDT}	-	1	3	uA	All output pins unload, WDT on, V_{DD}=5.0V
Input Low Voltage 1	V_{IL1}	GND	-	0.3 X V_{DD}	V	I/O Ports
Input High Voltage 1	V_{IH1}	0.7 X V_{DD}	-	V_{DD}	V	I/O Ports
Input Low Voltage 2	V_{IL2}	GND	-	0.2 X V_{DD}	V	$\overline{\text{RST}}$, T0, T1, T2, INT0/1/2/3/4, SCK, T2EX, RXD, TXD, $\overline{\text{SS}}$ (rigger trigger built-in), FLT, MISO, MOSI
Input High Voltage 2	V_{IH2}	0.8 X V_{DD}	-	V_{DD}	V	$\overline{\text{RST}}$, T0, T1, T2, INT0/1/2/3/4, SCK, T2EX, RXD, TXD, $\overline{\text{SS}}$ (schmit- trigger built-in), FLT, MISO, MOSI
Input Leakage Current	I_{IL}	-1	-	1	uA	Input pad, V_{IN} = V_{DD} or GND,
Output Leakage Current	I_{OL}	-1	-	1	uA	Open drain output, V_{DD} = 5.0V V_{OUT} = V_{DD} or GND
Pull-high Resistor	R_{PH}	-	20	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Output High Voltage	V_{OH}	V_{DD} – 0.7	-	-	V	I/O Ports, I_{OH} = -10mA, V_{DD} = 5.0V
Output Low Voltage	V_{OL1}	-	-	GND + 0.6	V	I/O Ports, I_{OL} = 15mA, V_{DD} = 5.0V
Internal regulator output	Vc	2.8	2.9	3.0	V	C pin output voltage. No load on regulator

∗: Data in “Typ.” Column is at 5.0V, 25℃,unless otherwise specified.
Maximum value of the supply current to VDD is 100mA.
Maximum value of the output current from GND is 150mA.

5V A/D Converter Electrical Characteristics

Parameter Sy	mbol	Min.	Typ.	Max.	Unit	Condition
Supply voltage	V_{AD}	4.5	5.0	5.5	V	
Resolution	NR	-	10	-	bit	GND ≤ V_{AIN} ≤ V_{REF}
A/D Input Voltage	V_{AIN}	GND	-	V_{REF}	V	
A/D Input Resistor*	R_{AIN}	2		-	MΩ	V_{IN}=5.0V
Recommended impedance of analog voltage source	Z_{AIN}	-	-	10	KΩ	
A/D conversion current	I_{AD}	-	1	3	mA	ADC module operating, V_{DD}=5.0V
A/D Input current	I_{ADIN}	-	-	10	uA	V_{DD} = 5.0V
Differential linearity error	DLE	-	-	±1	LSB	Fosc=12MHz, V_{DD} = 5.0V
Integral linearity error	I_{LE}	-	-	±2	LSB	Fosc=12MHz, V_{DD} = 5.0V
Full scale error	E_F	-	±1	±3	LSB	Fosc=12MHz, V_{DD} = 5.0V
Offset error	E_Z	-	±0.5	±2	LSB	Fosc=12MHz, V_{DD} = 5.0V
Total Absolute error	E_{AD}	-	-	±3	LSB	Fosc=12MHz, V_{DD} = 5.0V
Total Conversion time**	TCON	14	-	-	T_{AD}	10 bit resolution, V_{DD} = 5.0V

*: Here the A/D input Resistor is the DC input-resistance of A/D itself.

**: Here the resistance of the device connected with AD should be less than 10KΩ

AC Electrical Characteristics

VDD = 4.0V ~ 5.5V, GND = 0V, TA = -40°C to +85°C, FOSC = 12MHz, unless otherwise specified.

Parameter Sy	mbol	Min.	Typ.	Max.	Unit	Condition
RESET pulse width	t_{RESET}	10	-	-	us	Low active
RESET pin Pull-high Resistor	R_{RPH}	-	100	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
WDT Period	T_{WDT}	0.8	-	-	ms	
RC frequency	FRC	11.64	12.0	12.36	MHz	V_{DD} = 5V
Frequency Stability (RC)	|Δ F|/F	-	0.5	-	%	RC Oscillator: |F (5.0V)-F (4.0V)|/F (V_{DD} = 5V, T_A = 25℃)

Low Voltage Reset Electrical Characteristics

VDD = 4.0V ~ 5.5V, GND = 0V, TA = +25°C, unless otherwise specified.

Parameter Sy	mbol	Min.	Typ.	Max.	Unit	Condition
LVR Voltage	V_{LVR}	3.0	3.1	3.2	V	LVR enabled VDD =4.0V ~ 5.5V FOSC = 12MHz
Drop-Down Pulse Width for LVR	T_{LVR}	-	30	-	us	FOSC = 12MHz

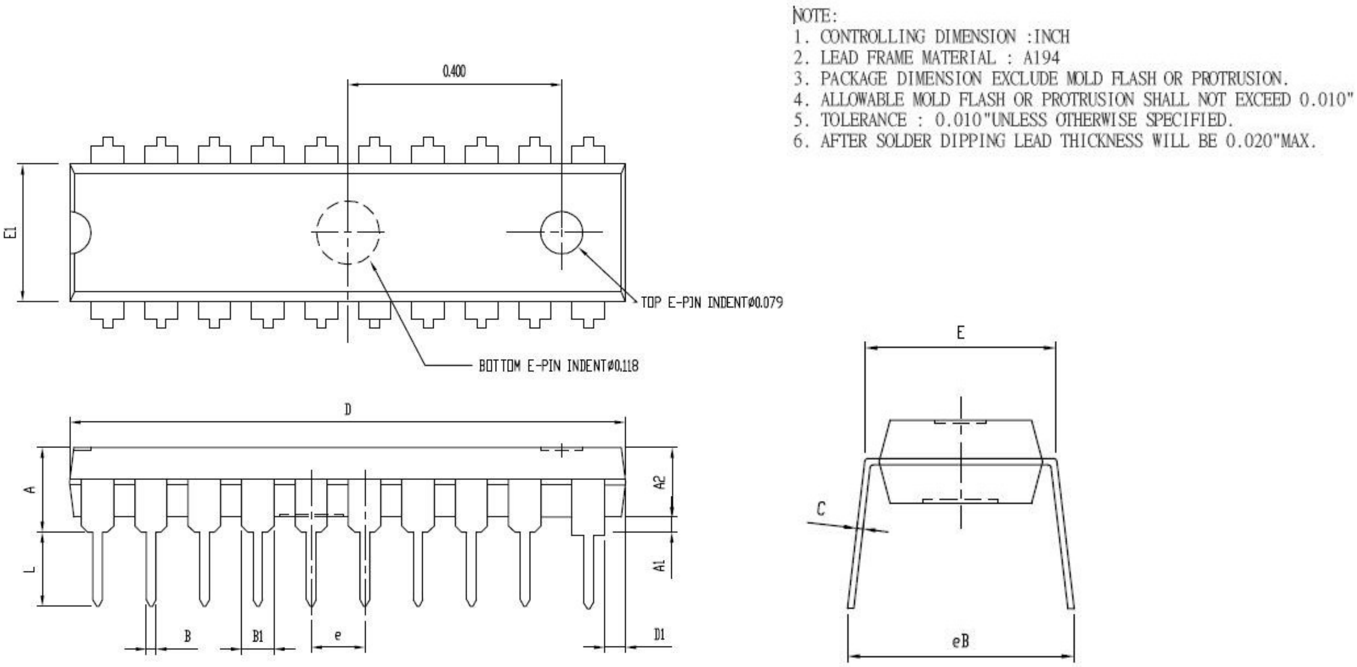
12MHz Ceramic Characteristics

Parameter Sy	mbol	Min.	Typ.	Max.	Unit	Condition
Frequency	F12M	-	12	-	MHz	
Load Capacitance	CL	-	20		pF	

Ordering Information

Part No.	Package
ZEN8105	20 DIP

20L 300mil PDIP Information



NOTE:
1. CONTROLLING DIMENSION :INCH
2. LEAD FRAME MATERIAL : A194
3. PACKAGE DIMENSION EXCLUDE MOLD FLASH OR PROTRUSION.
4. ALLOWABLE MOLD FLASH OR PROTRUSION SHALL NOT EXCEED 0.010"
5. TOLERANCE : 0.010"UNLESS OTHERWISE SPECIFIED.
6. AFTER SOLDER DIPPING LEAD THICKNESS WILL BE 0.020"MAX.

Product SPEC. Change Notice

ZEN8105 Specification Revision History

Version	Content	Date
0.0	Original	Dec.2005
	Add register for N-channel open drain share with AN (P1OS in I/O section, SFR section, SFR summary)	05-12-20
	Combining port0-5 to port0-3, port0/1/2 are 8-bit ports, port 3 is 5-bit port (port reg in IO section, SFR, initial value, pin configuration, block diagram, pin description, pin function)	05-12-21
	Change PWE to OCON register	05-12-22
	Add temperature condition of RC; revise buzzer section; rename pwm12&pwm8 as pwm0&1 Delete s-dip/sop package of 32 pin; add T2EX as Schmitt trigger; change frequency 400k – 12M;	05-12-29
	Revise buzzer section; Remove TIE bit in pwm1 register; add EPWM1 bit in register IEN1; add notes of warm-up time when power-on reset; confirm that both level and edge triggered interrupt can wake up power-down mode; power-down current is TBD by design; change RESET function from 2 clock cycles to a high level not less than 10us; change INTX in timer0,1 block diagram to INTmn(mn=00,01); add 2 nops before stop operation Note: the output function of IO when external interrupt occurs is TBD The description of timer1 in mode 3 is still in discussion The flash interface is TBD	06-01-05
	Revise port0-3 /p0-3cr0/p0-3pcr0 initial value	06-1-16
	Add register MPAGE at 0F7h,change register PWMC8,PWMCR8 to 0Fah,0FBh	06-01-18
	add timer0/1/2 compare function. Add register ccpr0/1/2,tcome and related description and correct sfr pwm renamed as pwm0/1/2. pwm0 is 12-bit,pwm1/2 is 8 bit; add pwm1/2 duty reg pwm1/2d and pwm1/2 period reg pwm1/2p; port 25 is shared with pwm2 output. Adc conversion time is shortened to 10us. Revised block diagram and pin description diagram and io function description. Sfr address is revised based on RD8051 cpu core.	06-04-10
	Change the way that how to write Flash	06-04-14
	Rearrange pin location, not necessary pin to pin match MB89F202	06-05-17
	Delete DPL2, DPh2, DPS in SFR; change PWM01/11/21 Dead Time reg; add note that PWM01/11/21 has no output if PWM0/1/2 is disabled.	
	Delete P17/RST open-drain description	
	Add ceramic oscillator type	
	Del C register; Revise the description of how to quit IDLE/POWER-DOWN mode.	
V0.1	Revised based on SH79F32 PDCN; added FLT pin and function.; External interrupt can be triggered on both rising edge and falling edge; PWM interrupt added.	06-10-12
	Rearrange the pin location	06-10-24
	Revised external interrupt 0/1 to external interrupt 0-4 Exchange bit MODF and MSTR in SPI block Re-arrange the location of EXF0/1 in SFR map	06-11-07
	Correct block diagram of timer2 compare function	06-11-07
	Add rules for change dead time in PWM block; Clarify the conditions for response external interrupt	06-11-14
	Rearrange the location of SPCON,SPSTA,SPDAT registers and FOCON, IB_CLK0, IB_CLK1,IB_OFFSET,IB_DATA register; Correct the way to clear TF0/1; Exchange bit MODF and MSTR in SPI block to the original arrangement; Describing FLT has no pull-high.	06-11-17
	Rearrange the location of SPCON, SPSTA, SPDAT registers and FOCON, IB_CLK0, IB_CLK1, IB_OFFSET,IB_DATA register;	06-11-17
	Add 3 reset flags in WDCON register; Add 32-pin SOP package	06-11-27

	Add WDOF1 flag in WDCON Rename WDCON to RSTSTAT	06-12-04
	Add instruction set; Remove compare function in Timer2; Revise compare function diagram of Timer0/1	06-12-11
	RESET High active changed to Low active;	07-01-11
	Revise description of SPI, defining data in SPDAT is invalid when SPI is disabled	07-02-01
	Add 28 pin SOP package;	07-02-15
V0.2	Delete 28 pin package	07-06-29
	Add order information and package information	07-06-29
	Delete SCM block	07-06-29
	Get rid of FCOF description	07-06-29
	AC/DC Electrical Characteristics change	07-06-29
	Uart/SPI Diagram update	07-06-29
	Sysck must be lower than 8MHz when Flash in SSP mode.	07-06-29
	Get rid of external int in level trig mode wakeup IDLE/Power Down function.	07-06-29
	Add warm up time description table	07-06-29
	Get rid of Crystal type for oscillator	07-08-14
	Update AC/DC Electrical Characteristics	07-08-15
	Get rid off wake up power down mode notes(external interrupt level mode)	07-10-15
	Update flash SSP operation temperature	07-10-15
V0.3	Regulator output voltage change from 2.5V to 2.9V.	07-12-15
	Update low operation voltage from 3.0V to 4.0V	07-12-15
	Add PWM dead time register description	07-12-15

Contents