R8032TTEX

8-Bit RISC MCU IP Specification

VERISION:1.1

R D C ***RISC DSP Controller***

RDC Semiconductor Co., Ltd

http:\\www.rdc.com.tw

Tel. 886-3-666-2866

Fax 886-3-563-1498

R8032TTEX 8-BIT MCU IP SPECIFICATION

1. Introduction

- **RISC Architecture**
- **Synchronous Design**
- **Static Design**
- **Synthesizable**
- **Silicon proved (0~66 MHz at 0.5um process)**
- **RTL Code with verilog format**
- **Application**

  -DSC,CF Card, CD ROM Controller, Pattern Recognition, LCD Moniter Controller, USB Device Controller, Scanner Controller, MP3 Controller, Modem Controller, Voice Recognition, …etc.

2.Features

- **Instruction compatible is compatible with generic 8051**
- **256 byte scratchpad RAM interface**
- **Two external interrupts**
- **Memory Addressing Capability**

  -64K Byte external RAM & ROM
- **8-bit I/O port x 4 (P0~P3)**
- **Extra I/O port (P4)**
- **16-bit timer/counter x3**
- **Full duplex UART x2**
- **Dual Data Pointer**
- **Watch Dog Timer x1**
- **Support Power Down and Idle Mode**
- **Power Down waked up by Interrupt**
- **Variable Length MOVX to access slow peripheral RAM**
- **Programmable clock source for timer(1/4,1/12)**
- **Enhance MUL/MOVX/INC Dptr instruction**
- **Code access insert wait state**
- **PWM x 2**
- **MOVC instruction inhibit**
- **Serial EEPROM interface**
- **A5 mode (Debug mode)**

- **Programmable Watch Dog Timer**
- **Extra external interrupt x2**
- **Power Management Mode**
  - **Programmable clock source save power**
  - **Automatic baud rate recovery**
  - **Power down control**
- **Enhanced UART**
  - **Framing error detection**
  - **Automatic address recognition**
  - **RX/TX FIFO**
  - **Clock off**
- **MCS x 1**

3.GENERAL DESCRIPTION

The 8032T is a high-performance 8051 family compatible micro-controller based on RISC architecture & Pipeline design. This IP Specification of interface timing, external Data Memory read / write timing and external Program Memory read timing are different from that of the standard 80C52. But instruction-set is fully compatible with standard 8051 family.

4.FUNCTIONAL DESCRIPTION

Memory

The R8032TTEX manipulates operands in four memory spaces. There are 64K-byte Program Memory space, 64K-byte External Data Memory space, 256-byte Internal Data Memory, and with a 16-bit Program Counter space. The Internal Data Memory address space is further divided into the 256-byte Internal Data RAM and 128-byte Special Function Register address space. The up 128-bytes RAM can reach by indirect addressing. Four Register Banks, 128 addressable bits, and the stack reside in the Internal Data RAM.

I/O ports

The R8032TTE has 8-bit I/O ports. The four ports and one extra port (Port 4)provide 32 I/O lines to interface to the external world. All four ports are both byte and bit addressabl but port 4 is only byte addressable. The R8032TTE uses dedicate bus when external program is running or external memory/device is accessed. Port 1 is used for both I/O and external interrupts.

Interrupts

In the R8032TTEX there are six hardware resources that generate an interrupt request. The starting addresses of the interrupt service program for each interrupt source are like standard 8052. The external interrupt request inputs ($\overline{INT0}$, $\overline{INT1}$) can be programmed for either negative edge or low level-activated operation.

3 Timers / Counters

The R8032TTEX has three 16-bit timers/counters that are same as the timers of the standard 8051 family. The R8032TTEX has two additional watchdog timers for system failure monitor.

Serial I/O ports

The R8032TTEX has 2 programmable, full-duplex serial I/O ports that the function is same as that of 8051 family and dependent on requirement.

Power Management

The R8032TTEX default support IDLE and POWER-DOWN modes of operation. In the IDLE mode, the CPU core is stopped operation while the peripherals continue operating. In the POWER-DOWN mode, all the clocks are stopped. The power-down mode can be waked up by $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt with level trigger. The extra power management can be found on PMR register in SFR.

Dual Data Pointer

The R8032TTEX has 2 data pointers (DTPR, DTPR1). These two data pointers can help users enhance lots of block data memory moving. Using dual data pointers to move block data almost saves half of the time spent by original 8051 codes.

Watch Dog Timers Interrupt / Reset

The R8032TTEX creates one programmable watchdog timers to monitor system failure. That is maximum 2^{26} .

Hardware Multiply

R8032TTEX includes a hardware multiplier to enhance calculating speed. R8032TTEX can finished one multiply instruction at 1 machine cycle.

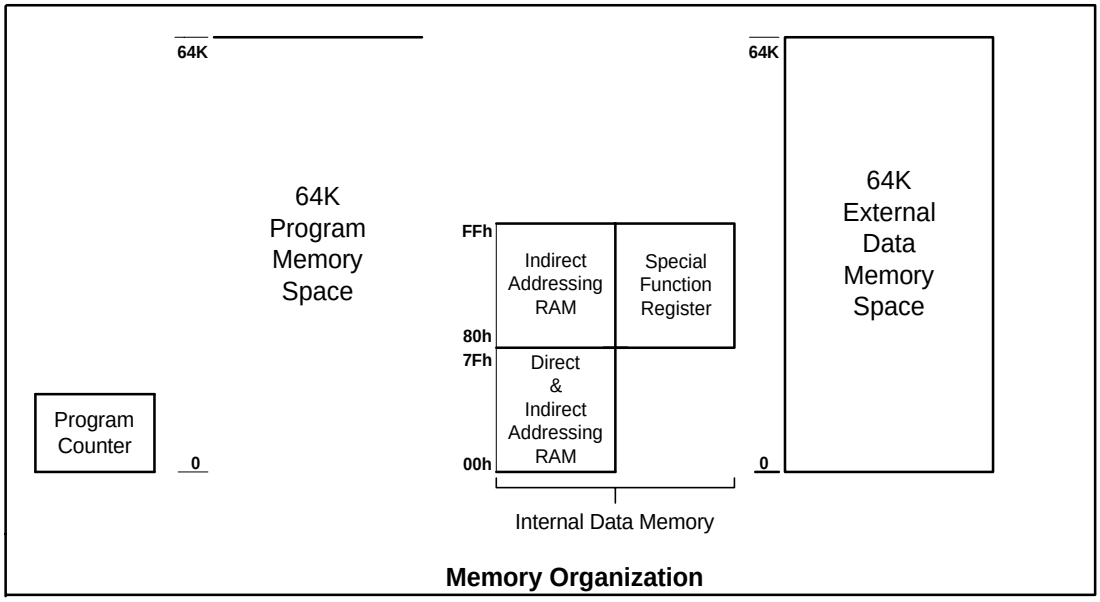
5.MEMORY ORGANIZATION

In the R8032TTEX the memory is organized as three address spaces and the program counter. The memory spaces shown in memory map.

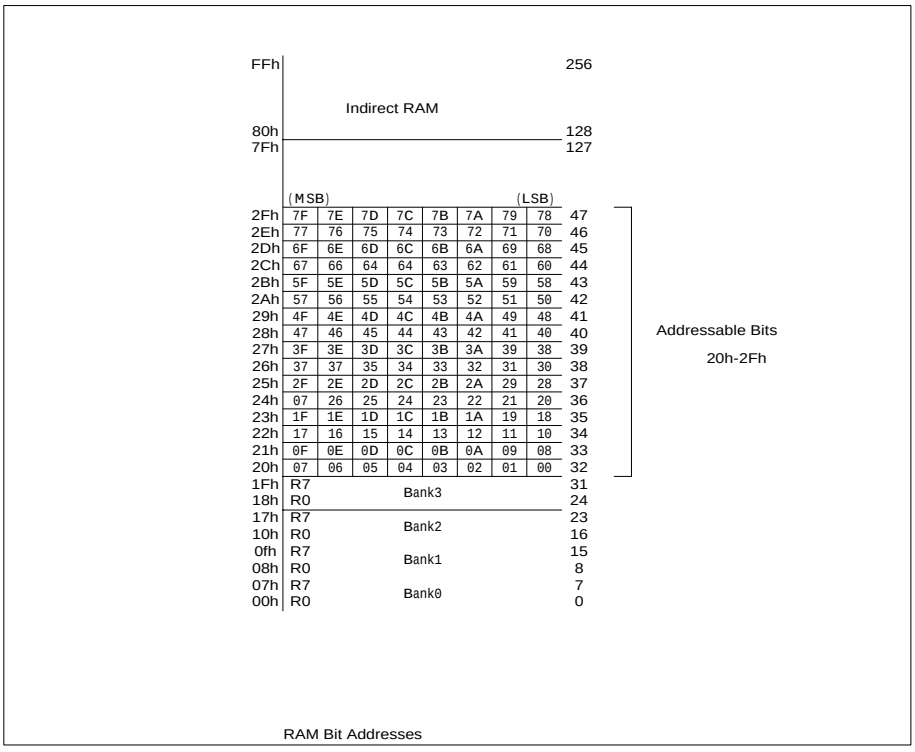
- 16-bit Program Counter
- 64k-byte Program Memory address space
- 64k-byte External Data Memory address space
- 256-byte Internal Data Memory address

The 16-bit Program Counter register provides the R8032TTEX with its 64k addressing capabilities. The program Counter allows the user to execute calls and branches to any location within the Program Memory space. There are no instructions that permit program execution to move from the Program Memory space to any of the data memory spaces.

The R8032TTEX 64k-byte Program Memory address space is located in the dedicate address bus. The 64k-byte External Data Memory address space is automatically accessed when the MOVX instruction is executed.The Internal Data Memory space is subdivided into a 256-byte Internal Data RAM address
Space and a 128-byte Special Function Resister address space as shown in the SFRs Map.
The Internal Data RAM address space is 0 to 255. Four 8-Register Banks occupy locations 0 through 31. The stack can be located anywhere in the Internal Data RAM address space.
In addition, 128 bit locations of the on-chip RAM are accessible through Direct Addressing.



Memory Organization



Special Function Registers Description

The Special Function Register address space is 80h to FFh. All registers except the Program Counter and the four 8-Register Banks reside here. The SFRs are accessed using direct addressing only. All of the SFRs are the compatible with standard 8052 .

F8H	EIP		PCLStack	PCLA5Entry	PCLExEntry			
F0H	B		PCHStack	PCHA5Entry	PCHExEntry		PWM21Cf	PWM11Cf
E8H	EIE		CdInsWtSt	E2RmDvAdr	E2RmAdrH	E2RmAdrL	E2RmData	E2RmCfSt
E0H	ACC	PDCON	ReP2					
D8H	WDTCON		P4EnOn	UxFIFOCf	PWM2Cf	PWM1Cf	PWMData1	PWMData2
D0H	PSW		E2RmInTFg		MCS0	MCS0L		
C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2		
C0H	SCON1	SBUF1			PMR	STATUS		
B8H	IP	SADEN	SADEN1					
B0H	P3							
A8H	IE	SADDR	SADDR1					
A0H	P2				Reserved	Reserved		
98H	SCON	SBUF						
90H	P1	EXIF						
88H	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	P4
80H	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON

EIP Address:F8h

bit 7							bit 0
—	—	—	—	—	—	PX3	PX2

Extent Interrupt Priority Control Register

Bit 7~bit2: reserved

PX3 : External interrupt 3 priority bit

Set this bit will let External Interrupt 3 at high priority.

PX20 : External interrupt 2 priority bit

Set this bit will let External Interrupt 2 at high priority.

B Address:F0h

bit 7							bit 0
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0

B Register. The B Register is used as both a source and destination in multiply and divide operations.

EIE Address:E8h

bit 7							bit 0
—	—	—	—	—	—	EX3	EX2

Extent Interrupt Enable Register.
EX3 : External Interrupt 3 Enable
Set this bit enables external interrupt 3.
EX2 : External Interrupt 2 Enable
Set this bit enables external interrupt 2.
Bit 7~bit 2:reserved

P4 Address:8FH

bit 7							bit 0
P4_7	P4_6	P4_5	P4_4	P4_3	P4_2	P4_1	P4_0

Port4: Extra I/O
P4 is the SFR that contains data to be driven from the port 4 pins.
Read_modify_write instructions that read port 4 read this register, other instructions that read port1 read the port 4 pins.

EXIF Address:91H

bit 7							bit 0
—	—	IE3	IE2	—	—	—	—

IE3: Extra Interrupt 3 flag. Set by hardware when external Interrupt 3 Negative edge is detected or set by software, cleared when Int processed.
IE2: Extra Interrupt 2 flag. Set by hardware when external Interrupt 2 Positive edge is detected or set by software. Cleared when Int processed.
Bit 7,6,3,2,1,0: reserved

ACC Address:E0h

bit 7							bit 0
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0

Accumulator. The instruction use the accumulator as both source and destination for calculations and moves.

PSW Address:D0h

bit 7							bit 0
CY	AC	F0	RS1	RS0	OV	F1	PARITY

Program Status Word.

CY: Carry Flag

CY is set if the operation result in a carry out of (during addition) or a borrow into (during subtraction) the high-order bit of the result; otherwise CY is cleared.

AC: Auxiliary-Carry Flag

AC is set if the operation results in a carry out of the low-order 4 bits of the result (during addition) or a borrow form the high-order bits into the low-order 4 bits (during subtraction); otherwise AC is cleared.

F0: User Flag 0

General-purpose flag.

RS1,RS0: Register Bank Select Bits 1 and 0

These bits select the memory locations that comprise the active bank of the register file.

RS1	RS0	Bank	Address
0	0	0	00-07h
0	1	1	08h-0Fh
1	0	2	10h-17h
1	1	3	18h-1Fh

OV: Overflow Flag

This bit is set if an addition or signed variables results in an overflow error (i.e., if the magnitude of the sum or difference is too great for the seven LSBs in 2's-complement representation). The overflow flag is also set if multiplication product overflows one byte or if a division by zero is attempted.

UD: User-definable Flag

This general-purpose flag is available to the user.

P: Parity Flag

This bit indicates the parity of the accumulator. It is set if an odd number of bits in the accumulator are set Otherwise, it is cleared. Not all instructions update the parity bit. The parity bit is set or cleared by instructions that change the contents to the accumulator.

T2CON Address:C8h

bit 7							bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Timer 2 Control Register.

TF2: Timer2 Overflow Flag

Set by time 2 overflow. Must be cleared by software. TF2 is not set if RCLK=1or TCLK=1.

EXF2: Timer 2 External Flag

If EXEN2=1, capture or reload caused by a negative transition on T2EX sets EFX2. EFX2 dose not cause an interrupt in up/down counter mode (DCEN=1)

RCLK: Receive Clock Bit

Selects timer 2 overflow pulses (RCLK=1) or timer 1overfiow pulses (RCLK=0) as the baud rate generator for port modes 1 and 3.

TCLK: Transmit Clock Bit

Select timer 2 overflow pulses (TCLK=1) or timer 1 overflow pulses (TCLK=0) as the band rate generator for serial port modes 1 and 3.

EXEN2: Timer 2 External Enable Bit

Setting EXEN2 causes a capture or reload to occur as result of a negative transition on T2EX unless timer 2 is being used as the baud rate generator for the serial port. Clearing EXEN2 causes timer 2 to ignore events at T2EX.

TR2: Timer 2 Run Control Bit

Setting this bit starts the timer.

C/$\overline{T2}$: Timer 2 Counter/Timer Select

C/$\overline{T2}$ = 0 selects timer operation: timer 2 counts the divided-down system clock.

C/$\overline{T2}$ = 1 selects counter operation: timer 2 counts negative transitions on external pin T2.

CP/$\overline{RL2}$: Capture/Reload Bit

When set, captures occur on negative transitions at T2EX if EXEN2=1.When cleared, auto-reloads occur on timer 2 overflows or negative transitions at T2EX if EXEN2=1. The CP/$\overline{RL2}$ bit is ignored and timer 2 forced to auto-reload on timer 2 overflow, if RCLK =1 or TCLK = 1.

T2MOD Address:C9h

bit 7							bit 0
–	–	–	–	–	–	T2OE	DCEN

Timer 2 Mode Control Register.

Bit7-Bit2 : Reserved

T2OE: Timer 2 Output Enable Bit

In the timer 2 clock-out mode, connects the programmable clock output to external pin T2.

DCEN: Down Count Enable Bit

Configures timer 2 as an up/down counter.

RCAP2L Address:CAh

bit 7							bit 0
RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0

Timer2 Capture LSB

Low byte of the timer2 reload/recapture register. This register stores 8-bit values to be loaded into or captured from the timer register TL2 in timer 2

RCAP2H Address:CBh

bit 7							bit 0
RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0

Timer2 Capture MSB

High byte of the timer2 reload/recapture register. This register stores 8-bit values to be loaded into or captured from the timer register TH2 in timer2

TL2 Address:CCh

bit 7							bit 0
TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0

Timer 2 LSB

Low byte of the timer 2 timer register

TH2 Address:CDh

bit 7							bit 0
TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0

Timer 2 MSB

High byte of the timer 2 timer register

IP Address:B8h

bit 7							bit 0
—	PS1	PT2	PS	PT1	PX1	PT0	PX0

Interrupt Priority Control Register

IP.7 : Reserved bits

PS1: Serial port 1 priority control bit. 1: high priority interrupt

PT2: Timer 2 interrupt priority control bit. 1:

PS0: Serial port0 priority control bit. 1: high priority interrupt

PT1: Timer 1 interrupt priority control bit.

PX1: External interrupt 1 priority control bit. 1: high priority interrupt

PT0: Timer 0 interrupt priority control bit.

PX0: External interrupt 0 priority control bit.1

P3 Address:B0h

bit 7							bit 0
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Port 3. P3 is the SFR that contains data to be driven out from the port 3 pins. Read-modify-write instructions that read port 3 read this register. Other instructions that

read port 3 read the port 3 pins.

IE Address:A8h

bit 7							bit 0
EA	ES1	ET2	ES	ET1	EX1	ET0	EX0

Interrupt Enable Register.

EA: Global Interrupt Enable

Setting this bit enables all interrupts that are individually enabled by bits 0-6. Clearing this bit disables all interrupts.

ES1: Serial port 1 Interrupt Enable

Setting this bit enables the serial port 1 interrupt.

ET2: Timer 2 Overflow Interrupt Enable

Setting this bit enables the timer 2 overflow interrupt.

ES0: Serial port0 Interrupt Enable

Setting this bit enables the serial port0 interrupt.

ET1: Timer 1 Overflow Interrupt Enable

Setting this bit enables the timer 1 overflow interrupt.

EX1: External Interrupt 1 Enable

Setting this bit enables external interrupt 1.

ET0: Timer 0 Overflow Interrupt Enable

Setting this bit enables the timer 0 overflow interrupt.

EX0: External Interrupt 0 Enable

Setting this bit enables external interrupt 0.

P2 Address:A0h

bit 7							bit 0
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0

Port 2. P2 is the SFR that contains data to be driven out from the port 2 pins. Read-

modify-write instructions that read port 2 read this register. Other instructions that read port 2 read the port 2 pins.

SBUF Address:99h

bit 7							bit 0
SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0

Serial Data Buffer. Writing to SBUF loads the transmit buffer to the serial I/O port. Reading SBUF reads the receive buffer of the serial port.

SBUF1 Address:C1h

bit 7							bit 0
SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0

Serial Data Buffer. Writing to SBUF1 loads the transmit buffer to the serial I/O port1. Reading SBUF reads the receive buffer of the serial port1.

P1 Address:90h

bit 7							bit 0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Port 1. P1 is the SFR that contains data to be driven out from the port 1 pins. Read-modify-write instructions that read port 1 read this register. Other instructions that read port 1 read the port 1 pins.

TCON Address:88h

bit 7							bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Timer/Counter Control Register..

TF1: Timer 1 Overflow Flag.

Set by hardware when the timer 1 register overflows. Cleared by hardware when the processor vectors to the interrupt routine.

TR1:Timer 1 Run Control Bit.

Set/cleared by software to turn timer 1 on/off.

TF0: Timer 0 Overflow Flag.

Set by hard ware when the timer 0 register overflows. Cleared by hardware when the processor vectors to the interrupt routine.

TR0: Timer 0 Run Control Bit.

Set/cleared by software to turn timer 1 on/off.

IE1: Interrupt 1 Edge Detect

Set by hard ware when an external interrupt is detected on the $\overline{INT1}$ pin edge- or level-triggered (see IT1). Cleared when interrupt is processed if edge- triggered.

IT1: Interrupt 1 Type Select Bit.

Set this bit select edge-triggered (high-to-low) for external interrupt 1. Clear this bit to select level-triggered (active low).

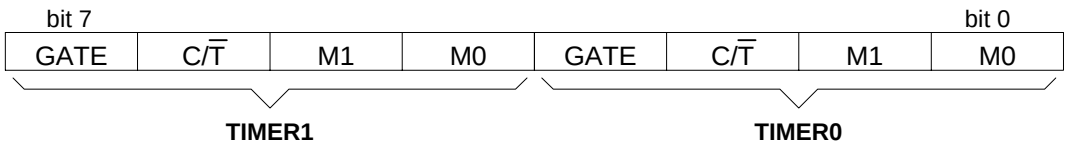
IE0: Interrupt 0 Edge Detect

Set by hardware when an external interrupt is detected on the $\overline{INT0}$ pin edge- or level-triggered (see IT0). Cleared when interrupt is processed if edge- triggered.

IT0: Interrupt 0 Type Select Bit.

Set this bit to select edge-triggered (high-to-low) for external interrupt 0. Clear this bit to select level-triggered (active low).

TMOD Address:89h



Timer/Counter Mode Control Register.

GATE (TMOD.7): Timer 1 Gate. Control: enables/disables the ability of timer 1 to increment

0: Timer 1 will clock when TR1=1, regardless of the state of $\overline{InT1}$

1: Timer 1 will clock only when TR1=1 and $\overline{InT1}$=1

C/$\overline{T}$ (TMOD.6): Timer 1 Counter/Timer Select

C/$\overline{T}$ = 0 selects timer operation: timer 1 counts the divided-down system clock.

C/$\overline{T}$ = 1 selects counter operation: timer 1 counts negative transitions on external pin T1.

M1 (TMOD.5), M0 (TMOD.4) :Timer 1 mode select

M1	M0	MODE	
0	0	0	: 8-bit timer/counter (TH1) with 5-bit prescalar (TL1)
0	1	1	: 16-bit timer/counter
1	0	2	: 8-bit auto-reload timer/counter (TL1). Reload from TH1 at overflow.
1	1	3	: timer 1 halted. Retains count.

GATE (TMOD.3) : Timer 0 Gate.

When GATE = 0, run control bit TR0 gates the input signal to the timer register. When GATE = 1 and TR0 = 1,external signal $\overline{INT0}$ gates the timer input.

C/$\overline{T}$ (TMOD.2): Timer 0 Counter/Timer Select

C/$\overline{T}$ = 0 selects timer operation: timer 0 counts the divided-down system clock.

C/$\overline{T}$ = 1 selects counter operation: timer 0 counts negative transitions on external pin T0.

M1 (TMOD.1), M0 (TMOD.0) : Timer 0 Mode Select.

M1	M0	MODE	
0	0	0	: 8-bit timer/counter (TH0) with 5-bit prescalar (TL0)
0	1	1	: 16-bit timer/counter
1	0	2	: 8-bit auto-reload timer/counter (TL0). Reload from TH0 at overflow.
1	1	3	: TL0 is an 8-bit timer/counter. TH0 is an 8 bit timer using timer 1's TR1 and TF1 bits.

TL0 Address:8Ah

bit 7							bit 0
TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0

Low Byte of the timer 0 timer register.

TL1 Address:8Bh

bit 7							bit 0
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

Time 1 LSB

This register contains the lease significant byte Timer1

TH0 Address:8Ch

bit 7							bit 0
TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

High Byte of the timer 0 timer register.

TH1 Address:8Dh

bit 7							bit 0
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

High Byte of the timer 1 timer register.

P0 Address:80h

bit 7							bit 0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

Port 0. P0 is the SFR that contains data to be driven out from the port 0 pins. Read-modify-write instructions that read port 0 read this register. Other instructions that read port 1 read the port 0 pins.

SP Address:81h

bit 7							bit 0
SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0

Stack Pointer. The 8-bit SP contains the address at which the last byte was push onto the stack. This is also the address of the next byte that will be stopped. The SP is incremented before very PUSH operation.. SP can be read or written to under software control.

DPL **Address:82h**

bit 7							**bit 0**
DPL.7	**DPL.6**	**DPL.5**	**DPL.4**	**DPL.3**	**DPL.2**	**DPL.1**	**DPL.0**

Data Pointer Low. DPL is the low byte of the 16-bit data pointer, DPTR.

DPH Address:83h

bit 7							bit 0
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0

Data Pointer High. DPH is the high byte of the 16-bit data pointer, DPTR.

DPL1 Address:84h

bit 7							bit 0
DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0

Data Pointer 1 Low. DPL1 is the low byte of the 16-bit data pointer, DPTR.

DPH1 **Address:85h**

bit 7							**bit 0**
DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0

Data Pointer 1 High. DPH1 is the high byte of the 16-bit data pointer, DPTR.

PCON Address:87h

bit 7							bit 0
SMOD1	SMOD0	—	—	GF1	GF0	PD	IDL

Power Control Register.

SMOD1: Double Baud Rate Bit.

When set, Doubles the baud rate when timer 1 is used and mode 1, 2, or 3 is selected in the SCON register.

SMOD0: This bit selects function of the SCON0.7 and SCON1 bits.

0= SCON0.7 and SCON1.7 control the SMO function defined for the SCON0 and SCON1 register

1=SCON0.7 and SCON1.7 are converted to the Framing Error(FE) flag for the respective serial port

PCON.5-PCON.4: The Reserved Bits.

GF1,GF0: for general purpose register. It will be (1,1) when it came from Idle mode

PD: Power- Down Mode Bit.

When set, activates power-down mode. Clear by hardware when an interrupt or reset occurs.

IDL: Idle Mode Bit.

When set, activates Idle mode. Clear by hardware when an interrupt or reset occurs.

PDCON Address:E1H

bit 7							bit 0
ICEINTFg	—	P3EnOn	P2EnOn	P1EnOn	P0EnOn	JWP	PDC

Power Down Controller register

PDCON register

address E1H

ICEINTFg, --, P3EnOn, P2EnOn, P1EnOn, P0EnOn, JWP, PDC

PDC : for power down control

0 : orginal (pull-high)

1 : SFR value (default)

JWP : Just Wake up

0 : Execute Int after wake up (default)

1 : Don't execute Int after wake up

P0EnOn : 1 / 0 [default] => always enable / depend on SFR

P1EnOn : 1 / 0 [default] => always enable / depend on SFR

P2EnOn : 1 / 0 [default] => always enable / depend on SFR

P3EnOn : 1 / 0 [default] => always enable / depend on SFR

ICEINTFg: for firmware reference [active low]

WDTCON Address:0D8H

bit 7							bit 0
SMOD_1	—	—	—	—	WDTRSTFg	WDTEN	WDTRST

Watch Dog Timer Controller register

Address 0D8H

SMOD_1,--,--,--,--, --,WDTEN,WDTRSTFg,WDTRST

SMOD_1 : Serial Modification, This bit controls the doubling of the serial port 1 buad rate in modes 1,2,3

WDTEN : Watchdog Timer Reset Enable

WDTRST : Watchdog Timer Reset

WDTRSTFg : The Reset Flag of Watchdog timer

SCON1 Address:0C0H

bit 7							bit 0
SM0_1/FE_1	SM1_1	Sm2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1

SCON_1 register

Address 0C0H

SM0_1/FE_1,SM1_1,SM2_1,REN_1,TB8_1,RB8_1,TI_1,RI_1

FE_1 Framing Error Bit.

This bit is set by the receiver when an invalid stop bit is detected.

The FE bit is not cleared by valid frames but should be cleared by software.

The "SMOD0" bit must be set to enable access to the FE bit.

SM0_1:1 Serial port mode control Set/cleared by software

SM2_1 Set by software to disable reception of frames for which bit8 zero

REN_1 Receiver ENable bit.Set/cleared by software to enable/ disable serial data reception

TB8_1 Transmit Bit8 Set/Cleared by hardware to determine state of ninth data bit transmitd in 9-bit UART mode

RB8_1 Receive Bit8 Set/cleared by hardware to indicate state of ninth data bit

received

TI_1 Transmit Interrupt flag.Set by hardware when byte transmitted. Cleared by software after serving.

RI_1 Received Interrupt flag.Set by hardware when byte received. Cleared by software after serving.

SM1_1:SM2_1 SELECT

00 Shift reg. I/O expansion

01 8 bit UART,variable data rate

10 9 bit UART,fixed data rate

11 9 bit UART,variable data rate

SCON Address:098H

bit 7							bit 0
SM0/FE	SM1_0	SM2_0	REN	TB8	RB8	TI	RI

SCON register

Address 098H

SM0/FE,SM1,SM2,REN,TB8,RB8,TI,RI

FE Framing Error Bit.

This bit is set by the receiver when an invalid stop bit is detected.

The FE bit is not cleared by valid frames but should be cleared by software.

The "SMOD0" bit must be set to enable access to the FE bit.

SM0:1 Serial port mode control Set/cleared by software

SM2 Set by software to disable reception of frames for which bit8 zero

REN Receiver ENable bit.Set/cleared by software to enable/ disable serial data reception

TB8 Transmit Bit8 Set/Cleared by hardware to determine state of ninth data bit transmitd in 9-bit UART mode

RB8 Receive Bit8 Set/cleared by hardware to indicate state of ninth data bit received

TI Transmit Interrupt flag.Set by hardware when byte transmitted. Cleared by software after serving.

RI Received Interrupt flag.Set by hardware when byte received. Cleared by software after serving.

SM1_0:SM2_0 SELECT

00 Shift reg. I/O expansion

01 8 bit UART,variable data rate

10 9 bit UART,fixed data rate

11 9 bit UART,variable data rate

SADEN0/SADEN1

SADEN0 Address: B9H

bit 7 bit 0

SADEN0.7	SADEN0.6	SADEN0.5	SADEN0.4	SADEN0.3	SADEN0.2	SADEN0.1	SADEN0.0

SADEN1 Address: BAH

bit 7 bit 0

SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0

SADEN0 / SADEN1 register
Slave Address Mask
Enable Register 0/1
address 0B9H / 0BAH
Slave Address Enable
For automatic address recognition
It is the mask of Slave Address
0 : don't care
1 : dependent on Slave Address
default : 8‘h0

When a bit in this register set, the corresponding in SADDR register will be exactly compared with the incoming serial port data, to determine if a receiver interrupt should be generated , when a bit in this register is cleared, the corresponding bit in the SADDR0 register becomes a don’t care.

SADDR0/SADDR1

SADDR0 Address: A9H

bit 7 bit 0

SADDR0.7	SADDR0.6	SADDR0.5	SADDR0.4	SADDR0.3	SADDR0.2	SADDR0.1	SADDR0.0

SADDR1 Address: AAH

bit 7							bit 0
SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0

SADDR0 / SADDR1 register

Slave Address Register

address 0A9H / 0AAH

Slave Address

default : 8‘h0

This register is programmed with the given or broadcast address assigned to serial port 0/1.

PMR Address:C4H

bit 7							bit 0
CD1	CD0	SWB	UARTOFF	—	—	—	—

Power Manager Register

address C4H

CD1, CD0 , SWB, UARTOFF, --, --, --, --

CD1, CD0 : (0, 0) => Reserve

: (0, 1) => Normal / Default

: (1, 0) => 1/16

: (1, 1) => 1/256

SWB : Switchback Enable

This bit allows an enabled external interrupt or serial port activity to force the Clock Divide Control bits to the divide by 4 state (01). Upon internal acknowledgement of an external interrupt, the device will switch modes at the start of the jump to the interrupt service routine. Note that this means that an external interrupt must actually be recognized (i.e. be enabled and not masked by higher priority interrupts) for the switchback to occur. For serial port reception, the switch occurs at the start of the struction following the falling edge of the start bit.

UARTOFF : 1 => disable the clock input of uart

0 => enable the clock input of uart

Status Address:C5h

bit 7							bit 0
—	HIP	LIP	—	SPTA1	SPRA1	SPTA0	SPRA0

Status Register

address : C5h

--, HIP, LIP, --, SPTA1, SPRA1, SPTA0, SPRA0

HIP : High Pirority Interrupt Status

LIP : Low Pirority Interrupt Status

SPTA1 : Serial Port1 Transmit Activity Monitor

SPRA1 : Serial Port1 Receive Activity Monitor

SPTA0 : Serial Port0 Transmit Activity Monitor

SPRA0 : Serial Port0 Receive Activity Monitor

CKCON Address:08Eh

bit 7							bit 0
WDT1	WDT0	T2M	T1M	T0M	MD2	MD1	MD0

Clock Control register

address : 08Eh

WDT1, WDT0, T2M, T1M, T0M, MD2, MD1, MD0

WDT1, WDT0 : WDT time-out counter select

0,0 - 17 bit counter / 0,1 - 20 bit counter

1,0 - 23 bit counter / 1,1 - 26 bit counter

T2M : Timer2 clock = system clock /4 or 12 (1/0)

T1M : Timer1 clock = system clock /4 or 12 (1/0)

T0M : Timer0 clock = system clock /4 or 12 (1/0)

MD2, MD1, MD0 : Insert Wait-state for Movx

(0, 0, 0) : No Wait-State / (0, 0, 1) : 4T

(0, 1, 0) : 8T / (0, 1, 1) : 12T

(1, 0, 0) : 16T / (1, 0, 1) : 20T

(1, 1, 0) : 24T / (1, 1, 1) : 28T

UxFIFOCf Address:DBH

bit 7							bit 0
—	T2D2	T1D2	T0D2	TX1FIFOEn	U1FIFOFn	TX0FIFOEn	U0FIFOEn

Uart FIFO Configure register

address DBH

--, T2D2, T1D2, T0D2, TX1FIFOEn,U1FIFOEn , TX0FIFOEn, U0FIFOEn

U0FIFOEn : Uart0 Rx FIFO Enable / Disable_
1 : Enable => Interrupt or Flag / 4 Bytes
0 : Disable (default)=> Interrupt or Flag / 1 Bytes
TX0FIFOEn : Uart0 Tx FIFO Enable / Disable_
1 : Enable => Interrupt or Flag / 4 Bytes
0 : Disable (default)=> Interrupt or Flag / 1 Bytes
U1FIFOEn : Uart1 Rx FIFO Enable / Disable_
1 : Enable => Interrupt or Flag / 4 Bytes
0 : Disable (default)=> Interrupt or Flag / 1 Bytes
TX1FIFOEn : Uart1 Tx FIFO Enable / Disable_
1 : Enable => Interrupt or Flag / 4 Bytes
0 : Disable (default)=> Interrupt or Flag / 1 Bytes
T0D2 : Timer0 clock = system_clock (1/0 => On / Off [default])
T1D2 : Timer1 clock = system_clock (1/0 => On / Off [default])
T2D2 : Timer2 clock = system_clock (1/0 => On / Off [default])

P4EnOn Address:DAH

bit 7							bit 0
P47EnOn	P46EnOn	P45EnOn	P44EnOn	P43EnOn	P42EnOn	P41EnOn	P40EnOn

P4EnOn register
address DAH
P47EnOn, P46EnOn, P45EnOn, P44EnOn,
P43EnOn, P42EnOn, P41EnOn, P40EnOn
P4#EnOn : 1 / 0 [default] => always enable / depend on SFR # : 7 ~ 0

E2RmDvAdr Address:EBH

bit 7							bit 0
1	0	1	0	A2	A1	A0	RD/WR_

E2Rom Device Adress Register
address EDH
1 , 0 , 1 , 0 , A2, A1, A0, RD/WR_
(A2, A1, A0) = (0, 0, 0) Default : First Device
RD/WR_ : Active High / Active Low (default)

E2RmAdrH/E2RmAdrL/E2RmData

bit 7 bit 0

E2RmAdrH7	E2RmAdrH6	E2RmAdrH5	E2RmAdrH4	E2RmAdrH3	E2RmAdrH2	E2RmAdrH1	E2RmAdrH0

bit 7 bit 0

E2RmAdrL7	E2RmAdrL6	E2RmAdrL5	E2RmAdrL4	E2RmAdrL3	E2RmAdrL2	E2RmAdrL1	E2RmAdrL0

bit 7 bit 0

E2RmData7	E2RmData6	E2RmData5	E2RmData4	E2RmData3	E2RmData2	E2RmData1	E2RmData0

E2Rom High Byte Address Register

address ECH

E2Rom Low Byte Address Register

address EDH

E2Rom Data (RD/WR_) Register

address EEH

E2RmCfSt Address:EFH

bit 7 bit 0

Div1	Div0	E2RomOn	E2RomGo	E2RomIntEn	E2RomIntP	E2RomOnst	E2RomGoSt

E2Rom Configure/Status Register

address EFH

Div1, Div0, E2RomOn, E2RomGo, E2RomIntEn, E2RomIntP, E2RomOnSt, E2RomGoSt

(Div1, Div0) = (0, 0) => scl = system_clk / 256 (default)

(Div1, Div0) = (0, 1) => scl = system_clk / 128

(Div1, Div0) = (1, 0) => scl = system_clk / 64

(Div1, Div0) = (1, 1) => scl = system_clk / 32

E2RomOn : Enable the Serial EEPROM interface (default Off)

E2RomGo : Star to RD/WR procedure (default off) #Set by Software / Clear by Hardware

E2RomIntEn : E2ROM interrupt enable signal : enable / diable (1/0 [default])

E2RomIntP : The priority signal of E2ROM interrupt signal : High / Low (1/0)

E2RomOnSt : The status of enable interface #It is valid when E2RomOn is activ

#Please reference D2H E2ROMOnINTFg

E2RomGoSt : The status of previous RD/WR procedure #It is valid when

E2romOn & ~E2RomGo

E2RmINTFg Address:D2H

bit 7							bit 0
E2ROMINTFg	ICEINTON	E2ROMOnINTFg	—	—	—	—	—

E2RmINTFg register

address D2H

E2ROMINTFg, ICEINTON, E2ROMOnINTFg, --, --, --, --, --

E2ROMINTFg : The Interrupt Flag of E2ROM # Set by hardware / Clear by software

ICEINTON : for firmware reference [active low / Read Only

E2ROMOnINTFg : The interrupt Flag of E2ROMOn Status # It can be the flag of E2ROMOn status, replacing EFH E2ROMOnSt

ReP2

bit 7							bit 0
ReP2.7	ReP2.6	ReP2.5	ReP2.4	ReP2.3	ReP2.2	ReP2.1	ReP2.0

ReP2 register

address E2H

It will replace P2 [HighByte Address] by ReP2 when the instructions , MOVX @Ri, A / MOVX A, @Ri, have been executed.

PWM1CF/PWM11CF/PWMData1

bit 7							bit 0
PWM1CF7	PWM1CF6	PWM1CF5	PWM1CF4	PWM1CF3	PWM1CF2	PWM1CF1	PWM1CF0

bit 7							bit 0
PWM11CF7	PWM11CF6	PWM11CF5	PWM11CF4	PWM11CF3	PWM11CF2	PWM11CF1	PWM11CF0

bit 7							bit 0
PWMData7	PWMData6	PWMData5	PWMData4	PWMData3	PWMData2	PWMData1	PWMData0

PWM1 Configure

address DDH

M: 0 ~ 255

PWM11 Configure

address F7H

N : 0 ~ 255

PWMData1

address DEH

W: 0 ~ N

PWM_Frequency = {System_clock} / {(M+1)*(N+1)}

PWM_Width => (H, L) = (W+1, N-W)

CdInsWtSt Address:EAh

bit 7							bit 0
IMC	—	PWM1On	PWM2On	CDW3	CDW2	CDW1	CDW0

Code Insert Wait-State Register

address EAh

IMC, --, PWM1On, PWM2On, CDW3, CDW2, CDW1, CDW0

(CDW3, CDW2, CDW1, CDW0) = (0, 0, 0, 0) => Reserve

(CDW3, CDW2, CDW1, CDW0) = (0, 0, 0, 1) => InsCycle = 0

(CDW3, CDW2, CDW1, CDW0) = (0, 0, 1, 0) => InsCycle = 1

(CDW3, CDW2, CDW1, CDW0) = (0, 0, 1, 1) => InsCycle = 2

(CDW3, CDW2, CDW1, CDW0) = (0, 1, 0, 0) => InsCycle = 3

(CDW3, CDW2, CDW1, CDW0) = (0, 1, 0, 1) => InsCycle = 4

(CDW3, CDW2, CDW1, CDW0) = (0, 1, 1, 0) => InsCycle = 5

(CDW3, CDW2, CDW1, CDW0) = (0, 1, 1, 1) => InsCycle = 6

(CDW3, CDW2, CDW1, CDW0) = (1, 0, 0, 0) => InsCycle = 7

(CDW3, CDW2, CDW1, CDW0) = (1, 0, 0, 1) => InsCycle = 8

(CDW3, CDW2, CDW1, CDW0) = (1, 0, 1, 0) => InsCycle = 9

(CDW3, CDW2, CDW1, CDW0) = (1, 0, 1, 1) => InsCycle = 10

(CDW3, CDW2, CDW1, CDW0) = (1, 1, 0, 0) => InsCycle = 11

(CDW3, CDW2, CDW1, CDW0) = (1, 1, 0, 1) => InsCycle = 12

(CDW3, CDW2, CDW1, CDW0) = (1, 1, 1, 0) => InsCycle = 13

(CDW3, CDW2, CDW1, CDW0) = (1, 1, 1, 1) => InsCycle = 14

If the Chip configure is no internal rom => default InsCycle = 7 else the default => InsCycle = 0

PWM2On : Channel 1 ON / OFF (default = 0 / OFF)

PWM1On : Channel 1 ON / OFF (default = 0 / OFF)

IMC: MOVC protect bit.

MCS0H Address:D5H

bit 7							bit 0
—	—	MCS0En	MCS04	MCS03	MCS02	MCS01	MCS00

MCS0H Configure

address D5H

--, --, MCS0En, MCS04, MCS03, MCS02, MCS01, MCS00

MCS0En : Enable MCS0

1 & (MCS0L <= MCS0H) : Enable

0 : Disable (default)

(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 1, 1, 1) => FFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 1, 1, 0) => F7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 1, 0, 1) => EFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 1, 0, 0) => E7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 0, 1, 1) => DFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 0, 1, 0) => D7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 0, 0, 1) => CFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 1, 0, 0, 0) => C7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 1, 1, 1) => BFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 1, 1, 0) => B7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 1, 0, 1) => AFFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 1, 0, 0) => A7FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 0, 1, 1) => 9FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 0, 1, 0) => 97FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 0, 0, 1) => 8FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (1, 0, 0, 0, 0) => 87FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 1, 1, 1) => 7FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 1, 1, 0) => 77FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 1, 0, 1) => 6FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 1, 0, 0) => 67FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 0, 1, 1) => 5FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 0, 1, 0) => 57FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 0, 0, 1) => 4FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 1, 0, 0, 0)=>47FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 1, 1, 1)=>3FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 1, 1, 0)=>37FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 1, 0, 1)=>2FFF

(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 1, 0, 0)=>27FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 0, 1, 1)=>1FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 0, 1, 0)=>17FF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 0, 0, 1)=>0FFF
(MCS04, MCS03, MCS02, MCS01, MCS00) = (0, 0, 0, 0, 0)=>07FF

MCS0L Address:D5H

bit 7							bit 0
—	—	—	MCS04L	MCS03L	MCS02L	MCS01L	MCS00L

MCS0H Configure
address D5H
--, --, --, MCS04L, MCS03L, MCS02L, MCS01L, MCS00L
MCS0En : Enable MCS0
1 & (MCS0L <= MCS0H) : Enable
0 : Disable (default)
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 1, 1, 1) => F800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 1, 1, 0) => F000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 1, 0, 1) => E800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 1, 0, 0) => E000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 0, 1, 1) => D800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 0, 1, 0) => D000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 0, 0, 1) => C800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 1, 0, 0, 0) => C000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 1, 1, 1) => B800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 1, 1, 0) => B000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 1, 0, 1) => A800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 1, 0, 0) => A000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 0, 1, 1) => 9800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 0, 1, 0) => 9000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 0, 0, 1) => 8800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (1, 0, 0, 0, 0) => 8000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 1, 1, 1) => 7800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 1, 1, 0) => 7000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 1, 0, 1) => 6800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 1, 0, 0) => 6000

(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 0, 1, 1) => 5800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 0, 1, 0) => 5000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 0, 0, 1) => 4800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 1, 0, 0, 0)=>4000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 1, 1, 1)=>3800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 1, 1, 0)=>3000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 1, 0, 1)=>2800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 1, 0, 0)=>2000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 0, 1, 1)=>1800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 0, 1, 0)=>1000
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 0, 0, 1)=>0800
(MCS04L, MCS03L, MCS02L, MCS01L, MCS00L) = (0, 0, 0, 0, 0)=>0000

DPS Address:86H

bit 7							bit 0
—	—	—	—	—	—	—	Dps

Data Point Select

address 86H

--, --, --, --, --, --, --, Dps

-- : Reserve

Dps : 0 => Dph, Dpl will be selected

1 => Dph_1, Dpl_1 wiil be selected

PCLStack Address:FAH

bit 7							bit 0
7	6	5	4	3	2	1	0

address FAH

HardWare Stack for Program counter Low-Byte Register

PCHStack Address:F2H

bit 7							bit 0
7	6	5	4	3	2	1	0

address F2H

HardWare Stack for Program counter High-Byte Register

PCLA5Entry Address:FBH

bit 7							bit 0
7	6	5	4	3	2	1	0

address FBH

The Low-Byte of PC for A5 Entry

PCHA5Entry Address:F3H

bit 7							bit 0
7	6	5	4	3	2	1	0

address F3H

The High-Byte of PC for A5 Entry

PCLExEntry Address:FCH

bit 7							bit 0
7	6	5	4	3	2	1	0

address FCH

The Low-Byte of PC for External INT(ICEMD_) Entry

PCHExEntry Address:F4H

bit 7							bit 0
7	6	5	4	3	2	1	0

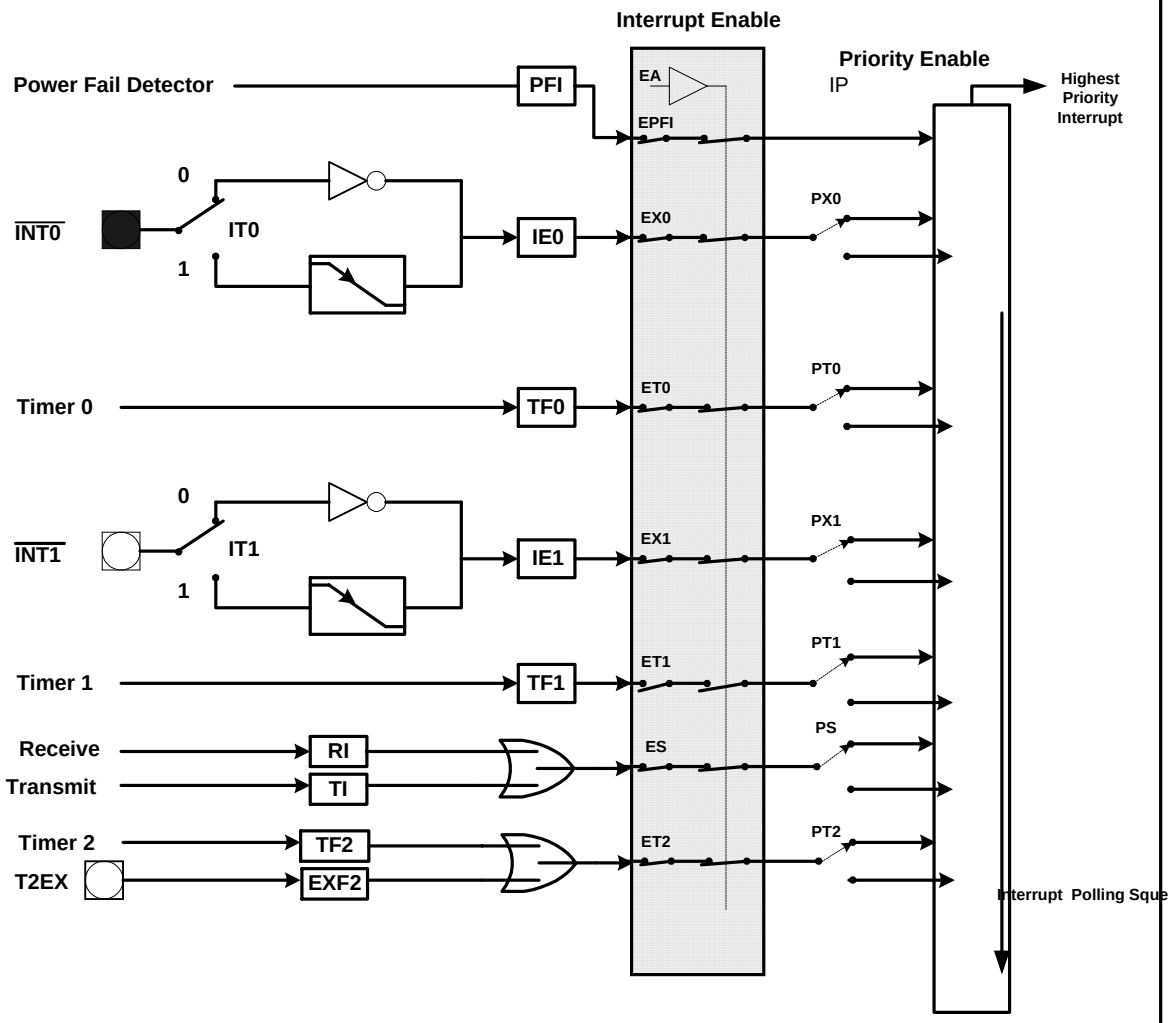
address F4H

The High-Byte of PC for External INT(ICEMD_) Entry

6.ON-CHIP PERIPHERALS

Interrupt Source	Request Flag		Priority Flag	Enable Flag	Vector Address	Priority-Within-Level	Flog Cleared by Hardware?
External Request	IE0/TCON.1		PX0/IP.0	EX0/IE.0	0003h	1	Edge-Yes Level-No
Internal Timer0/Counter0	TF0/TCON.5		PT0/IP.1	ET0/IE.1	000Bh	2	Yes
External Request	IE1/TCON.3		PX1/IP.2	EX1/IE.2	0013h	3	Edge-Yes Level-No
Internal Timer1/Counter1	TF1/TCON.7		PT1/IP.3	ET1/IE.3	001Bh	4	Yes
Internal Serial Port	Xmit	Ti/SCON.1	PS/IP.4	ES/IE.4	0023h	5	No
	Rcvr	RI/SCON.0					
Internal Timer2/Counter2	TF2/T2CON.7		PT2/IP.5	ET2/IE.5	002Bh	6	No
	EXF2/TSCON.6						

Interrupt System Table



Interrupt Control System Configuration

External Interrupt

External Interrupt $\overline{INT0}$ and $\overline{INT1}$ pins may each be programmed to be level-triggered or edge triggered, depend upon bits IT0 and IT1 in the TCON register. If IT0 or IT1 = 0, $\overline{INT0}$ or $\overline{INT1}$ is triggered by detected low at the pin. If IT0 or IT1 = 1, $\overline{INT0}$ or $\overline{INT1}$ is negative-edge triggered. External interrupts are enabled with bits EX0 and EX1in the IE register. Evens on the external interrupt pins set the interrupt flags IE0 or IE1 in TCON. These request bits are cleared by hardware vectors to service routines only if the interrupt is negative-edge triggered. If the interrupt is level-triggered, the interrupt service routine must clear the request bit. External hardware must release $\overline{INT0}$ or $\overline{INT1}$ before the service routine completes, or an additional interrupt is requested.

External interrupt pins are sampled once every oscillator clock's rising edge. A level-triggered interrupt pin held low or high for at least three clocks guarantees detection. Edge-triggered external interrupts only the request pin for one clock time. This ensures edge recognition and sets interrupt request bit EX0 or EX1. The R8032TTEX clears EX0 or EX1 automatically during service routine fetch cycles for edge-triggered interrupts.

Timer Interrupts

Three timer-interrupt request bits TF0 ,TF1 and TF2 are set by timer 0 ,timer 1 and timer 2 overflow. When timer 0 and timer 1 interrupts are generated, the bits TF0 and TF1 are cleared by an on-chip hardware vector to an interrupt service routine. Timer 2 is different from timer 0 or timer 1. Timer 2 has to clear TF2 bit by software writing when timer 2 interrupt is generated. Timer interrupts are enabled by bits ET0, ET1, and ET2 in the IE register.

Timer 2 interrupts are generated by a logical OR of bits TF2 and EXF2 in register T2CON. Neither flag is cleared by a hardware vector to a service routine. In fact, the interrupt service routine must determine if TF2 or EXEF2 generated the interrupt, and then clear the bit. Timer 2 interrupt is enabled by ET2 in register IE.

Serial Port Interrupt

Serial port interrupts are generated by the logical OR of bits RI and TI in the SCON register. Neither flag is cleared by a hardware vector to the service routine. The service routine resolves RI and TI interrupt generation and clears the serial port request flag. The serial port interrupt is enabled by bit ES in the IE register. In the same way by using serial port 1. Serial port 1 control register is

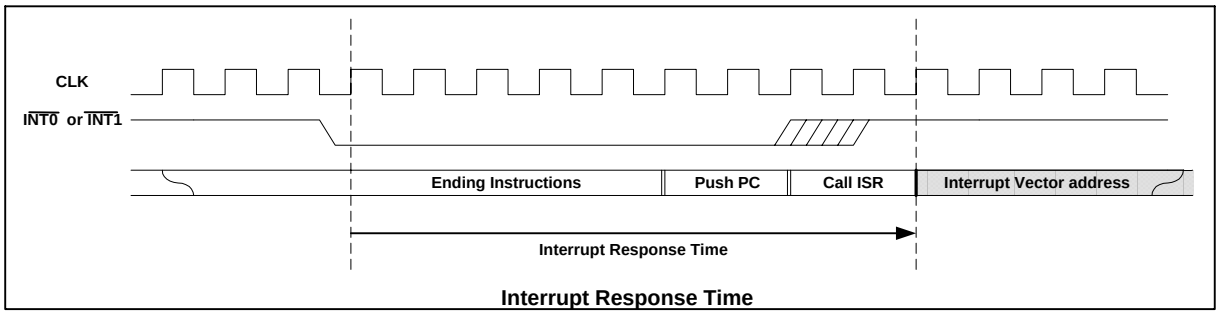
SCON1, and the buffer is SBUF1. Here is one thing to be noticed that serial port 1 only uses timer 1 to generate baud rate.

Interrupt Priority

R8032TTEX has 2 level priorities. Setting / clearing a bit in the Interrupt Priority register (IP) or Extent Interrupt Priority register (EIP) established its associated interrupt request as a high / low priority. If a low-priority level interrupt is being serviced, a high-priority level interrupt will interrupt it. However, an interrupt source cannot interrupt a service program of the same or higher level. The interrupt priority is shown on Interrupt Control System Configuration.

Interrupt Response Time

The Figure of Interrupt Response Time shows the response time is between the interrupt request being active and the interrupt service routing being executed. The minimum interrupt response time is eight clocks that when an interrupt request asserts after the ending instruction execution completes. The maximum interrupt response time is 24 clocks when an interrupt request asserts during the ending instruction, DJNZ direct, rel or others instruction sets which operation period is 16 clocks, is decoded ok. However a high priority interrupt asserts during a low priority interrupt service program is executing, both the minimum and the maximum interrupt response times are 8 clocks and 24 clocks.



Interrupt Response Time

7.TIMER/COUNTERS

Timer 0

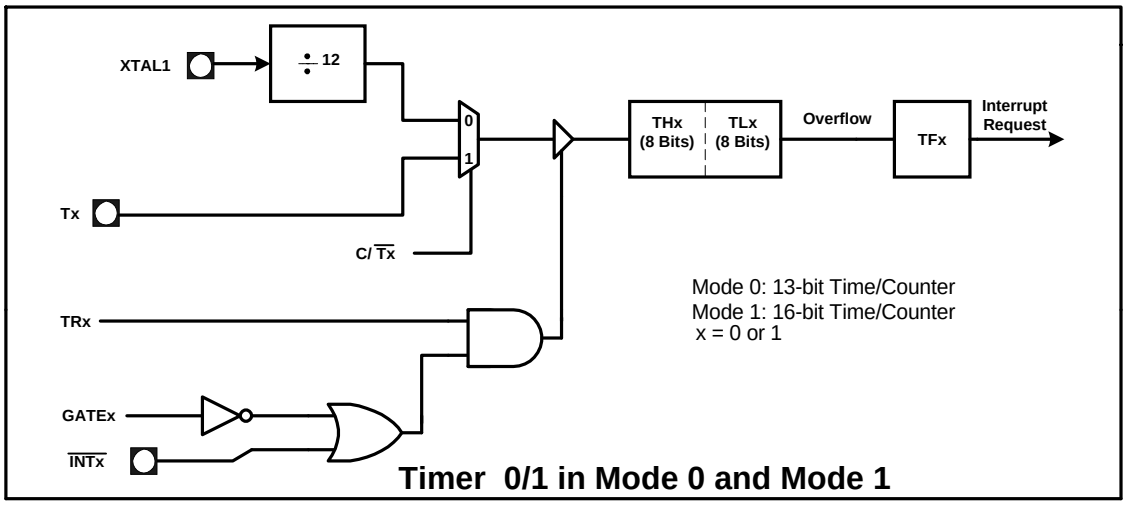
Timer 0 functions as either a timer or event counter in four modes of operation. Timer 0 is controlled by the four low-order bits of the TMOD register and bits 5, 4, 1, and 0 of the TCON register. The TMOD register selects the method of timer gating (GATE), timer or counter operation (C/$\overline{T}$), and mode of operation (M1, M0). The TCON register provides timer 0 control functions: overflag (TF0), run control (TR0), interrupt flag (IE0), and interrupt type control (IT0). For normal timer operation (GATE = 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE and TR0 allows $\overline{INT0}$ to control timer operation.

Timer0/Mode 0 (13-bit Timer)

Mode 0 configures timer 0 as a 13-bit timer which is set up as an 8-bit timer (TH0 register) with a modulo 32 prescalar implemented with the lower five bits of the TL0 register. The upper three bits of TL0 register are indeterminate and should be ignored. Prescalar overflow increments the TH0 register.

Timer 0/ Mode 1 (16-bit Timer)

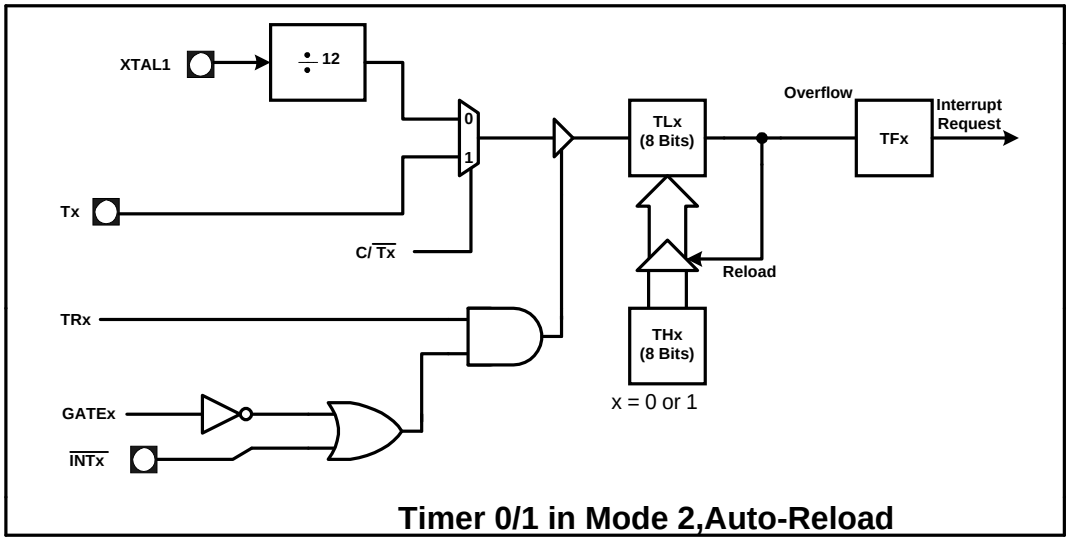
Mode 1 configures timer 0 as a 16-bit timer with TH0 and TL0 connected in cascade. The selected input increments TL0.



Timer 0/1 in Mode 0 and Mode 1

Timer 0/ Mode 2 (8-bit Timer With Auto-reload)

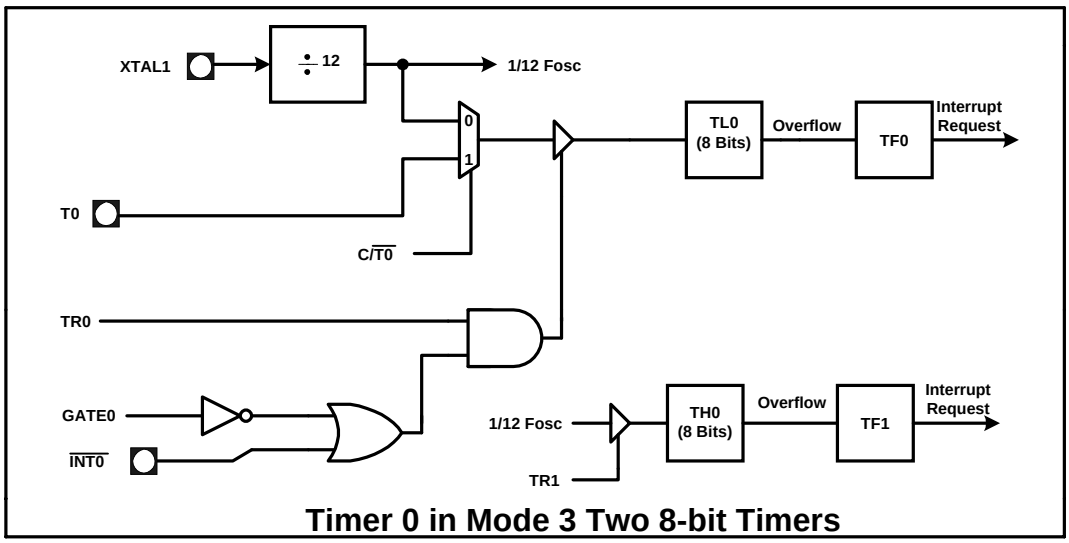
Mode 2 configures timer 0 as an 8-bit timer (TL0 register) that automatically reloads from the TH0 register. TL0 overflow sets the timer overflow flag (TF0) in the TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged.



Timer 0/1 in Mode 2,Auto-Reload

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

Mode 3 configures timer 0 such that registers TL0 and TH0 operate as separate 8-bit timers. This mode is provided for application requiring an additional 8-bit timer or counter. TL0 uses the timer 0 control bits C/$\overline{T}$ and GATE in TMOD, and TR0 in TCON in the normal manner. TH0 is locked into a timer function (counting Fosc/12) and takes over use of the timer 1 interrupt (TF1) and run control (TR1) bits. Thus, operation of timer 1 is restricted when timer 0 is in mode 3.



Timer 0 in Mode 3 Two 8-bit Timers

Timer 1

Timer 1 functions as either a timer or event counter in three modes of operation. The logical configuration for modes 0,1,and 2 are the same as that of Timer 0. Timer 1's mode 3 is a hold-count mode.

Timer 1 is controlled by the four high-order bits of the TMOD register and bits 7,6,3,and 2 of the TCON register. The TMOD register selects the method of timer gating (GATE),timer or counter operation (C/$\overline{T}$), and mode of operation (M1 and M0). The TCON register provides timer 1 control functions: overflow flag (TF1),run control (TR1),interrupt flag(IE1), and interrupt type control (IT1).

For normal timer operation (GATE = 0), setting TR1 allows timer register TL1 to be incremented by the selected input. Setting GATE and TR1 allows external pin $\overline{INT1}$ to control timer operation. This setup can be used to make pulse width measurements.

Timer 1/ Mode 0 (13-bit Timer)

Mode 0 configures timer 0 as a 13-bit timer, which is set up as an 8-bit timer (TH1 register) with a modulo-32 prescalar implemented with the lower 5 bits of the TL1 register. The upper 3 bits of the TL1 register are ignored. Prescalar overflow increment the TH1 register.

Timer1/ Mode 1 (16-bit Timer)

Mode 1 configures timer 1 as a16-bit timer with TH1 and TL1 connected in cascade. The selected input increments TL1.

Timer 1/ Mode 2 (8-bit Timer)

Mode 2 configures timer 1 as an 8-bit timer (TL1 register) with automatic reload from the TH1 register on overflow. Overflow from TL1 sets overflow flag TF1 in the TCON register and reloads

TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Timer 1/ Mode3 (Halt)

Placing timer in mode 3 causes it to halt and its count. This can be used to halt timer 1 when the TR1 run control bit is not available, i.e., when timer 0 is in mode 3.

TIMER 2

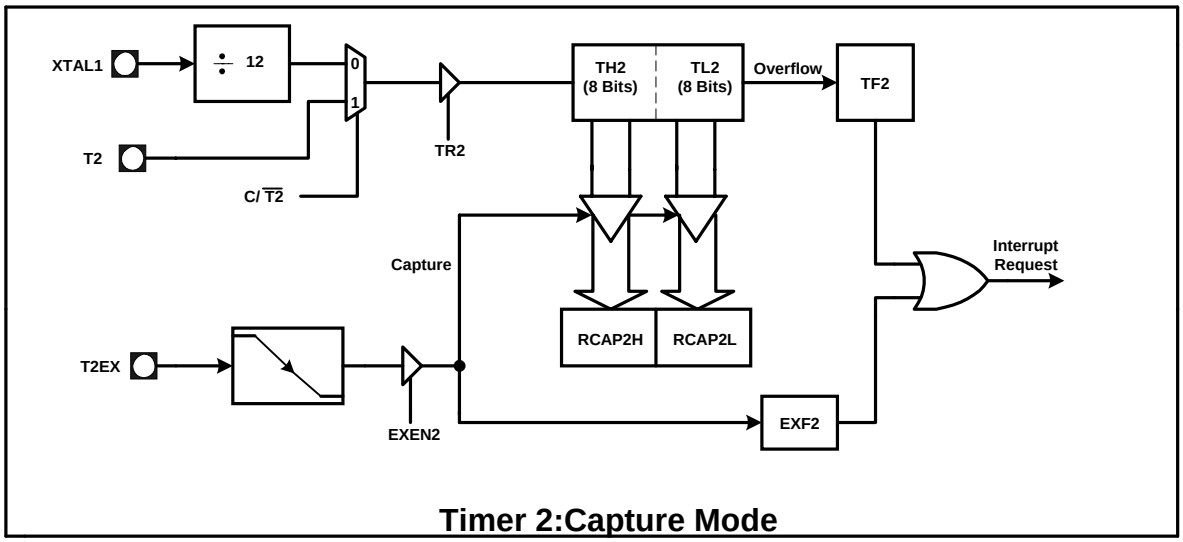
Timer 2 is a 16-bt timer/count is maintained by two eight-bit timer register, TH2 and TL2, connected in cascade. The timer/counter 2 mode control register T2MOD and the timer /counter control register T2CON control the operation of timer 2.

Timer 2 provides the following operating modes: capture mode, auto-reload mode, baud rate generator mode, and programmable clock-out mode. Select the operating mode with T2MOD and TCON register bits as shown in table of Timer 2 Modes of Operation. Auto-reload is the default mode. Setting RCLK and/or TCLK selects the baud rate generator mode.

Timer 2 operation is similar to timer 0 and timer 1. C/$\overline{T2}$ selects Fosc/12 (timer operation) or external pin T2 (counter operation) as the timer register input. Setting TF2 to be incremented by the selected input.

Timer 2/ Capture Mode

In the capture mode, timer 2 function as a 16-bit timer or counter. An overflow condition sets bit TF2,which you can use to request an interrupt. Setting the external enable bit EXEN2 allows the RCAP2Hand RCAP2L registers to capture the current value in timer registers TH2 and TL2 in response to a l-to-0 transition at external input T2EX. The transition at T2EX also sets bit EXF2 on T2CON. The EXF2 bit, like TF2, can generate an interrupt. TR2 must be enabled when running this mode.



Timer 2:Capture Mode

Timer 2/ Auto-reload Mode

The auto-reload mode configures timer 2 as a 16-bit timer or event counter with automatic reload. The timer operates an as an up counter or as an up/down counter, as determined by the down counter enable bit (DCEN). At device reset, DCEN is cleared, so in the auto-reload mode, timer 2 defaults to operation as an up counter. TR2 must be enabled when running this mode.

Up Counter Operation

When DCEN = 0, timer 2 operates as an up counter. If EXEN = 0, timer 2 counts up to FFFFH and sets the TF2 overflow flag. The overflow condition loads the 16-bit value in the reload/capture registers (RCAP2H, RCAP2L) into the timer registers (TH2, TL2). The values in RCAP2H and RCAP2L are preset by software.

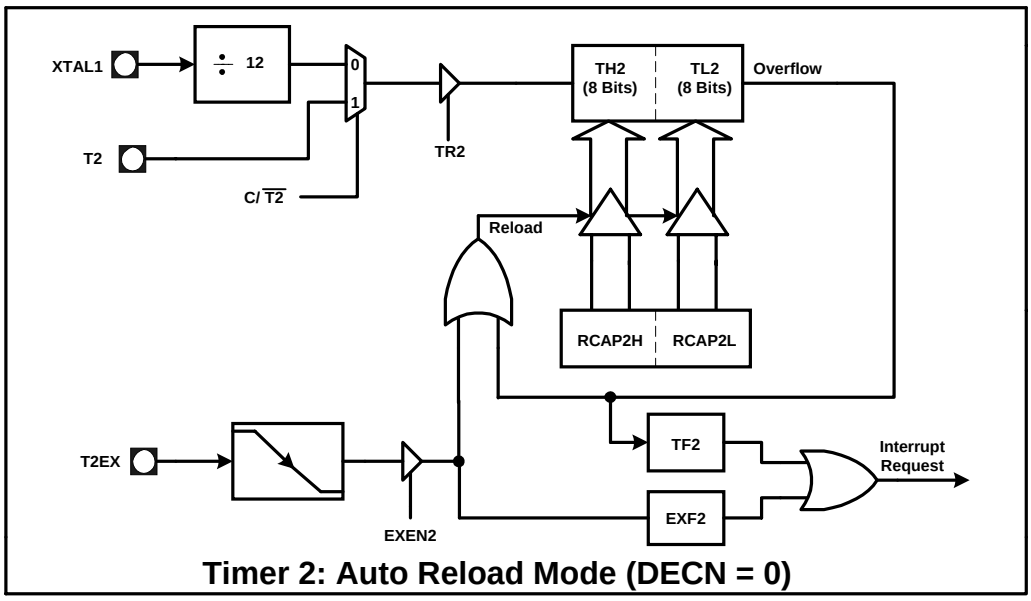
If EXEN2 = 1, the timer registers are reloaded by either a timer overflow or a high-to-low transition at external input T2EX. This transition also sets the EXF2 bit in the T2CON register. Either TF2 or EXF2 bit can generate a timer 2 interrupt request. TR2 must be enabled when running this mode.

Up/Down Counter Operation

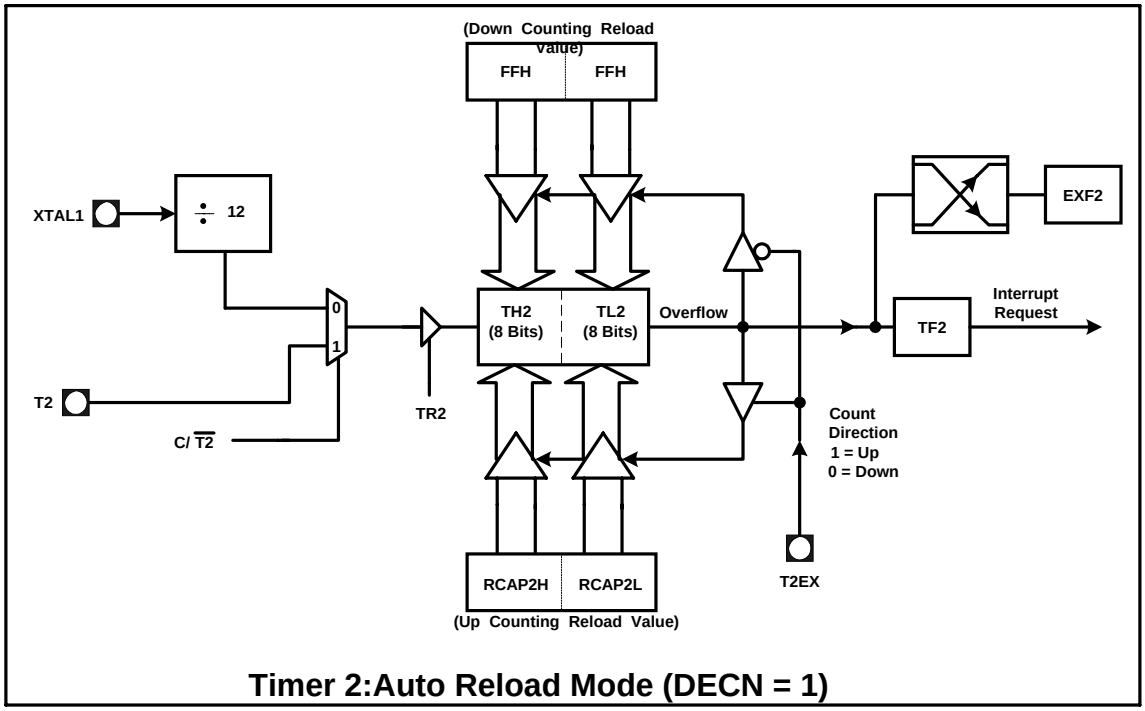
When DCEN = 1, timer 2 operates as an up/down counter. External pin T2EX controls the direction of the count. When T2EX is high, timer 2 counts up. The timer overflow occurs at FFFFH which sets the timer 2 overflow flag (TF2) and generates an interrupt request. The overflow also causes the 16-bit value in RCAP2H and RCAP2L to be loaded into the timer registers TH2 and TL2.

When T2EX is low, timer 2 counts down. Timer underflow occurs when the count in the timer registers (TH2, TL2) equals the value stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and reloads FFFFH into the timer registers.

The EXF2 bit toggles when timer 2 overflows or underflows changing the direction of the count. When timer 2 operates as an up/down counter, EXF2 does not generate an interrupt. This bit can be used to provide 17-bit resolution. TR2 must be enabled when running this mode.



Timer 2: Auto Reload Mode (DECN = 0)



Timer 2:Auto Reload Mode (DECN = 1)

Timer 2/ Baud Rate Generator Mode

This mode configures timer 2 as a baud rate generator for use with the serial port. Select this mode by setting the RCLK and/ or TCLK bits in T2CON.

Timer 2/ Clock-out Mode

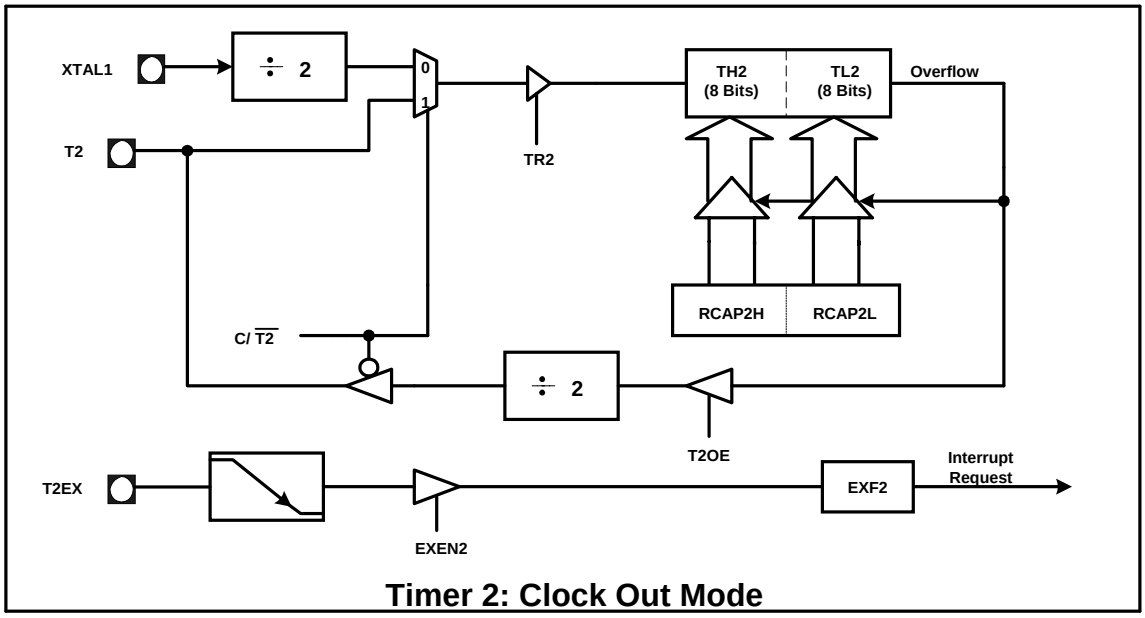
In the clock-out mode, timer 2 functions as a 50%-duty-cycle, variable-frequency clock. The input

clock increments TL0 at frequency Fosc/2. The timer repeatedly counts to overflow from a preloaded value. At overflow, the contents of the RCAP2H and RCAP2L registers are loaded into TH2/TL2. In this mode, timer 2 overflows do not generates interrupts. The formula gives the clock-out frequency as a function of the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$\text{Clock-out Frequency} = \frac{Fosc}{4X(65536 - RCAP2H, RCAP2L)}$$

Timer 2 Modes of Operation

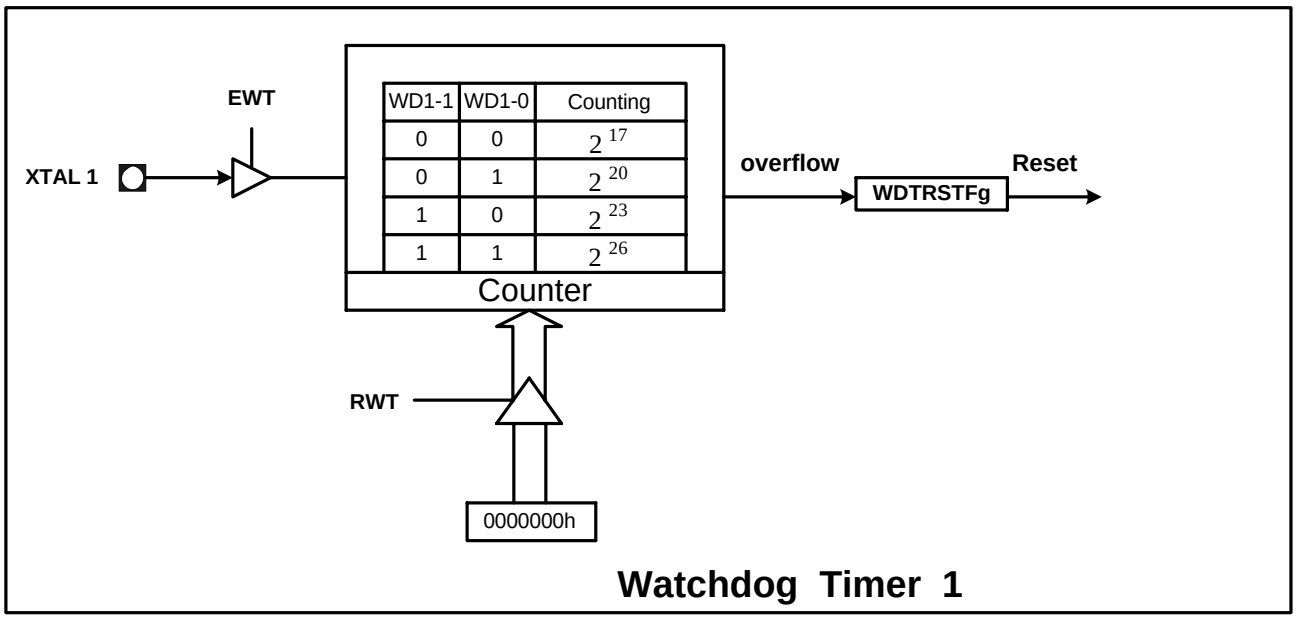
Mode	RCLK OR TCLK (in T2COON)	CP/RL2# (in T2MOD)	T2OE (in T2MOD)
Auto-reload Mode	0	0	0
Capture Mode	0	1	0
Baud Rate Generator Mode	1	X	X
Programmable Clock-Out	X	0	1



Timer 2: Clock Out Mode

Watchdog Timer

The watchdog timer has system reset functions. User can set WD1-1, WD1-0 (in register CKCON, 8Eh) to choose 2^{17}, 2^{20}, 2^{23} or 2^{26} counter for Watchdog Timer. After the Watchdog Timer counting the specific counter and occurring an overflow, sets WDTRSTFg (in register WDCON, D8h) and finally resets the R8032TTEX. If R8032TTEX has been reset by Watchdog Timer , WDTRSTFg remains one and POR (in register WDCON, D8h) is zero. On the other hand, if R8032TTEX has been power-on reset, WDTRSTFg is zero and POR one.



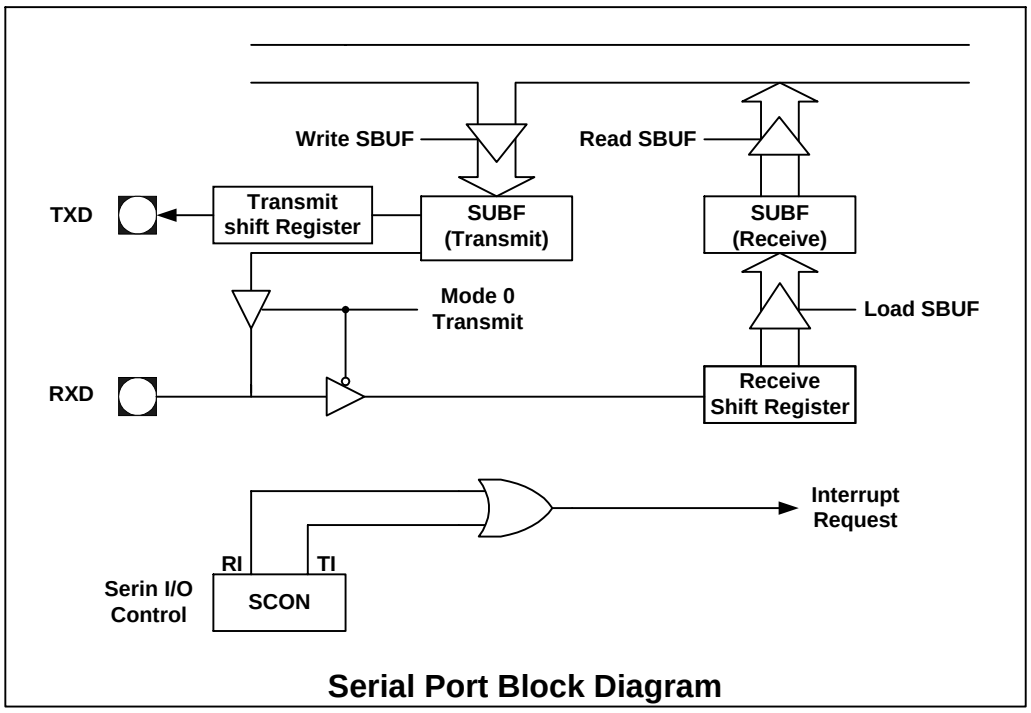
SERIAL I/O PORT

The serial I/O port provides both synchronous and asynchronous communication modes. It operates as a universal asynchronous receiver and transmitter (UART) in three full-duplex modes (modes 1, 2, and 3). Asynchronous transmission and reception can occur simultaneously and at different baud rates. The serial port also operates in a single synchronous mode (mode 0).

The synchronous mode (mode 0) operates at a single baud rate. Mode 2 operates at two baud rates. Modes 1 and 3 operate over a wide range of baud rates, which are generated by timer 1 and timer 2.

The serial port signals are defined in Table of Serial Port Signals, and the serial port special function registers (SBUF, SCON) are described in the section of Special Function Registers.

For the three asynchronous modes, the UART transmits on the TXD pin and receives on the RXD pin. For the synchronous mode (mode 0), the UART outputs a clock signal on the TXD pin and sends and receives messages in the RXD pin. The SBUF register, which holds received bytes and bytes to be transmitted, actually consists of two physically different registers. To send, software writes a byte to SBUF; to receive, software reads SBUF. The receive shift register allows reception of a second byte before the byte has been read from SBUF. However, if software has not read the first byte by the time the second byte is received, the second byte will overwrite the first. the UART sets interrupt bits TI and RI on transmission and reception, respectively. These two bits share a single interrupt request and interrupt vector.



Serial Port Block Diagram

Serial Port Signals

Function Name	Type	Description	Multiplexed With
TXD	O	**Transmit Data.** In mode 0, TXD transmits the clock signal. In modes 1, 2,and 3,TXD transmits serial data.	P3.1
RXD	I/O	**Receive Data.** In mode 0, RXD transmits and receives serial data. In mode 1, 2,and 3, RXD receives serial data.	P3.0

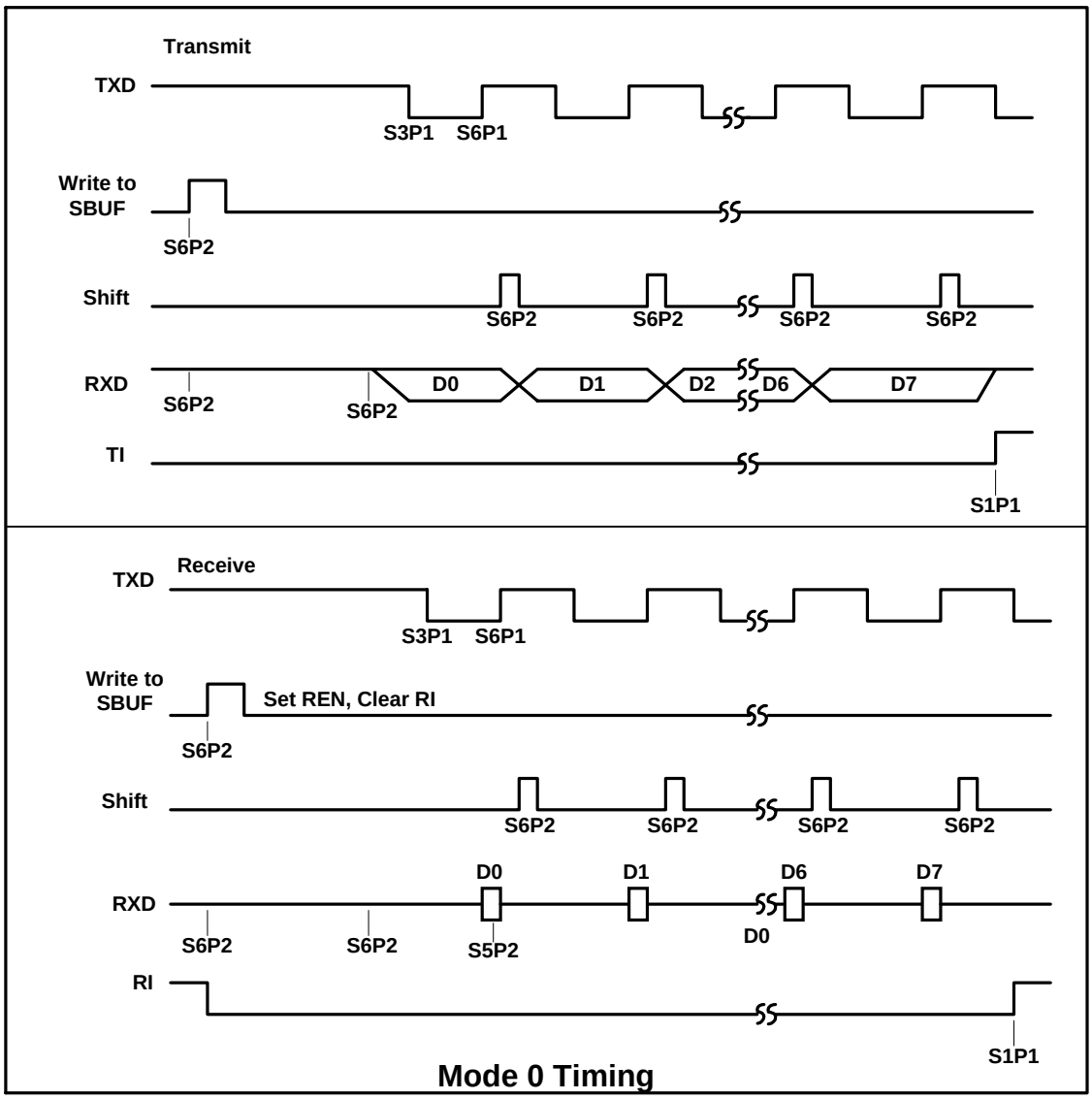
Synchronous Mode (Mode 0)

Mode 0 is a half-duplex, synchronous mode, which is commonly used to expand I/O capabilities of device with shift registers. The transmit data (TXD) pin outputs a set of eight clock pulses which the receive data (RXD) pin transmits or receives a byte of data. The eight data bits are transmitted and received least-significant bit (LSB) first. shifts occur in the last phase (S6P2) of every peripheral cycle, which corresponds to a baud rate of Fosc/12.

Trans mission (Mode 0)

Follow these steps to begin a transmission:

1. Write to the SCON register, clearing bits SM0,SM1, and REN.
2. Write the byte to be transmitted to the SBUF register. This write starts the transmission .



Mode 0 Timing

Hardware executes the write to SBUF in the last phase (S6P2) of a peripheral cycle. At S6P2 of the following cycle, hardware shifts the LSB (D0) onto the RXD pin. At S3P1 of the nest cycle, the TXD pin goes low for the first clock-signal pulse. Shifts continue every peripheral cycle. In the ninth cycle after the write to SBUF, the MSB (D7) is on the RXD pin. At the beginning of the tenth cycle, hardware drives the RXD pin high and assert TI (S1P1) to indicate the end of the transmission.

Reception (Mode 0)

To start a reception in mode 0, write to the SCON register. Clear bits SM0, SM1, and RI and set the REN bit.

Hardware executes the write to SCON in the last phase (S6P2) of a peripheral cycle. In the second peripheral cycle following the write to SCON, TXD goes low at S3P1 for the first clock-signal

pulse, and the LSB (D0) is sampled on the RXD pin at S5P2. The D0 bit is then shifted into the shift register. After eight shifts at S6P2 of every peripheral cycle, the LSB (D7) is shifted into the shift register, and hardware asserts RI (S1P1) to indicate a completed reception. Software can then read the received byte from SBUF.

Asynchronous Modes (Modes 1, 2, and 3)

The serial port has three asynchronous modes of operation.

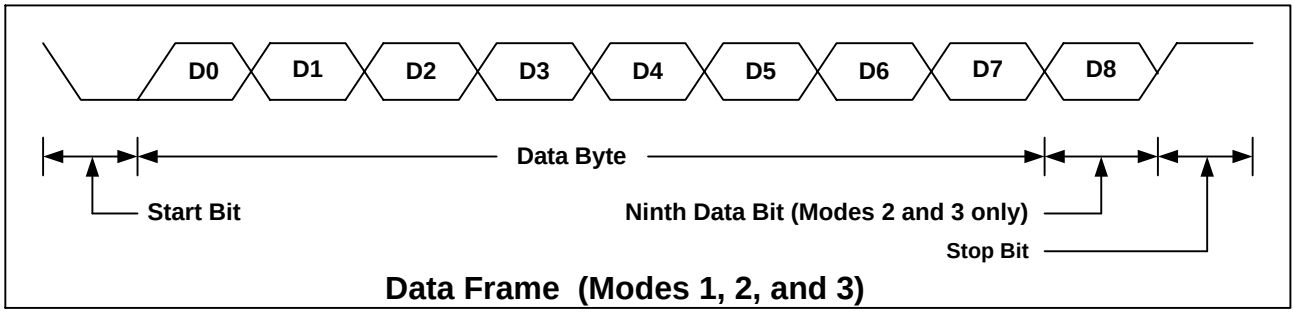
Mode 1

Mode 1 is a full-duplex, asynchronous mode. The data frame consists of 10 bits: one start bit, eight data bits, and one stop bit. Serial data is transmitted on the TXD pin and received on the RXD pin. When a message is received, the stop bit is read in the RB8 bit in the SCON register. The baud rate is generated by overflow of timer 1 or timer 2.

Mode 2 and 3

Modes 2 and 3 are full-duplex, asynchronous modes. The data frame consists of 11 bits: one start bit, eight data bits (transmitted and received LSB first), one programmable ninth data bit, and one stop bit is read from the RB8 bit in the SCON register. On transmit, the ninth data bit is written to the TB8 bit in the SCON register. Alternatively, you can use the ninth bit as a command/data flag.

- In mode 2, the baud rate is programmable to 1/32 or 1/64 of the oscillator frequency.
- In mode 3, the baud rate is generated by overflow of timer 1 or timer 2.



Data Frame (Modes 1, 2, and 3)

Transmission (Modes 1, 2, 3)

Follow these steps to initiate a transmission:

1. Write to the SCON register. Select the mode with the SM0 and SM1 bits, and clear the REN bit. For modes 2 and 3, also write the ninth bit to the TB8 bit.
2. Write the byte to be transmitted to the SBUF register. This write stars the transmission.

Reception (Modes 1, 2, 3)

To prepare for a reception, set the REN bit in the SCON register. The actual reception is then initiated by a detected high-to-low transition on the RXD pin.

Baud Rates

Baud Rate for Mode 0

The baud rate for mode 0 id fixed at Fosc/12.

Baud Rtes for Mode 2

Mode 2 has two baud rates, which are selected by the SMOD bit in the PCON register. The following expression defines the baud rate:

$$\text{Sreial I/O Mode 2 Baud Rate} = 2^{SMOD} \times \frac{Fosc}{64}$$

Baud Rates for Modes 1 and 3

In modes 1 and 3, the baud rate is generated by overflow of timer (default) and/or timer 2. You may select either or both timer(s) to generate the baud rate(s) for the transmitter and/or the receiver.

Timer 1 Generated Baud Rates (Mode 1 and 3)

Timer 1 is the default baud rate generator for the transmitter and the receiver in modes 1 and 3. The baud rate is determined by the timer 1 overflow rate and the value of SMOD, as shown in the following formula:

$$\text{Sreial I/O Mode 1 and 3 Baud Rate} = 2^{SMOD} \times \frac{\text{Timer 1 Onerflow Rate}}{32}$$

Selecting Timer 1 as the Baud Rate Generator

To select timer 1 as the baud rate generator:

- Disable the timer interrupt by clearing the IE0 register.
- Configure timer 1 as a timer or an event counter (set or clear the C/T bit in the TMOD register).

$$\text{Sreial I/O Mode 1 and 3 Baud Rate} = 2^{SMOD} \times \frac{Fosc}{32 \times 12 \times [256 - (Th1)]}$$

- Select timer mode 0-3 by programming the M1, M0 bits in the TMOD register.

In most applications, timer 1 is configured as a timer in auto-reload mode (high nibble of TMOD = 0010B). The resulting baud rate is defined by the following expression:

Timer 1 can generate very low baud rates with the following setup:

- Enable the timer 1 interrupt by setting the ET1 bit in the IE register.
- Configure timer 1 to run as a 16-bit timer (high nibble of TMOD = 0001B).

- Use the timer 1 interrupt to initiate a 16-bit software reload.

Timer 2 Generated Baud Rates (Modes 1 and 3)

Timer 2 may be selected as the baud rate generator for the transmitter and/or receiver. The timer 2 baud rate generator mode is similar to the auto-reload mode. A rollover in the TH2 register reloads registers TH2 and TL2 with the 16-bit value on registers RCAP2H and RCAP2L, which are preset by software.

The timer 2 baud rate is expressed by the following formula:

$$\textbf{Sreial I/O Mode 1 and 3 Baud Rate} = \frac{\textbf{Timer 2 Onerflow Rate}}{\textbf{16}}$$

Selecting Timer 2 as the Baud Rate Generator

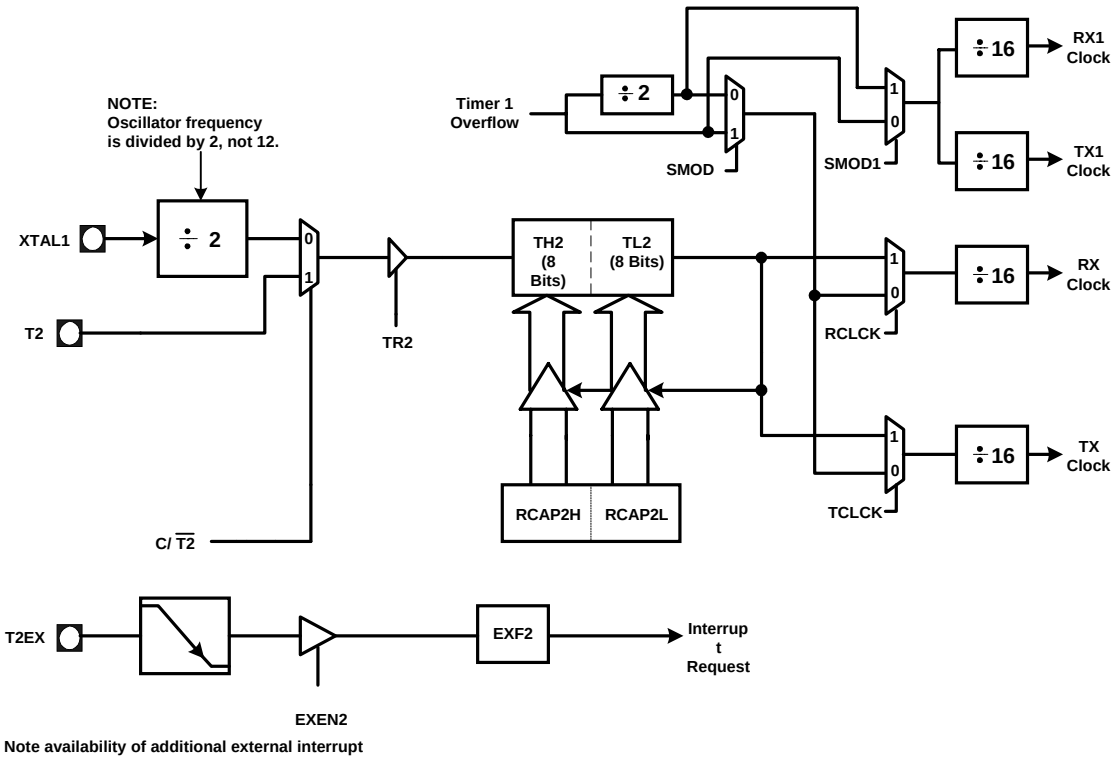
To select timer 2 as the baud rate generator for the transmitter and/or receiver, program the RCLCK and TCLCK bits in the T2CON register. Setting RCLK and/or TCLK puts timer 2 into its baud rate generator mode. In this mode, a rollover in the TH2 register does not set the TF2 bit in the T2CON register. Also, a high-to-low transition at the T2EX pin sets the EXF2 bit in the T2CON register but does not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). You can use the TT2EX pin as an additional external interrupt by setting the EXEN2 bit in T2CON.

NOTE : Turn the timer off (clear the TR2 bit in the T2CON register) before accessing registers TH2, TL2, RCAP2H,and RCAP2L.

You may configure timer 2 as a timer or a counter. In most applications, it is configured for timer operation (i.e., the C/T2 bit is clear in the T2CON register).

Selecting the Baud Rate Generator(s)

RCLK Bit	TCLCK Bit	Receiver Baud Rate Generator	Transmitter Baud Rate generator
0	0	Timer 1	Timer 1
0	1	Timer 1	Timer 2
1	0	Timer 2	Timer 1
1	1	Timer 2	Timer 2



Timer 2 in Baud Rate Generator Mode

Note that timer 2 increments every state time (2Tosc) when it is in the baud rate generator mode. In the baud rate formula that follows, "RCAP2H,RCAP2L" denotes the contents of RCAP2H and RCAP2L taken as a 16-bit unsigned integer:

$$\text{Sreial I/O Mode 1 and 3 Baud Rate} = \frac{\text{Fosc}}{32 \times [65536 - (\text{RCAP2H,RACAP2L})]}$$

NOTE : When timer 2 is configured as a timer and is in baud rate generator mode, do not read or write the TH2 or TL2 registers. The timer is being incremented every state time, and the result of a read or write may not be accurate. In addition, you may read, but not write to, the RCAP2 registers; a write may overlap a reload and cause write and/or reload errors.

Serial I/O Port 1

Serial I/O port 1 is the same as serial I/O port mentioned above. RXD1 is at P1.2 and TXD1 at P1.3. The Serial I/O port 1 has its own buffer (SBUF1, C1h) and control register (SCON1, C0h). All functions and structures are the same as serial I/O port. But the only difference is that serial I/O port 1 only uses timer 1 for baud rate at mode 1 and mode 3. The double baud rate bit SMODE1 is at WDCON (D8h) register.

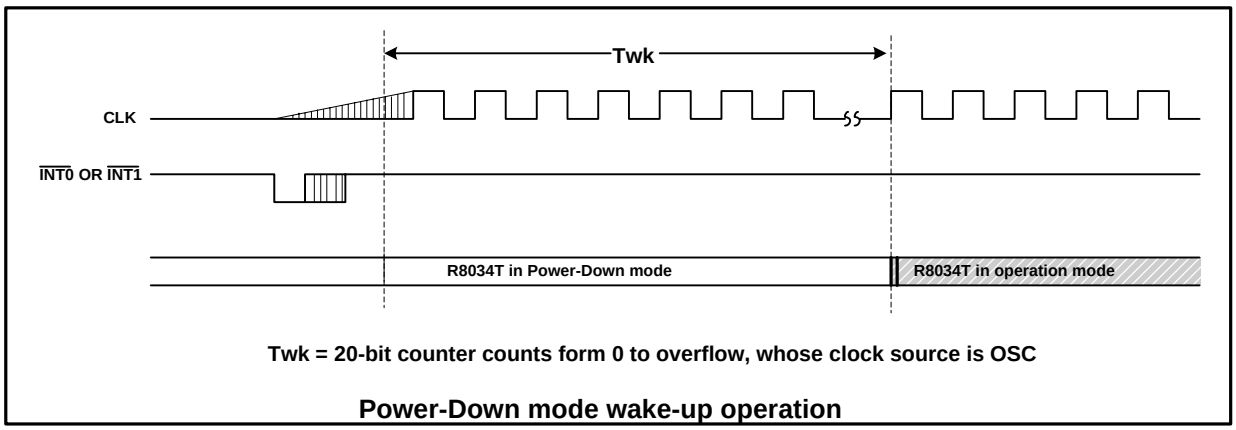
8.POWER-DOWN and IDLE-MODE

Idle-Mode

When set IDL bit in PCON(87h), the μP will enter idle mode. In idle mode , the μP is idle while all the on-chip peripherals remain active. The internal RAM and SFRs registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Power-Down Mode

When set PD bit in PCON(87h), the μP will enter power down-mode .In the power-down mode, the oscillator is stopped. Before entering power-down mode, the RNGEN bit in VDT(D9h) should be clear for saving power consumption. The power down mode can be wakened up by the hardware reset or by the external enable interrupt with level trigger activation (ITx in register TCON is set to 0). The Program Counter, internal RAM and SFRs registers retain their values and will not be changed after the power-down mode is terminated by external interrupt. The reset will restart the CPU, while the SFRs with initial values and the internal RAM retain their values.



Twk = 20-bit counter counts form 0 to overflow, whose clock source is OSC

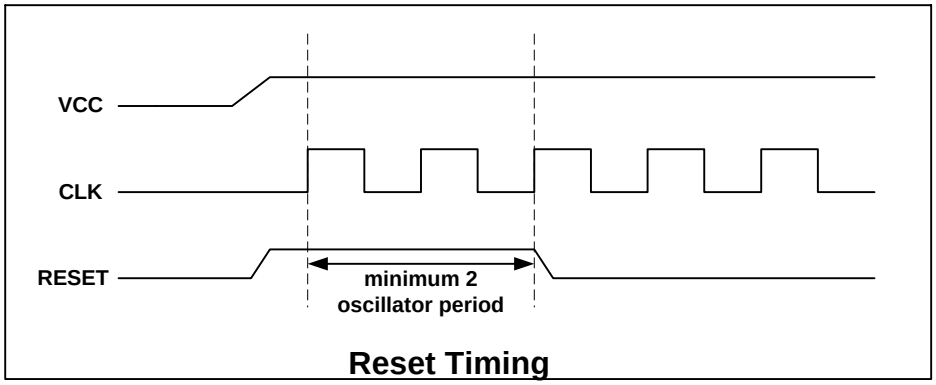
Power-Down mode wake-up operation

Status of External Pins During Idle and Power-Down

Mode	ALE	$\overline{\text{PSEN}}$	Port 0- Port 3
Idle	0	1	Retain the values before enter Idle Mode
Power-Down	0	1	FFH

9.RESET

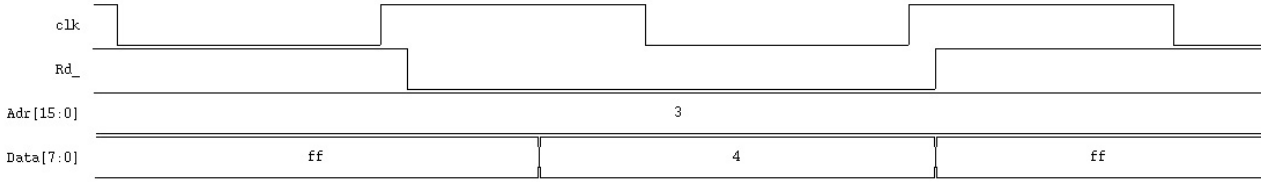
Processor initialization is accomplished with activation of the RESET pin. To reset the processor, this pin should be held high for at least two oscillator periods.



Reset Timing

10.Memory Access

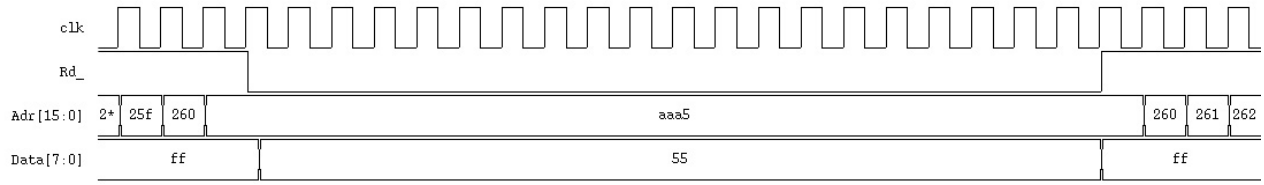
1. Data Memory Read Cycle Timing—Normal Case(1T)



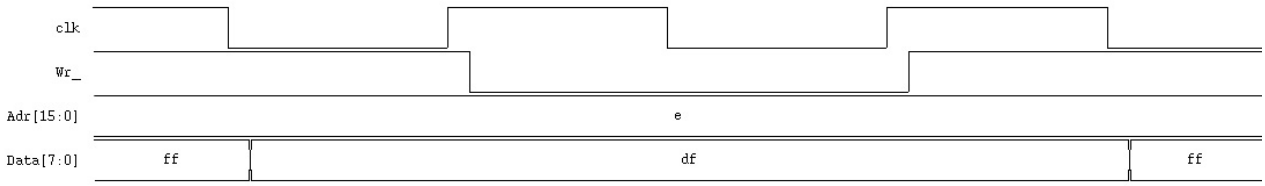
2. Data Memory Read Cycle Timing—Insert Wait State Case

CKCON [2:0]: MD2, MD1, MD0 [101]

Normal + Wait State Cycle = 5x4 clock cycle = 20 clock cycle



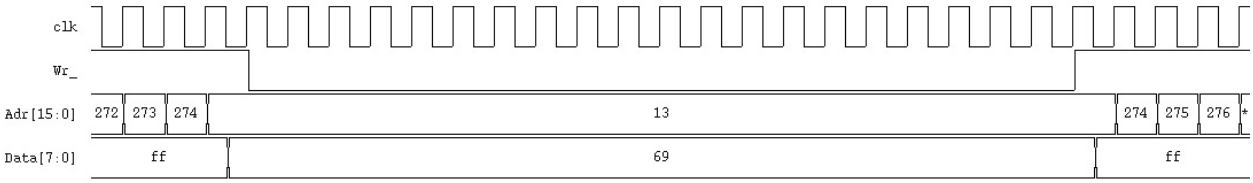
3. Data Memory Write Cycle Timing—Normal Case(1T)



4.Data Memory Write Cycle Timing—Insert Wait State Case

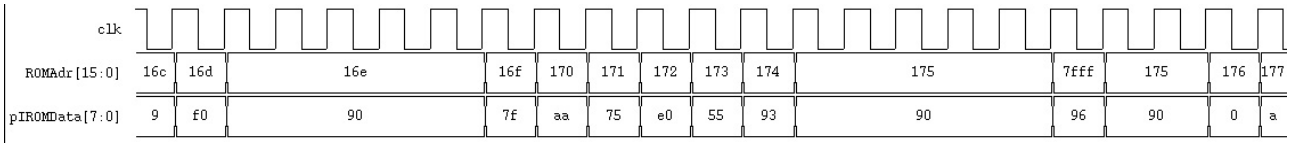
CKCON [2:0]: MD2, MD1, MD0 [101]

Normal + Wait State Cycle = 5x4 clock cycle = 20 clock cycle



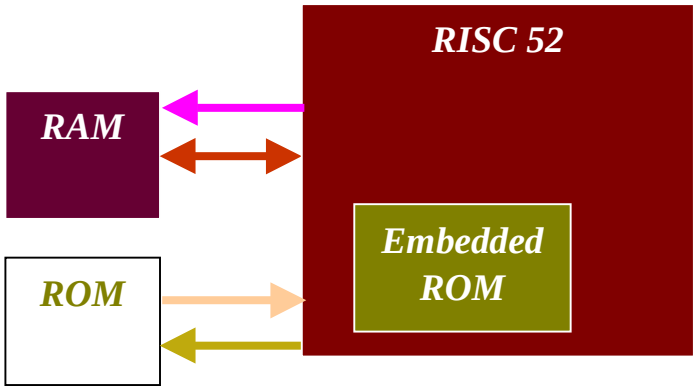
5. External ROM

External ROM data access timing cycle



11.Application Diagram

-R8032TTEX



12. Instruction Cycle

12.1 Instruction Cycle Table(TTEX)

Logical Operations

Instruction		**Oscillator Period**		
		80C32	**R8032TT**	**R8032TTEX**
ANL	**A, Rn**	*12*	*2*	*2*
ANL	*A, dir*	*12*	*3*	*3*
ANL	*A, @Ri*	*12*	*4*	*4*
ANL	*A, #data*	*12*	*2*	*2*
ANL	*dir, A*	*12*	*3*	*3*
ANL	*dir, #data*	*24*	*4*	*4*
ORL	*A, Rn*	*12*	*2*	*2*
ORL	*A, dir*	*12*	*3*	*3*
ORL	*A, @Ri*	*12*	*4*	*4*
ORL	*A, #data*	*12*	*2*	*2*
ORL	*dir, A*	*12*	*3*	*3*
ORL	*dir, #data*	*24*	*4*	*4*
XRL	*A, Rn*	*12*	*2*	*2*
XRL	*A, dir*	*12*	*3*	*3*
XRL	*A, @Ri*	*12*	*4*	*4*
XRL	*A, #data*	*12*	*2*	*2*
XRL	*dir, A*	*12*	*3*	*3*
XRL	*dir, #data*	*24*	*4*	*4*
CLR	*A*	*12*	*1*	*1*
CPL	*A*	*12*	*1*	*1*
RL	*A*	*12*	*1*	*1*
RLC	*A*	*12*	*1*	*1*
RR	*A*	*12*	*1*	*1*
RRC	*A*	*12*	*1*	*1*
SWAP	*A*	*12*	*1*	*1*

DATA Transfer

Instruction		**Oscillator Period**		
		80C32	**R8032TT**	**R8032TTEX**
MOV	**A, Rn**	*12*	*2*	*2*
MOV	*A, dir*	*12*	*3*	*3*
MOV	*A, @Ri*	*12*	*4*	*4*
MOV	*A, #data*	*12*	*2*	*2*
MOV	*Rn, A*	*12*	*1*	*1*
MOV	*Rn, dir*	*24*	*3*	*3*
MOV	*Rn, #data*	*12*	*2*	*2*
MOV	*dir, A*	*12*	*2*	*2*
MOV	*dir, Rn*	*24*	*3*	*3*
MOV	*dir, dir*	*24*	*4*	*4*
MOV	*dir, @Ri*	*24*	*4*	*4*
MOV,	*dir, #data*	*24*	*3*	*3*
MOV	*@Ri, A*	*12*	*3*	*3*
MOV	*@Ri, dir*	*24*	*4*	*4*

MOV	*@Ri, #data*	*12*	*3*	*3*
MOV	*DPTR,#data16*	*24*	*3*	*3*
MOVC	*@A+DPRT*	*24*	*10*	*7*
MOVC	*A, @A+PC*	*24*	*7*	*5*
MOVX	*A, @Ri*	*24*	*10*	*6*
MOVX	*A, @DPTR*	*24*	*11*	*4*
MOV	*@Ri, A*	*24*	*11*	*6*
MOVX	*@DPTR, A*	*24*	*11*	*4*
PUSH	*dir*	*24*	*3*	*3*
POP	*dir*	*24*	*4*	*4*
XCH	*A, Rn*	*12*	*4*	*4*
XCH	*A, dir*	*12*	*4*	*4*
XCH	*A, @Ri*	*12*	*6*	*6*
XCHD	*A, @Ri*	*12*	*8*	*8*

Arithmetic Operations

Instruction		**Oscillator Period**		
		80C32	**R8032TT**	**R8032TTEX**
ADD	*A, Rn*	*12*	*2*	*2*
ADD	**A, dir**	*12*	*3*	*3*
ADD	*A, @Ri*	*12*	*4*	*4*
ADD	*A, #data*	*12*	*2*	*2*
ADDC	*A, Rn*	*12*	*2*	*2*
ADDC	**A, dir**	*12*	*3*	*3*
ADDC	*A, @Ri*	*12*	*4*	*4*
ADDC	*A, #data*	*12*	*2*	*2*
SUBB	*A, Rn*	*12*	*2*	*2*
SUBB	**A, dir**	*12*	*3*	*3*
SUBB	*A, @Ri*	*12*	*4*	*4*
SUBB	*A, #data*	*12*	*2*	*2*
INC	*A*	*12*	*1*	*1*
INC	*Rn*	*12*	*2*	*2*
INC	*dir*	*12*	*3*	*3*
INC	*@Ri*	*12*	*4*	*4*
DEC	*A*	*12*	*1*	*1*
DEC	*Rn*	*12*	*2*	*2*
DEC	*dir*	*12*	*3*	*3*
DEC	*@Ri*	*12*	*4*	*4*
INC	*DPTR*	*24*	*4*	*2*
MUL	*AB*	*48*	*1*	*1*
DIV	*AB*	*48*	*14*	*14*
DA	*A*	*12*	*1*	*1*

Boolean Variable Manipulation

Instruction		**Oscillator Period**		
		80C32	**R8032TT**	**R8032TTEX**
CLR	*C*	*12*	*1*	*1*
CLR	**bit**	*12*	*3*	*3*

SETB	*C*	*12*	*1*	*1*
SETB	*bit*	*12*	*3*	*3*
CPL	*C*	*12*	*1*	*1*
CPL	**bit**	*12*	*3*	*3*
ANL	*C, bit*	*24*	*3*	*3*
ANL	*C, bit*	*24*	*3*	*3*
ORL	*C, bit*	*24*	*3*	*3*
ORL	**C, bit**	*24*	*3*	*3*
MOV	*C, bit*	*12*	*3*	*3*
MOV	*bit, C*	*24*	*3*	*3*
JC	*Rel*	*24*	*2*	*2*
JNC	*Rel*	*24*	*3*	*3*
JB	*Bit, rel*	*24*	*5*	*5*
JNB	*Bit, rel*	*24*	*5*	*5*
JBC	*Bit, rel*	*24*	*5*	*5*

Programming Branching

Instruction		**Oscillator Period**		
		80C32	**R8032TT**	**R8034TTEX**
ACALL	*addr11*	*24*	*4*	*4*
LCALL	**addr16**	*24*	*6*	*6*
RET		*24*	*7*	*6*
RETI		*24*	*7*	*6*
AJMP	*addr11*	*24*	*2*	*2*
LJMP	**addr16**	*24*	*3*	*3*
SJMP	*rel*	*24*	*3*	*3*
JMP	*@A+DPTR*	*24*	*4*	*4*
JZ	*rel*	*24*	*3*	*3*
JNZ	**rel**	*24*	*3*	*3*
CJNE	*A, dir, rel*	*24*	*5*	*4*
CJNE	*A, #data,rel*	*24*	*4*	*4*
CJNE	*Rn, dir, rel*	*24*	*4*	*4*
CJNE	*@Ri, dir, rel*	*24*	*5*	*5*
DJNZ	Rn, rel	*24*	*4*	*4*
DJNZ	*dir, rel*	*24*	*5*	*4*
NOP		*12*	*1*	*1*

13.Revision History

Rev.	Date	History
1.0	2001/10/17	Formal release
1.1	2002/05/02	Fix:MOV Rn,dir (from I/O)