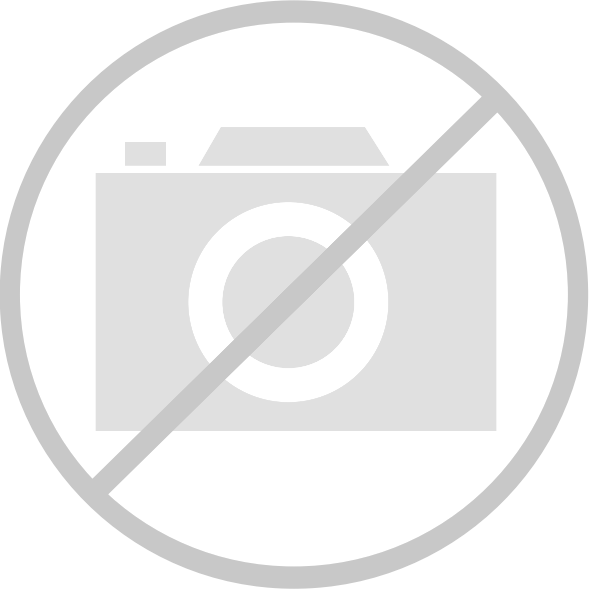
Nuvoton Technology Corporation - W78I054DDG Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, UART/USART
Peripherals	POR, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w78i054ddg

Pin Description, continued

SYMBOL	TYPE	DESCRIPTIONS
P3.0–P3.7	I/O H	PORT 3: Port 3 is a bi-directional I/O port with internal pull-ups. All bits have alternate functions, which are described below: RXD (P3.0): Serial Port 0 input TXD (P3.1): Serial Port 0 output $\overline{\text{INT0}}$ (P3.2) : External Interrupt 0 $\overline{\text{INT1}}$ (P3.3) : External Interrupt 1 T0 (P3.4) : Timer 0 External Input T1 (P3.5) : Timer 1 External Input $\overline{\text{WR}}$ (P3.6) : External Data Memory Write Strobe $\overline{\text{RD}}$ (P3.7) : External Data Memory Read Strobe
P4.0–P4.3	I/O H	PORT 4: Another bit-addressable bidirectional I/O port P4. P4.3 and P4.2 are alternative function pins. It can be used as general I/O port or external interrupt input sources ($\overline{\text{INT2}}$ / $\overline{\text{INT3}}$).

* Note: TYPE I: input, O: output, I/O: bi-directional, H: pull-high, L: pull-low, D: open drain.

In application if MCU pins need external pull-up, it is recommended to add a pull-up resistor (10KΩ) between pin and power (V_{DD}) instead of directly wiring pin to V_{DD} for enhancing EMC.

6 BLOCK DIAGRAM

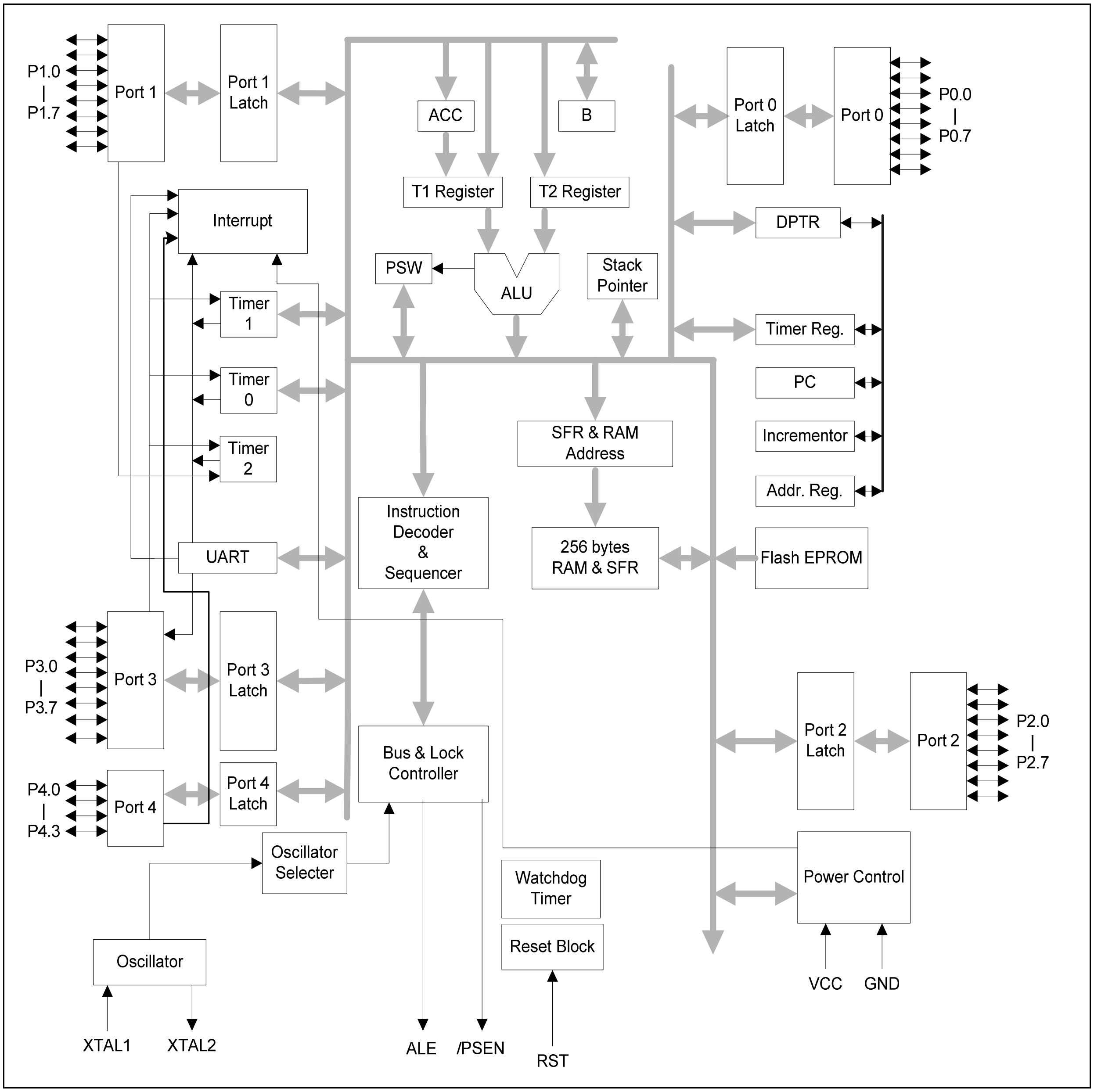


Figure 6- 1 W78I054D/W78I052D/W78I051D Block Diagram

8.2.2 Bit addressable Locations

The Scratch-pad RAM area from location 20h to 2Fh is byte as well as bit addressable. This means that a bit in this area can be individually addressed. In addition some of the SFRs are also bit addressable. The instruction decoder is able to distinguish a bit access from a byte access by the type of the instruction itself. In the SFR area, any existing SFR whose address ends in a 0 or 8 is bit addressable.

8.2.3 Stack

The scratch-pad RAM can be used for the stack. This area is selected by the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a jump, call or interrupt is invoked the return address is placed on the stack. There is no restriction as to where the stack can begin in the RAM. By default however, the Stack Pointer contains 07h at reset. The user can then change this to any value desired. The SP will point to the last used value. Therefore, the SP will be incremented and then address saved onto the stack. Conversely, while popping from the stack the contents will be read first, and then the SP is decreased.

Special Function Registers:

SYMBOL	DEFINITION	ADDRESS	MSB			BIT ADDRESS, SYMBOL				LSB	RESET
B	B register	F0H	(F7)	(F6)	(F5)	(F4)	(F3)	(F2)	(F1)	(F0)	0000 0000B
ACC	Accumulator	E0H	(E7)	(E6)	(E5)	(E4)	(E3)	(E2)	(E1)	(E0)	0000 0000B
P4	Port 4	D8H					INT2	INT3			0000 1111B
PSW	Program status word	D0H	(D7) CY	(D6) AC	(D5) F0	(D4) RS1	(D3) RS0	(D2) OV	(D1) F1	(D0) P	0000 0000B
TH2	T2 reg. high	CDH									0000 0000B
TL2	T2 reg. low	CCH									0000 0000B
RCAP2H	T2 capture low	CBH									0000 0000B
RCAP2L	T2 capture high	CAH									0000 0000B
T2MOD	Timer 2 Mode	C9								DCEN	0000 0000B
T2CON	Timer 2 control	C8H	(CF) TF2	(CE) EXF2	(CD) RCLK	(CC) TCLK	(CB) EXEN2	(CA) TR2	(C9) C/T2	(C8) CP/RL2	0000 0000B
SFRCN	SFR program of control	C7H			NOE	NCE	CTRL3	CTRL2	CTRL1	CTRL0	0000 0000B
SFRRD	SFR program of data register	C6H									0000 0000B
SFRAH	SFR program of address high byte	C5H									0000 0000B
SFRAL	SFR program of address low byte	C4H									0000 0000B
XICON	External interrupt control	C0H	PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2	0000 0000B
CHPCON	Chip control	BFH	SWRST	-		-	-	-	FBOOTSL	ENP	0000 0000B
EAPAGE	Erase page operation modes	BEH							EAPG1	EAPG0	0000 0000B
IP	Interrupt priority	B8H	(BF) -	(BE) -	(BD) PT2	(BC) PS	(BB) PT1	(BA) PX1	(B9) PT0	(B8) PX0	1100 0000B
IPH	Interrupt priority High	B7H									0000 0000B
P3	Port 3	B0H	(B7) RD	(B6) WR	(B5) T1	(B4) T0	(B3) INT1	(B2) INT0	(B1) TXD	(B0) RXD	1111 1111B
IE	Interrupt enable	A8H	(AF) EA	(AE) -	(AD) ET2	(AC) ES	(AB) ET1	(AA) EX1	(A9) ET0	(A8) EX0	0100 0000B
P2	Port 2	A0H	(A7) A15	(A6) A14	(A5) A13	(A4) A12	(A3) A11	(A2) A10	(A1) A9	(A0) A8	1111 1111B
SBUF	Serial buffer	99H									0000 0000B
SCON	Serial control	98H	(9F) SM0/FE	(9E) SM1	(9D) SM2	(9C) REN	(9B) TB8	(9A) RB8	(99) TI	(98) RI	0000 0000B
P1	Port 1	90H	(97)	(96)	(95)	(94)	(93)	(92)	(91) T2EX	(90) T2	1111 1111B
WDTC	Watchdog control	8FH	ENW	CLRW	WIDL	-	-	PS2	PS1	PS0	0000 0000B
AUXR	Auxiliary	8EH	-	-	-	-				ALEOFF	0000 0110B
TH1	Timer high 1	8DH									0000 0000B
TH0	Timer high 0	8CH									0000 0000B
TL1	Timer low 1	8BH									0000 0000B
TL0	Timer low 0	8AH									0000 0000B
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000 0000B
TCON	Timer control	88H	(8F) TF1	(8E) TR1	(8D) TF0	(8C) TR0	(8B) IE1	(8A) IT1	(89) IE0	(88) IT0	0000 0000B
PCON	Power control	87H	SMOD	SMOD0	-	POR	GF1	GF0	PD	IDL	0011 0000B
P0UPR	Port 0 pull up option Register	86H	-	-	-	-	-	-	-	P0UP	0000 0001B
DPH	Data pointer high	83H									0000 0000B

4	TR0	Timer 0 Run Control. This bit is set or cleared by software to turn timer/counter on or off.
3	IE1	Interrupt 1 Edge Detect Flag: Set by hardware when an edge/level is detected on $\overline{\text{INT1}}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin.
2	IT1	Interrupt 1 Type Control. Set/cleared by software to specify falling edge/ low level triggered external inputs.
1	IE0	Interrupt 0 Edge Detect Flag. Set by hardware when an edge/level is detected on $\overline{\text{INT0}}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the inverse of the pin.
0	IT0	Interrupt 0 Type Control: Set/cleared by software to specify falling edge/ low level triggered external inputs.

Timer Mode Control

Bit:	7	6	5	4	3	2	1	0
	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
	TIMER1				TIMER0			

Mnemonic: TMOD Address: 89h

BIT	NAME	FUNCTION
7	GATE	Gating control: When this bit is set, Timer/counter 1 is enabled only while the $\overline{\text{INT1}}$ pin is high and the TR1 control bit is set. When cleared, the $\overline{\text{INT1}}$ pin has no effect, and Timer 1 is enabled whenever TR1 control bit is set.
6	$C/\overline{T}$	Timer or Counter Select: When clear, Timer 1 is incremented by the internal clock. When set, the timer counts falling edges on the T1 pin.
5	M1	Timer 1 mode select bit 1. See table below.
4	M0	Timer 1 mode select bit 0. See table below.
3	GATE	Gating control: When this bit is set, Timer/counter 0 is enabled only while the $\overline{\text{INT0}}$ pin is high and the TR0 control bit is set. When cleared, the $\overline{\text{INT0}}$ pin has no effect, and Timer 0 is enabled whenever TR0 control bit is set.
2	$C/\overline{T}$	Timer or Counter Select: When clear, Timer 0 is incremented by the internal clock. When set, the timer counts falling edges on the T0 pin.
1	M1	Timer 0 mode select bit 1. See table below.
0	M0	Timer 0 mode select bit 0. See table below.

M1, M0: Mode Select bits:

M1	M0	MODE
0	0	Mode 0: 13-bit timer/counter TLx serves as 5-bit pre-scale.
0	1	Mode 1: 16-bit timer/counter, no pre-scale.
1	0	Mode 2: 8-bit timer/counter with auto-reload from THx.
1	1	Mode 3: (Timer 0) TL0 is an 8-bit timer/counter controlled by the standard Timer0 control bits. TH0 is an 8-bit timer only controlled by Timer1 control bits. (Timer 1)

BIT	NAME	FUNCTION
7~0	SBUF	Serial data on the serial port is read from or written to this location. It actually consists of two separate internal 8-bit registers. One is the receive resister, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write access is to the transmit data buffer.

Port 2

Bit:	7	6	5	4	3	2	1	0
	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0

Mnemonic: P2 Address: A0h

BIT	NAME	FUNCTION
7-0	P2.[7:0]	Port 2 is a bi-directional I/O port with internal pull-ups. This port also provides the upper address bits for accesses to external memory.

Interrupt Enable

Bit:	7	6	5	4	3	2	1	0
	EA	-	ET2	ES	ET1	EX1	ET0	EX0

Mnemonic: IE Address: A8h

BIT	NAME	FUNCTION
7	EA	Global enable. Enable/Disable all interrupts.
6	-	Reserved
5	ET2	Enable Timer 2 interrupt.
4	ES	Enable Serial Port 0 interrupt.
3	ET1	Enable Timer 1 interrupt.
2	EX1	Enable external interrupt 1.
1	ET0	Enable Timer 0 interrupt.
0	EX0	Enable external interrupt 0.

Port 3

Bit:	7	6	5	4	3	2	1	0
	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Mnemonic: P3 Address: B0h

P3.7-0: General purpose Input/Output port. Most instructions will read the port pins in case of a port read access, however in case of read-modify-write instructions, the port latch is read. These alternate functions are described below:

BIT	NAME	FUNCTION
7	P3.7	$\overline{\text{RD}}$

EAPAGE ERASE PAGE Operation Modes

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	EAPG1	EAPG0

Mnemonic: EAPAGE   Address: BD

BIT	NAME	FUNCTION
1	EAPG1	1.To ease PAGE1 when ease command is set.(LD flash)
0	EAPG0	1.To ease PAGE0 when ease command is set. (AP Flash)

```
;CPU Clock = 12MHz/12T mode
READ_TIME           EQU     1
PROGRAM_TIME        EQU     50
ERASE_TIME          EQU     5000
```

Erase_AP Flash:

```
    mov     EAPAGE,#01h         ;set EAPAGE is APROM
    mov     SFRCN,#ERASE_ROM
    mov     TL0,#LOW (65536-ERASE_TIME)
    mov     TH0,#HIGH(65536-ERASE_TIME)
    setb    TR0
    mov     CHPCON,#00000011b
    mov     EAPAGE,#00h         ;clear EAPAGE
    clr     TF0
    clr     TR0
    ret
```

Erase_LD Flash:

```
    mov     EAPAGE,#02h         ;set EAPAGE is LDROM
    mov     SFRCN,#ERASE_ROM
    mov     TL0,#LOW (65536-ERASE_TIME)
    mov     TH0,#HIGH(65536-ERASE_TIME)
    setb    TR0
    mov     CHPCON,#00000011b
    mov     EAPAGE,#00h         ;clear EAPAGE
    clr     TF0
    clr     TR0
    ret
```

Chip Control

Bit:	7	6	5	4	3	2	1	0

Op-code	HEX Code	Bytes	W78I054D/W78I052D/W78I051D series Clock cycles
SUBB A, R3	9B	1	12
SUBB A, R4	9C	1	12
SUBB A, R5	9D	1	12
SUBB A, R6	9E	1	12
SUBB A, R7	9F	1	12
SUBB A, @R0	96	1	12
SUBB A, @R1	97	1	12
SUBB A, direct	95	2	12
SUBB A, #data	94	2	12
INC A	04	1	12
INC R0	08	1	12
INC R1	09	1	12
INC R2	0A	1	12
INC R3	0B	1	12
INC R4	0C	1	12
INC R5	0D	1	12
INC R6	0E	1	12
INC R7	0F	1	12
INC @R0	06	1	12
INC @R1	07	1	12
INC direct	05	2	12
INC DPTR	A3	1	24
DEC A	14	1	12
DEC R0	18	1	12
DEC R1	19	1	12
DEC R2	1A	1	12
DEC R3	1B	1	12
DEC R4	1C	1	12
DEC R5	1D	1	12
DEC R6	1E	1	12
DEC R7	1F	1	12
DEC @R0	16	1	12

Op-code	HEX Code	Bytes	W78I054D/W78I052D/W78I051D series Clock cycles
DEC @R1	17	1	12
DEC direct	15	2	12
MUL AB	A4	1	48
DIV AB	84	1	48
DA A	D4	1	12
ANL A, R0	58	1	12
ANL A, R1	59	1	12
ANL A, R2	5A	1	12
ANL A, R3	5B	1	12
ANL A, R4	5C	1	12
ANL A, R5	5D	1	12
ANL A, R6	5E	1	12
ANL A, R7	5F	1	12
ANL A, @R0	56	1	12
ANL A, @R1	57	1	12
ANL A, direct	55	2	12
ANL A, #data	54	2	12
ANL direct, A	52	2	12
ANL direct, #data	53	3	24
ORL A, R0	48	1	12
ORL A, R1	49	1	12
ORL A, R2	4A	1	12
ORL A, R3	4B	1	12
ORL A, R4	4C	1	12
ORL A, R5	4D	1	12
ORL A, R6	4E	1	12
ORL A, R7	4F	1	12
ORL A, @R0	46	1	12
ORL A, @R1	47	1	12
ORL A, direct	45	2	12
ORL A, #data	44	2	12
ORL direct, A	42	2	12

Op-code	HEX Code	Bytes	W78I054D/W78I052D/W78I051D series Clock cycles
MOV A, direct	E5	2	12
MOV A, #data	74	2	12
MOV R0, A	F8	1	12
MOV R1, A	F9	1	12
MOV R2, A	FA	1	12
MOV R3, A	FB	1	12
MOV R4, A	FC	1	12
MOV R5, A	FD	1	12
MOV R6, A	FE	1	12
MOV R7, A	FF	1	12
MOV R0, direct	A8	2	24
MOV R1, direct	A9	2	24
MOV R2, direct	AA	2	24
MOV R3, direct	AB	2	24
MOV R4, direct	AC	2	24
MOV R5, direct	AD	2	24
MOV R6, direct	AE	2	24
MOV R7, direct	AF	2	24
MOV R0, #data	78	2	12
MOV R1, #data	79	2	12
MOV R2, #data	7A	2	12
MOV R3, #data	7B	2	12
MOV R4, #data	7C	2	12
MOV R5, #data	7D	2	12
MOV R6, #data	7E	2	12
MOV R7, #data	7F	2	12
MOV @R0, A	F6	1	12
MOV @R1, A	F7	1	12
MOV @R0, direct	A6	2	24
MOV @R1, direct	A7	2	24
MOV @R0, #data	76	2	12
MOV @R1, #data	77	2	12

Op-code	HEX Code	Bytes	W78I054D/W78I052D/W78I051D series Clock cycles
MOV direct, A	F5	2	12
MOV direct, R0	88	2	24
MOV direct, R1	89	2	24
MOV direct, R2	8A	2	24
MOV direct, R3	8B	2	24
MOV direct, R4	8C	2	24
MOV direct, R5	8D	2	24
MOV direct, R6	8E	2	24
MOV direct, R7	8F	2	24
MOV direct, @R0	86	2	24
MOV direct, @R1	87	2	24
MOV direct, direct	85	3	24
MOV direct, #data	75	3	24
MOV DPTR, #data 16	90	3	24
MOVC A, @A+DPTR	93	1	24
MOVC A, @A+PC	83	1	24
MOVX A, @R0	E2	1	24
MOVX A, @R1	E3	1	24
MOVX A, @DPTR	E0	1	24
MOVX @R0, A	F2	1	24
MOVX @R1, A	F3	1	24
MOVX @DPTR, A	F0	1	24
PUSH direct	C0	2	24
POP direct	D0	2	24
XCH A, R0	C8	1	12
XCH A, R1	C9	1	12
XCH A, R2	CA	1	12
XCH A, R3	CB	1	12
XCH A, R4	CC	1	12
XCH A, R5	CD	1	12
XCH A, R6	CE	1	12
XCH A, R7	CF	1	12

Op-code	HEX Code	Bytes	W78I054D/W78I052D/W78I051D series Clock cycles
XCH A, @R0	C6	1	12
XCH A, @R1	C7	1	12
XCHD A, @R0	D6	1	12
XCHD A, @R1	D7	1	12
XCH A, direct	C5	2	24
CLR C	C3	1	12
CLR bit	C2	2	12
SETB C	D3	1	12
SETB bit	D2	2	12
CPL C	B3	1	12
CPL bit	B2	2	12
ANL C, bit	82	2	24
ANL C, /bit	B0	2	24
ORL C, bit	72	2	24
ORL C, /bit	A0	2	24
MOV C, bit	A2	2	12
MOV bit, C	92	2	24
ACALL addr11	71, 91, B1, 11, 31, 51, D1, F1	2	24
LCALL addr16	12	3	24
RET	22	1	24
RETI	32	1	24
AJMP ADDR11	01, 21, 41, 61, 81, A1, C1, E1	2	24
LJMP addr16	02	3	24
JMP @A+DPTR	73	1	24
SJMP rel	80	2	24
JZ rel	60	2	24
JNZ rel	70	2	24
JC rel	40	2	24
JNC rel	50	2	24

16 SERIAL PORT

Serial port in this device is a full duplex port. The serial port is capable of synchronous as well as asynchronous communication. In Synchronous mode the device 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 serial port can operate in four different modes as described below.

16.1 MODE 0

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 device whether it 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 of the oscillator frequency. This Baud Rate is determined by the SM2 bit (SCON.5). When this bit is set to 0, then the serial port runs at 1/12 of the clock. This additional facility of programmable baud rate in mode 0 is the only difference between the standard 8051 and W78I054D/W78I052D/W78I051D.

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 this device and the device at the other end of the line. Any instruction that causes a write to SBUF will start the transmission. The shift clock will be activated and data will be shifted out on the RxD pin till all 8 bits are transmitted. If SM2 = 1, then the data on RxD will appear 1 clock period before the falling edge of shift clock on TxD. The clock on TxD then remains low for 2 clock periods, and then goes high again. If SM2 = 0, the data on RxD will appear 3 clock periods before the falling edge of shift clock on TxD. The clock on TxD then remains low for 6 clock periods, and then goes high again. This ensures that at the receiving end the data on RxD line can either be clocked on the rising edge of the shift clock on TxD or latched when the TxD clock is low.

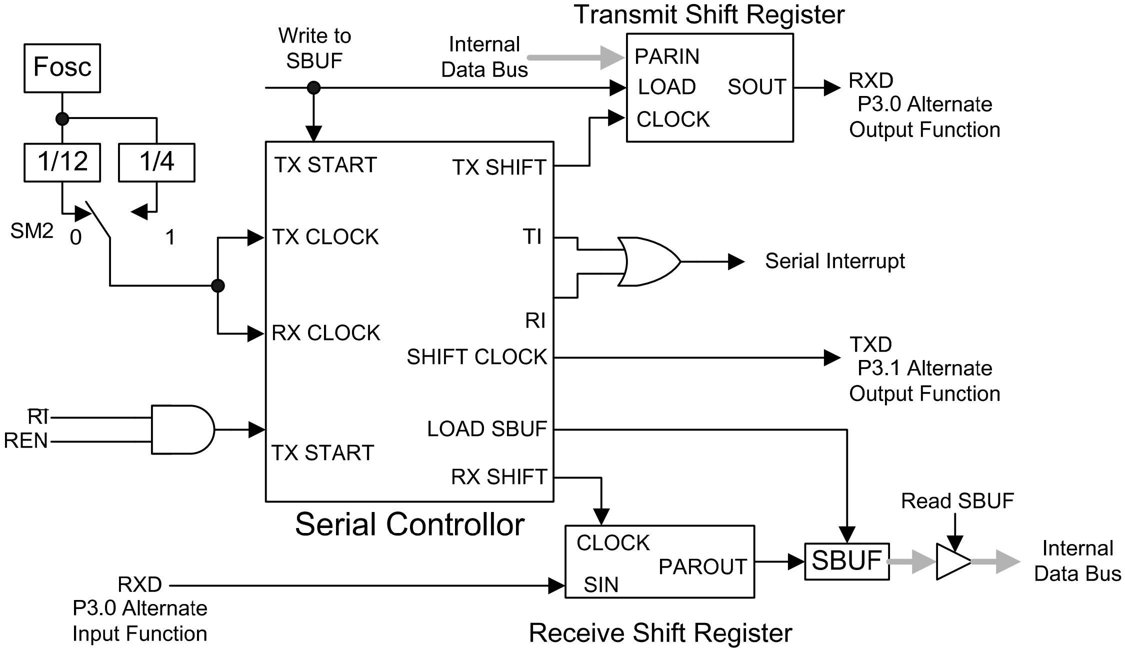


Figure 16- 1 Serial port mode 0

The TI flag is set high in S6P2 following the end of transmission of the last bit. The serial port will receive data when REN is 1 and RI is zero. The shift clock (TxD) will be activated and the serial port will latch data on the rising edge of shift clock. The external device should therefore present data on the falling edge on the shift clock. This process continues till all the 8 bits have been received. The RI flag is set in S6P2 following the last rising edge of the shift clock on TxD. This will stop reception, till the RI is cleared by software.

16.2 MODE 1

In Mode 1, the full duplex asynchronous mode is used. Serial communication frames are made up of 10 bits transmitted on TXD and received on RXD. The 10 bits consist of a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in the SFR SCON. The baud rate in this mode is variable. The serial baud can be programmed to be 1/16 or 1/32 of the Timer 1 overflow. Since the Timer 1 can be set to different reload values, a wide variation in baud rates is possible.

Transmission begins with a write to SBUF. The serial data is brought out on to TxD pin at S6P2 following the first roll-over of divide by 16 counter. The next bit is placed on TxD pin at S6P2 following the next rollover of the divide by 16 counter. Thus the transmission is synchronized to the divide by 16 counter and not directly to the write to SBUF signal. After all 8 bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the S6P2 state after the stop bit has been put out on TxD pin. This will be at the 10th rollover of the divide by 16 counters after a write to SBUF.

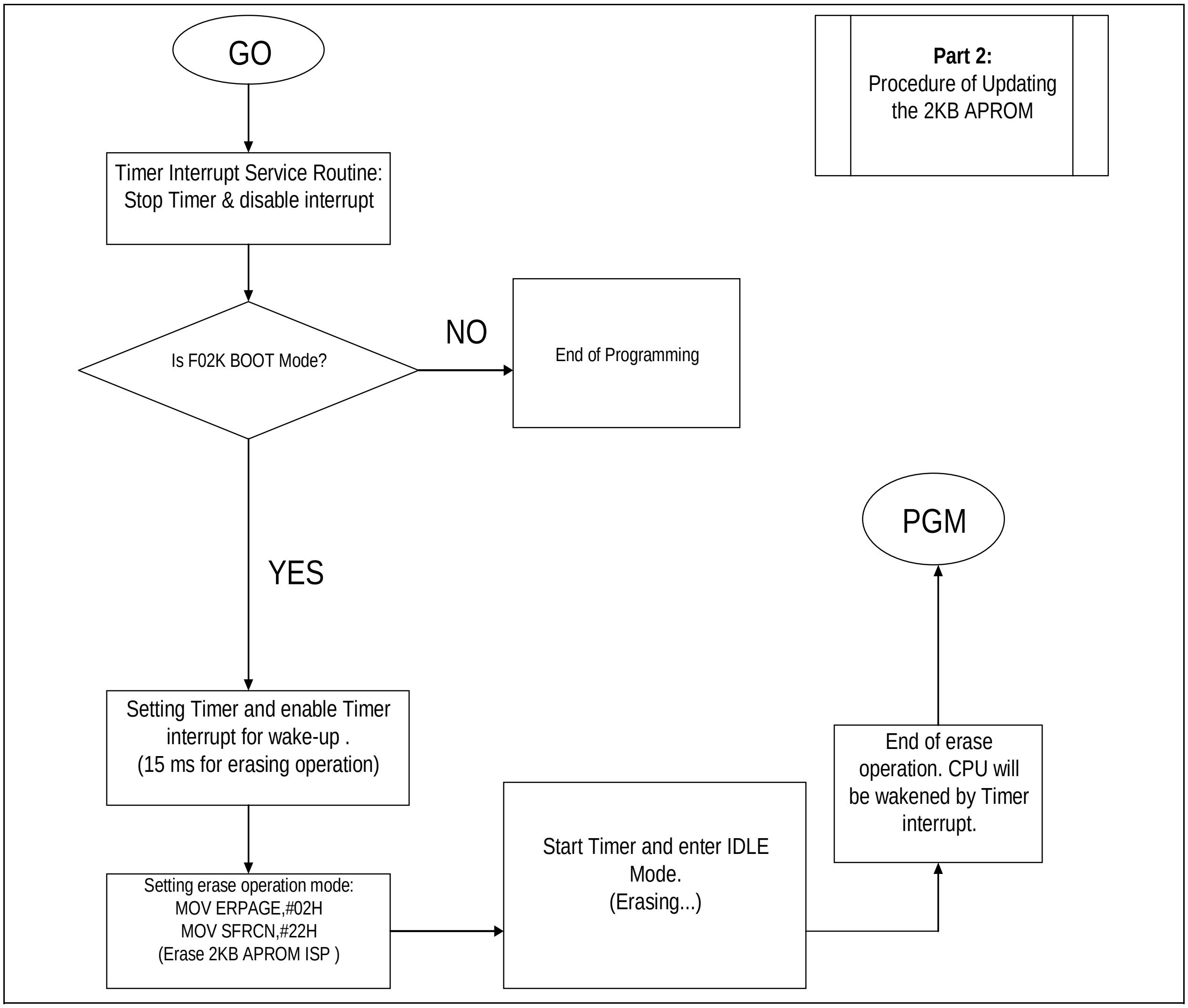
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. The 1-to-0 detector continuously monitors the RxD line, sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide by 16 counters is immediately reset. This helps to align the bit boundaries with the rollovers of the divide by 16 counters.

17 FLASH ROM CODE BOOT MODE SELECTION

The W78I054D/W78I052D/W78I051D boots from APROM program (16K/8K/4K bytes) or LDROM program (2K bytes) at power on reset or external reset.

BOOT MODE Select by CONFIG bits

CBS (CONFIG.2)	Config boot select at Power-on reset and external reset. 1: Boot from APROM (0x0000). 0: Boot from LDROM (0x3800).


GO
Part 2:
Procedure of Updating the 2KB APROM
Timer Interrupt Service Routine: Stop Timer & disable interrupt
Is F02K BOOT Mode?
NO
End of Programming
PGM
YES
Setting Timer and enable Timer interrupt for wake-up . (15 ms for erasing operation)
End of erase operation. CPU will be wakened by Timer interrupt.
Start Timer and enter IDLE Mode. (Erasing...)
Setting erase operation mode: MOV ERPAGE,#02H MOV SFRCN,#22H (Erase 2KB APROM ISP)

Bit 0: Lock bits

0: Lock enable

1: Lock disable

This bit is used to protect the customer's program code in the W78I054D/W78I052D/W78I051D. It may be set after the programmer finishes the programming and verifies sequence. Once these bits are set to logic 0, both the FLASH data and Special Setting Registers can not be accessed again.

Bit 1:MOVC inhibit

0: MOVC inhibit enable

1: MOVC inhibit disable

This bit is used to restrict the accessible region of the MOVC instruction. It can prevent the MOVC instruction in external program memory from reading the internal program code. When this bit is set to logic 0, a MOVC instruction in external program memory space will be able to access code only in the external memory, not in the internal memory. A MOVC instruction in internal program memory space will always be able to access the ROM data in both internal and external memory. If this bit is logic 1, there are no restrictions on the MOVC instruction.

Bit2: CBS

Config boot select at Power-on reset and external reset.

CBS=1: Boot from AP Flash block (default).

CBS=0: Boot from LD Flash block (0x3800).

Bit 3: NSR (Noise Sensitivity Reduction)

NSR=1: Noise Sensitivity Reduction is disabled.

NSR=0: Noise Sensitivity Reduction is enabled.

Bit 4: Must be “1”

Bit 5: Machine Cycle Select

This bit is select MCU core, default value is logic 1, the MCU core is 12T per instruction. Once this bit is set to logic 0, the MCU core is 6T per instruction.

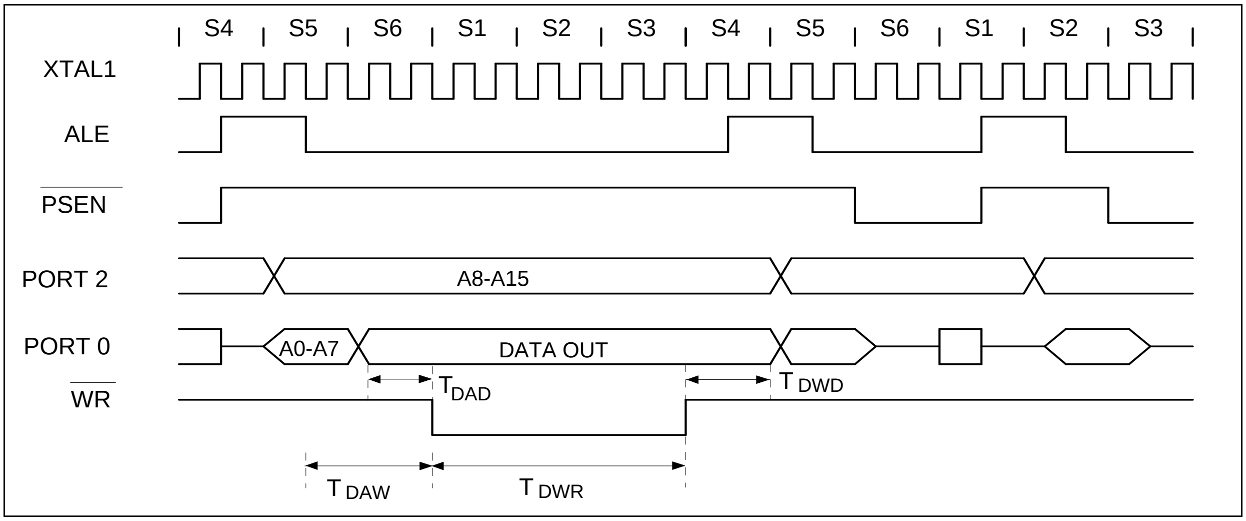
Bit 6: Must be “1”

Bit 7: Crystal Select

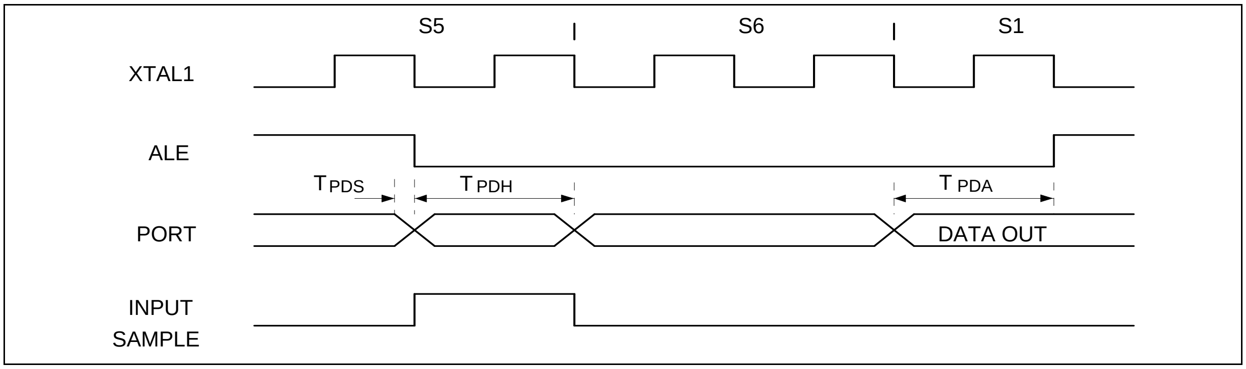
0 (24MHz): If system clock is slower than 24MHz, programming “0”. It can reduce EMI effect and save the power consumption.

1 (40MHz): If system clock is faster than 24MHz, programming “1”.

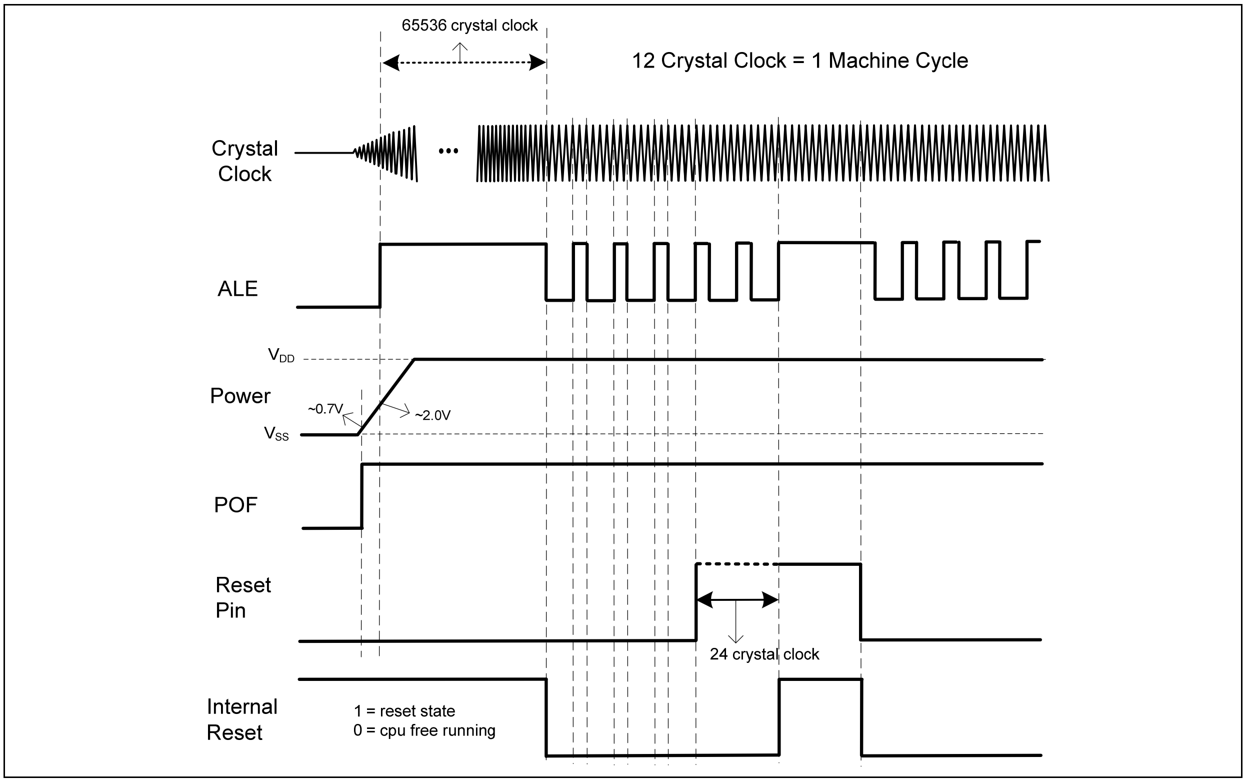
21.4.3 Data Write Cycle



21.4.4 Port Access Cycle



21.4.5 Reset Pin Access Cycle



```
    mov     TL0,#LOW (65536-ERASE_TIME)
    mov     TH0,#HIGH(65536-ERASE_TIME)
    setb    TR0
    mov     CHPCON,#00000011b
    mov     EAPAGE,#00h          ;clear EAPAGE
    clr     TF0
    clr     TR0
    ret
;************************************************************************
; * VERIFY APROM BANK
;************************************************************************
Erase_Verify_ROM:
    mov     SFRCN,#ERASE_VERIFY
    mov     DPTR,#0000h
er_lp:
    mov     TL0,#LOW (65536-READ_TIME)
    mov     TH0,#HIGH(65536-READ_TIME)
    mov     SFRAL,DPL
    mov     SFRAH,DPH
    setb    TR0
    mov     CHPCON,#00000011b
    clr     TF0
    clr     TR0
    mov     A,SFRFD
    cjne    A,#0FFh,Erase_Verify_Error
    inc     DPTR
    mov     R0,DPL
    cjne    R0,#LOW (APROM_END_ADDRESS),er_lp
    mov     R1,DPH
    cjne    R1,#HIGH(APROM_END_ADDRESS),er_lp
    ret
Erase_Verify_Error:
    mov     P1,#02h
    sjmp    $
;*******************************************************************************
;*PROGRAMMING APROM BANK, APROM write 55h,AAh,55h,AAh........
;*******************************************************************************
Program_APROM:
```