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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	24MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n79e352rafg

1. GENERAL DESCRIPTION

The N79E352(R) is an 8-bit Turbo 51 microcontroller which has Flash EPROM programmable hardware writer. The instruction set of the N79E352(R) is fully compatible with the standard 8052. The N79E352(R) contains a 8Kbytes of main Flash EPROM; a 256 bytes of RAM; 128 bytes NVM Data Flash EPROM; three 16-bit timer/counters; 2-channel 8-bit PWM; 1-channel UART and 1 additional input capture. These peripherals are supported by 11 interrupt sources four-level interrupt capability. To facilitate programming and verification, the Flash EPROM inside the N79E352(R) allows the program memory to be programmed and read electronically. Once the code is confirmed, the user can protect the code for security. N79E352(R) is designed for cost effective applications which can serve industrial devices, and other low power applications.

7. MEMORY ORGANIZATION

N79E352(R) separates the memory into two separate sections, the Program Memory and the Data Memory. The Program Memory is used to store the instruction op-codes, while the Data Memory is used to store data or for memory mapped devices.

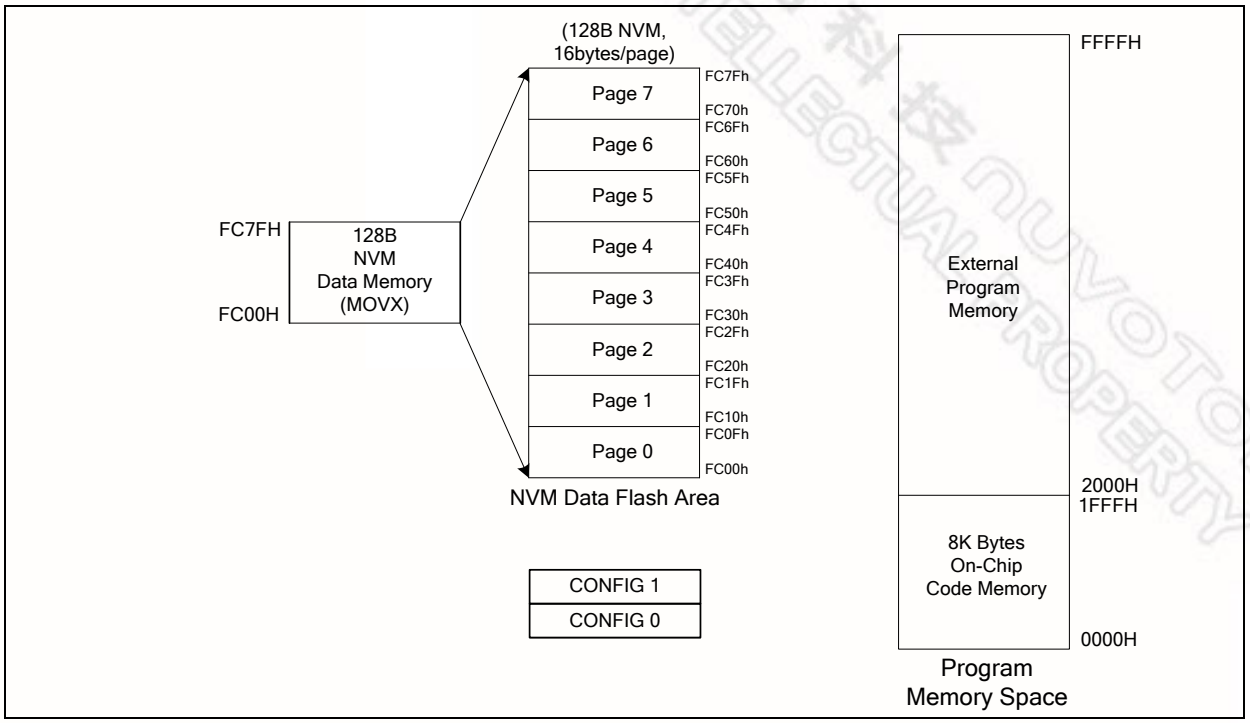


Figure 7-1: N79E352(R) Memory Map

7.1 Program Memory (on-chip Flash)

The Program Memory on N79E352(R) can be up to 8K bytes long. All instructions are fetched for execution from this memory area. The MOVC instruction can also access this memory region.

7.2 Data Memory

The N79E352(R) has NVM data memory of 128 bytes for customer's data store used. The NVM data memory has 8 pages area and each page has 16 bytes. The N79E352(R) can access up to 64Kbytes of external Data Memory. This memory region is accessed by the MOVX instructions. For NVM s/w read access, user require to set EnNVM bit, otherwise, the access will goes to external data memory. N79E352(R) has the standard 256 bytes of on-chip Scratchpad RAM. This can be accessed either by direct addressing or by indirect addressing. There are also some Special Function Registers (SFRs), which can only be accessed by direct addressing. Since the Scratchpad RAM is only 256 bytes, it can be used only when data contents are small.

7.3 Scratch-pad RAM and Register Map

As mentioned before, N79E352(R) has separate Program and Data Memory areas. The on-chip 256 bytes scratch pad RAM is in addition to the external memory. There are also several Special Function Registers (SFRs) which can be accessed by software. The SFRs can be accessed only by direct addressing, while the on-chip RAM can be accessed by either direct or indirect addressing.

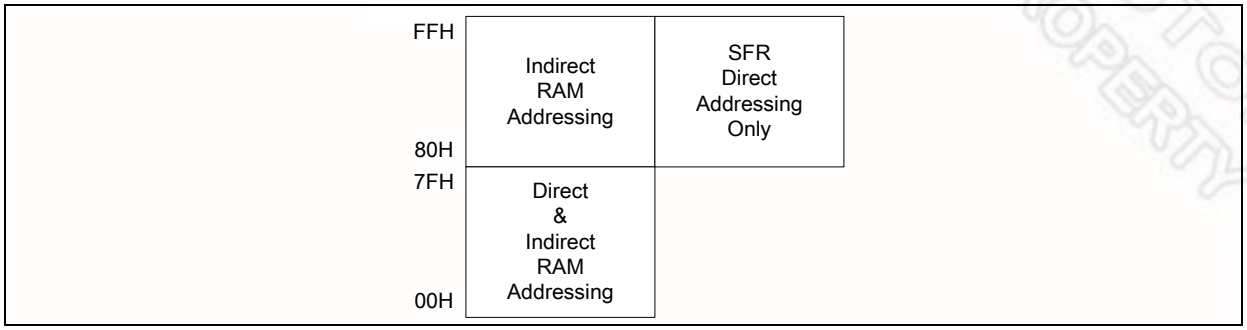


Figure 7-2: N79E352(R) RAM and SFR Memory Map

Since the scratch-pad RAM is only 256 bytes it can be used only when data contents are small. There are several other special purpose areas within the scratch-pad RAM. These are illustrated in next figure.

SYMBOL	DEFINITION	ADDRESS	MSB			BIT ADDRESS, SYMBOL				LSB	RESET
IP1	INTERRUPT PRIORITY 1	F8H	PCAP	PBO	-	PWDI	-	-	PKB	PI2	00x0 xx00B
IP1H	INTERRUPT HIGH PRIORITY 1	F7H	PCAPH	PBOH	-	PWDIH	-	-	PKBH	PI2H	00x0 xx00B
B	B REGISTER	F0H	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0000 0000B
P5M2	PORT 5 OUTPUT MODE 2	EEH	-	-	-	-	-	-	P5M2.1	P5M2.0	CONFIG0.PMODE=1; Xxxx xx00B CONFIG0.PMODE=0; Xxxx xx11B
P5M1	PORT 5 OUTPUT MODE 1	EDH	-	-	-	-	-	ENCLK	P5M1.1	P5M1.0	CONFIG0.PMODE=1; Xxxx x000B CONFIG0.PMODE=0; Xxxx x011B
PORTS	PORT SHMITT REGISTER	ECH	-	-	P5S	-	P3S	P2S	P1S	P0S	xx0x 0000B
KBL	KEYBOARD LEVEL REGISTER	E9H	KBL.7	KBL.6	KBL.5	KBL.4	KBL.3	KBL.2	KBL.1	KBL.0	0000 0000B
EIE	INTERRUPT ENABLE 1	E8H	ECPTF	EBO	-	EWDI	-	-	EKB	EI2	00x0 xx00B
CCH0	INPUT CAPTURE 0 HIGH	E5H	CCH0.7	CCH0.6	CCH0.5	CCH0.4	CCH0.3	CCH0.2	CCH0.1	CCH0.0	0000 0000B
CCL0	INPUT CAPTURE 0 LOW	E4H	CCL0.7	CCL0.6	CCL0.5	CCL0.4	CCL0.3	CCL0.2	CCL0.1	CCL0.0	0000 0000B
ACC	ACCUMULATOR	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000B
PWMCON1	PWM CONTROL REGISTER 1	DCH	PWMRUN	-	-	CLRPWM	-	-	-	-	0xx0 xxxxB
PWM1L	PWM 1 LOW BITS REGISTER	DBH	PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0	0000 0000B
PWM0L	PWM 0 LOW BITS REGISTER	DAH	PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0	0000 0000B
WDCON	WATCH-DOG CONTROL	D8H	WDRUN	POR	-	-	WDIF	WTRF	EWRST	WDCLR	POR: X1xx 0000B External reset: Xxxx 0xx0B Watchdog reset: Xxxx 01x0B
PWMCON3	PWM CONTROL REGISTER 3	D7H	-	-	PWM1OE	PWM0OE	PCLK.1	PCLK.0	FP1	FP0	Xx00 0000B
PSW	PROGRAM STATUS WORD	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	0000 0000B
NVMDATA	NVM DATA	CFH	NVMDATA.7	NVMDATA.6	NVMDATA.5	NVMDATA.4	NVMDATA.3	NVMDATA.2	NVMDATA.1	NVMDATA.0	0000 0000B
NVMCON	NVM CONTROL	CEH	EER	EWR	EnNVM	-	-	-	-	-	000x xxxxB
TH2	TIMER 2 MSB	CDH	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0	0000 0000B
TL2	TIMER 2 LSB	CCH	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0	0000 0000B
RCAP2H	TIMER 2 RELOAD MSB	CBH	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0	0000 0000B
RCAP2L	TIMER 2 RELOAD LSB	CAH	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0	0000 0000B
T2MOD	TIMER 2 MODE	C9H	-	-	-	ICEN0	T2CR	1	T2OE	DCEN	Xxx0 0100B
T2CON	TIMER 2 CONTROL	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL	0000 0000B
TA	TIMED ACCESS PROTECTION	C7H	TA.7	TA.6	TA.5	TA.4	TA.3	TA.2	TA.1	TA.0	0000 0000B
NVMADDR	NVM LOW BYTE ADDRESS	C6H	NVMADDR.7	NVMADDR.6	NVMADDR.5	NVMADDR.4	NVMADDR.3	NVMADDR.2	NVMADDR.1	NVMADDR.0	0000 0000B
STATUS	STATUS REGISTER	C5H	-	-	-	-	-	-	SPTA0	SPRA0	Xxxx xx00B
PMR	POWER MANAGEMENT REGISTER	C4H	CD1	CD0	SWB	-	-	ALE-OFF	-	-	010x xxxxB
ROMMAP	ROMMAP REGISTER	C2H	WS	1	-	-	-	1	1	0	01xxx110B
I2ADDR	I2C ADDRESS1	C1H	ADDR.7	ADDR.6	ADDR.5	ADDR.4	ADDR.3	ADDR.2	ADDR.1	GC	xxxxxxx0B

SYMBOL	DEFINITION	ADDRESS	MSB	BIT ADDRESS, SYMBOL						LSB	RESET
I2CON	I2C CONTROL REGISTER	C0H	-	ENSI	STA	STO	SI	AA	-	-	x00000xxB
I2TIMER	I2C TIMER COUNTER REGISTER	BFH	-	-	-	-	-	ENTI	DIV4	TIF	Xxxx x000B
I2CLK	I2C CLOCK RATE	BEH	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0	0000 0000B
I2STATUS	I2C STATUS	BDH	I2STATUS.7	I2STATUS.6	I2STATUS.5	I2STATUS.4	I2STATUS.3	I2STATUS.2	I2STATUS.1	I2STATUS.0	1111 1000B
I2DAT	I2C DATA	BCH	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0	xxxxxxxxB
SADEN	SLAVE ADDRESS MASK	B9H	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0	00000000B
IP0	INTERRUPT PRIORITY	B8H	-	-	PT2	PS	PT1	PX1	PT0	PX0	Xx00 0000B
IP0H	INTERRUPT HIGH PRIORITY	B7H	-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	Xx00 0000B
P2M2	PORT 2 OUTPUT MODE 2	B6H	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	CONFIG0.PMODE=1; 0000 0000B CONFIG0.PMODE=0; 1111 1111B
P2M1	PORT 2 OUTPUT MODE 1	B5H	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	CONFIG0.PMODE=1; 0000 0000B CONFIG0.PMODE=0; 1111 1111B
P1M2	PORT 1 OUTPUT MODE 2	B4H	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	CONFIG0.PMODE=1; 0000 0000B CONFIG0.PMODE=0; 1111 1111B
P1M1	PORT 1 OUTPUT MODE 1	B3H	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	CONFIG0.PMODE=1; 0000 0000B CONFIG0.PMODE=0; 1111 1111B
P0M2	PORT 0 OUTPUT MODE 2	B2H	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	1111 1111B
P0M1	PORT 0 OUTPUT MODE 1	B1H	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	1111 1111B
P3	PORT3	B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	1111 1111B
			/RD	/WR	T1	T0	/INT1	/INT0	TXD	RXD	
SADDR	SLAVE ADDRESS	A9H	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0	0000 0000B
IE	INTERRUPT ENABLE	A8H	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0x00 0000B
P4	PORT4	A5H	-	-	-	-	P4.3	P4.2	P4.1	P4.0	Xxxx 1111B
CAPCON1	CAPTURE CONTROL 1	A4H	0	T0CC	-	-	ENF0	-	-	CPTF0	00xx 0xx0B
CAPCON0	CAPTURE CONTROL 0	A3H	-	-	-	-	CCT0.1	CCT0.0	-	-	Xxxx 00xxB
AUXR1	AUX FUNCTION REGISTER 1	A2H	KBF	BOD	BOI	LPBOV	SRST	BOV1	BOV0	BOS	0000 0000B
KBI	KEYBOARD INTERRUPT	A1H	KBI.7	KBI.6	KBI.5	KBI.4	KBI.3	KBI.2	KBI.1	KBI.0	0000 0000B
P2	PORT 2	A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	1111 1111B
			A15	A14	A13	A12	A11	A10	A9	A8	
P3M2	PORT 3 OUTPUT MODE 2	9FH	P3M2.7	P3M2.6	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	CONFIG0.PMODE=1; 0000 0000B CONFIG0.PMODE=0; 1111 1111B

BIT	NAME	FUNCTION
5	T2M	Timer 2 clock select: 0: Timer 2 uses a divide by 12 clocks. 1: Timer 2 uses a divide by 4 clocks.
4	T1M	Timer 1 clock select: 0: Timer 1 uses a divide by 12 clocks. 1: Timer 1 uses a divide by 4 clocks.
3	T0M	Timer 0 clock select: 0: Timer 0 uses a divide by 12 clocks. 1: Timer 0 uses a divide by 4 clocks.
2~0	MD2~0	Stretch MOVX select bits: These three bits are used to select the stretch value for the MOVX instruction. Using a variable MOVX length enables the user to access slower external memory devices or peripherals without the need for external circuits. The $\overline{RD}$ or $\overline{WR}$ strobe will be stretched by the selected interval. When accessing the on-chip SRAM, the MOVX instruction is always in 2 machine cycles regardless of the stretch setting. By default, the stretch has value of 1. If the user needs faster accessing, then a stretch value of 0 should be selected. MD2 MD1 MD0 Stretch value MOVX duration 0 0 0 0 2 machine cycles 0 0 1 1 3 machine cycles *(Default)* 0 1 0 2 4 machine cycles 0 1 1 3 5 machine cycles 1 0 0 4 6 machine cycles 1 0 1 5 7 machine cycles 1 1 0 6 8 machine cycles 1 1 1 7 9 machine cycles

PORT 1

Bit:	7	6	5	4	3	2	1	0
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Mnemonic: P1 Address: 90h

P1.7-0: 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	P1.7	Dedicated I/O pin.
6	P1.6	Dedicated I/O pin.
5	P1.5	PWM1 or I/O pin by alternative.
4	P1.4	PWM0 or I/O pin by alternative.

I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0

Mnemonic: I2CLK Address: BEh

BIT	NAME	FUNCTION
7-0	I2CLK.[7:0]	The I2C clock rate bits.

I2C TIMER COUNTER REGISTER

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	ENTI	DIV4	TIF

Mnemonic: I2TIMER Address: BFh

BIT	NAME	FUNCTION
7~3	-	Reserved.
2	ENTI	Enable I2C 14-bits Timer Counter: 0: Disable 14-bits Timer Counter count. 1: Enable 14-bits Timer Counter count. After enable ENTI and ENSI, the 14-bit counter will be counted. When SI flag of I2C is set, the counter will stop to count and 14-bits Timer Counter will be cleared.
1	DIV4	I2C Timer Counter clock source divide function: 0: The 14-bits Timer Counter source clock is F_{CPU} clock. 1: The 14-bits Timer Counter source clock is divided by 4.
0	TIF	The I2C Timer Counter count flag: 0: The 14-bits Timer Counter is not overflow. 1: The 14-bits Timer Counter is overflow. Before enable I2C Timer (both ENTI, ENSI = [1,1]) the SI must be cleared. If I2C interrupt is enabled. The I2C interrupt service routine will be executed. This bit is cleared by software.

I2C CONTROL REGISTER

Bit:	7	6	5	4	3	2	1	0
	-	ENSI	STA	STO	SI	AA	-	-

Mnemonic: I2CON Address: C0h

BIT	NAME	FUNCTION
7	-	Reserved.
6	ENSI	0: Disable I2C Serial Function. The SDA and SCL output are in a high impedance state. SDA and SCL input signals are ignored, I2C is not in the addressed slave mode or it is not addressable, and STO bit in I2CON is forced to “0”. No other bits are affected. P1.3 (SCL) and P1.2 (SDA) may be used as open drain I/O ports. 1: Enable I2C Serial Function. The P1.2 and P1.3 port latches must be to logic 1.
5	STA	START flag: 0: The STA bit is reset, no START condition or repeated START condition will

		be generated. 1: The STA bit is set to enter a master mode. The I2C hardware checks the status of I2C bus and generates a START condition if the bus is free. If bus is not free, then I2C waits for a STOP condition and generates a START condition after a delay. If STA is set while I2C is already in a master mode and one or more bytes are transmitted or received, I2C transmits a repeated START condition. STA may be set any time. STA may also be set when I2C interface is an addressed slave mode.
4	STO	The bit STO bit is set while I2C is in a master mode. A STOP condition is transmitted to the I2C bus. When the STOP condition is detected on the bus, the I2C hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from a bus error condition. In this case, no STOP condition is transmitted to the I2C bus. However, the I2C hardware behaves as if a STOP condition has been received and it switches to the not addressable slave receiver mode. The STO flag is automatically cleared by hardware. If the STA and STO bits are both set, then a STOP condition is transmitted to the I2C bus if I2C is in a master mode (in a slave mode, I2C generates an internal STOP condition which is not transmitted). I2C then transmits a START condition.
3	SI	0: When the SI flag is reset, no serial interrupt is requested, and there is no stretching on the serial clock on the SCL line. 1: When a new I2C bus state is present in the I2STATUS register, the SI flag is set by hardware, and, if the EA and ES bits (in IE register) are both set, a serial interrupt is requested when SI is set. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available. When SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be cleared by software.
2	AA	Assert Acknowledge Flag: 0: A not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when: 1) A data has been received while I2C is in the master receiver mode. 2) A data byte has been received while I2C is in the addressed slave receiver mode. 1: An acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when: 1) The own slave address has been received. 2) A data byte has been received while I2C is in the master receiver mode. 3) A data byte has been received while I2C is in the addressed slave receiver mode. 4) The General Call address has been received while the general call bit (GC) in I2ADDR is set.
1~0	-	Reserved.

I2C ADDRESS REGISTER

Bit:	7	6	5	4	3	2	1	0
	I2ADDR.7	I2ADDR.6	I2ADDR.5	I2ADDR.4	I2ADDR.3	I2ADDR.2	I2ADDR.1	GC

Mnemonic: I2ADDR Address: C1h

BIT	NAME	FUNCTION

7	6	5	4	3	2	1	0

Mnemonic: RCAP2L Address: CAh

BIT	NAME	FUNCTION
7-0	RCAP2L	Timer 2 Capture LSB: This register is used to capture the TL2 value when a timer 2 is configured in capture mode.RCAP2L is also used as the LSB of a 16-bit reload value when timer 2 is configured in auto-reload mode.

INPUT CAPTURE 0 HIGH REGISTER

Bit:	7	6	5	4	3	2	1	0
	CCH0.7	CCH0.6	CCH0.5	CCH0.4	CCH0.3	CCH0.2	CCH0.1	CCH0.0

Mnemonic: CCH0 Address: E4h

BIT	NAME	FUNCTION
7-0	CCH0	Capture 0 high byte.

IINTERRUPT ENABLE REGISTER 1

Bit:	7	6	5	4	3	2	1	0
	ECPTF	EBO	-	EWDI	-	-	EKB	EI2

Mnemonic: EIE Address: E8h

BIT	NAME	FUNCTION
7	ECPTF	0: Disable capture interrupt. 1: Enable capture interrupt.
6	EBO	Enable brownout interrupt. 0: Disable brownout interrupt. 1: Enable brownout interrupt.
5	-	Reserved.
4	EWDI	0: Disable Watchdog Timer Interrupt. 1: Enable Watchdog Timer Interrupt.
3~2	-	Reserved.
1	EKB	0: Disable Keypad Interrupt. 1: Enable Keypad Interrupt.
0	EI2	0: Disable I2C Interrupt. 1: Enable I2C Interrupt.

KEYBOARD LEVEL

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

Mnemonic: KBL Address: E9h

BIT	NAME	FUNCTION
7~0	KBL.7~0	Keyboard trigger level. 0: Low level trigger.x pin. 1: High level trigger on KBI.x pin. [x = 0-7]

PORTS SHMITT REGISTER

Bit:	7	6	5	4	3	2	1	0

Instruction Timing for N79E352(R), continued

Instruction	HEX Op-Code	Bytes	N79E352(R) Machine Cycles	N79E352(R) Clock Cycles	8052 Clock Cycles	N79E352(R) vs. 8052 Speed Ratio
ADDC A, R0	38	1	1	4	12	3
ADDC A, R1	39	1	1	4	12	3
ADDC A, R2	3A	1	1	4	12	3
ADDC A, R3	3B	1	1	4	12	3
ADDC A, R4	3C	1	1	4	12	3
ADDC A, R5	3D	1	1	4	12	3
ADDC A, R6	3E	1	1	4	12	3
ADDC A, R7	3F	1	1	4	12	3
ADDC A, @R0	36	1	1	4	12	3
ADDC A, @R1	37	1	1	4	12	3
ADDC A, direct	35	2	2	8	12	1.5
ADDC A, #data	34	2	2	8	12	1.5
ACALL addr11	71, 91, B1, 11, 31, 51, D1, F1	2	3	12	24	2
AJMP ADDR11	01, 21, 41, 61, 81, A1, C1, E1	2	3	12	24	2
ANL A, R0	58	1	1	4	12	3
ANL A, R1	59	1	1	4	12	3
ANL A, R2	5A	1	1	4	12	3
ANL A, R3	5B	1	1	4	12	3
ANL A, R4	5C	1	1	4	12	3
ANL A, R5	5D	1	1	4	12	3
ANL A, R6	5E	1	1	4	12	3
ANL A, R7	5F	1	1	4	12	3
ANL A, @R0	56	1	1	4	12	3
ANL A, @R1	57	1	1	4	12	3
ANL A, direct	55	2	2	8	12	1.5
ANL A, #data	54	2	2	8	12	1.5
ANL direct, A	52	2	2	8	12	1.5
ANL direct, #data	53	3	3	12	24	2
ANL C, bit	82	2	2	8	24	3
ANL C, /bit	B0	2	2	8	24	3
CJNE A, direct, rel	B5	3	4	16	24	1.5
CJNE A, #data, rel	B4	3	4	16	24	1.5
CJNE @R0, #data, rel	B6	3	4	16	24	1.5
CJNE @R1, #data, rel	B7	3	4	16	24	1.5
CJNE R0, #data, rel	B8	3	4	16	24	1.5
CJNE R1, #data, rel	B9	3	4	16	24	1.5
CJNE R2, #data, rel	BA	3	4	16	24	1.5
CJNE R3, #data, rel	BB	3	4	16	24	1.5
CJNE R4, #data, rel	BC	3	4	16	24	1.5
CJNE R5, #data, rel	BD	3	4	16	24	1.5
CJNE R6, #data, rel	BE	3	4	16	24	1.5

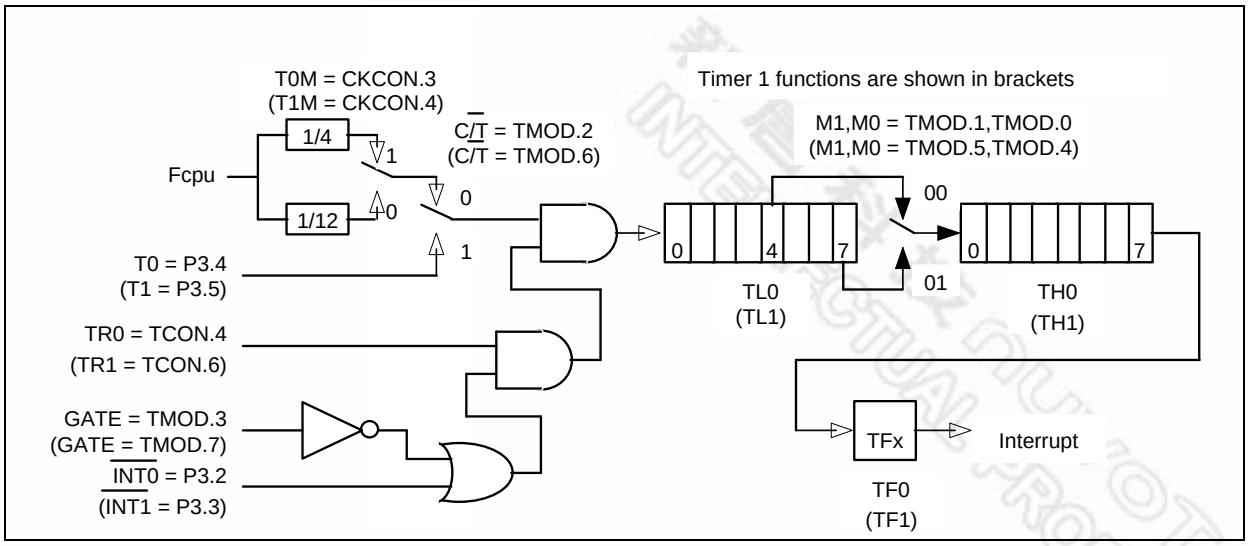


Figure 12-1: Timer/Counter Mode 0 & Mode 1

The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three basis. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port. If the first bit detected after the falling edge of RxD pin is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF.

After shifting in 8 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.

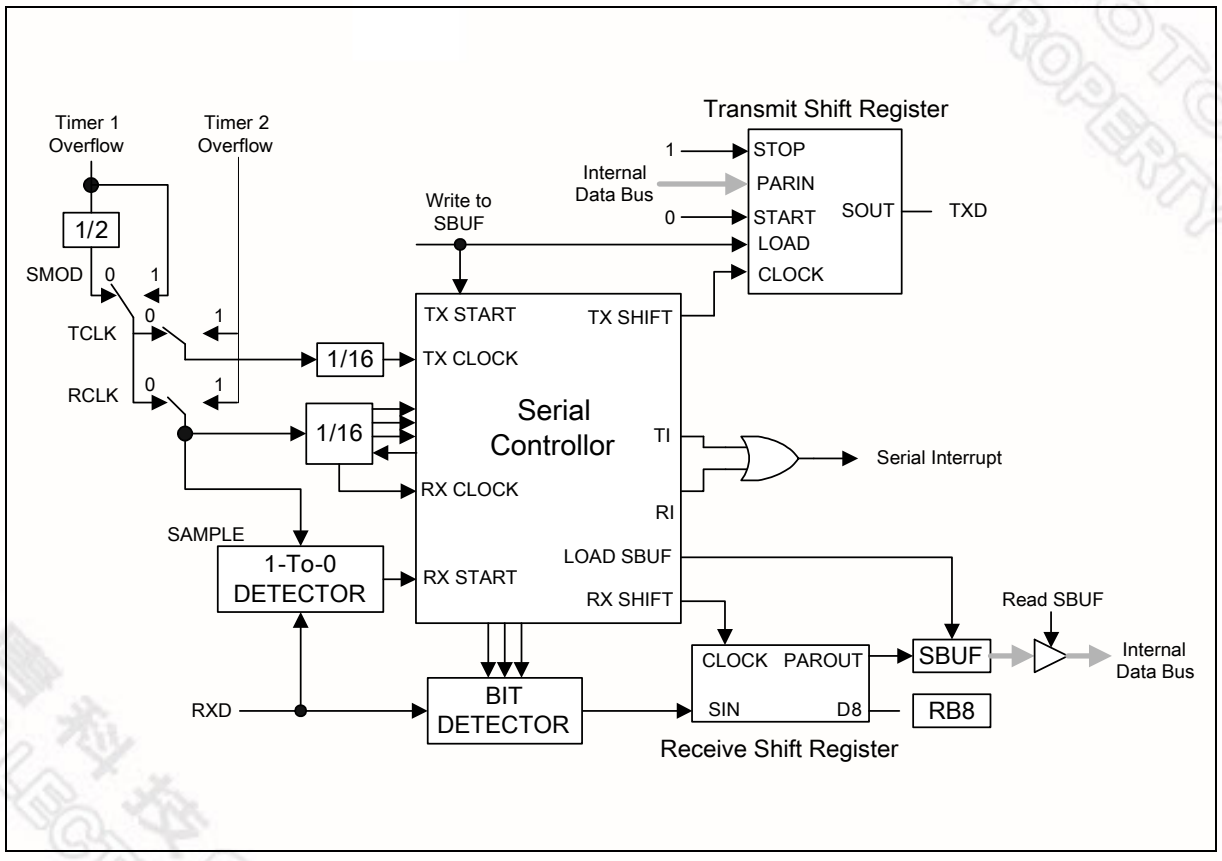


Figure 15-2: Uart Serial Port Mode 1

15.3 Mode 2

This mode uses a total of 11 bits in asynchronous full-duplex communication. The functional description is shown in the figure below. The frame consists of one start bit (0), 8 data bits (LSB first), a programmable 9th bit (TB8) and a stop bit (0). The 9th bit received is put into RB8. The baud rate is programmable to 1/32 or 1/64 of the oscillator frequency, which is determined by the SMOD bit in

PCON SFR. Transmission begins with a write to SBUF. The serial data is brought out on to TxD pin at C1 following the first roll-over of the divide by 16 counter. The next bit is placed on TxD pin at C1 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 9 bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the C1 state after the stop bit has been put out on TxD pin. This will be at the 11th rollover of the divide by 16 counter 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 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide by 16 counter. The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three basis. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port.

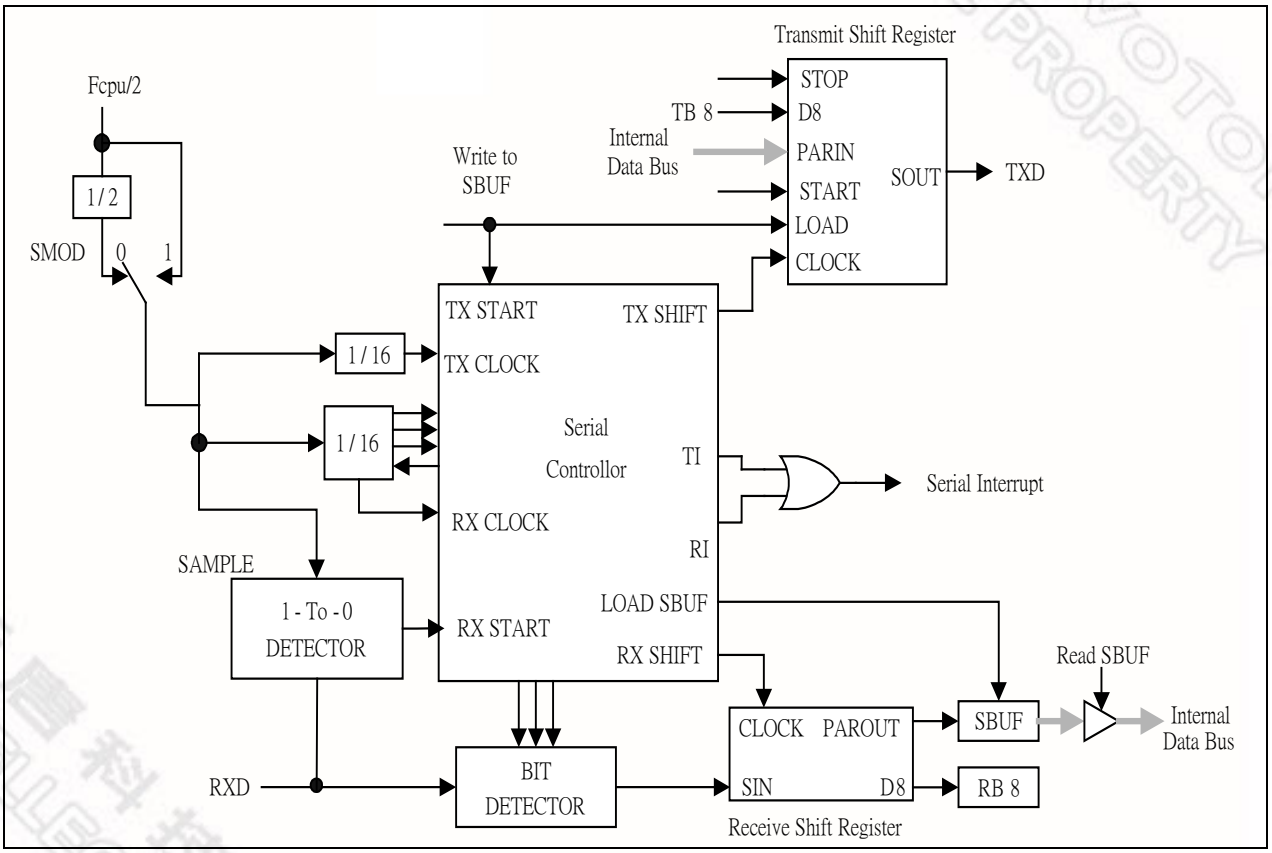


Figure 15-3: Uart Serial Port Mode 2

If the first bit detected after the falling edge of RxD pin, is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF. After shifting in 9 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.

15.5 Framing Error Detection

A Frame Error occurs when a valid stop bit is not detected. This could indicate incorrect serial data communication. Typically the frame error is due to noise and contention on the serial communication line. The N79E352(R) has the facility to detect such framing errors and set a flag which can be checked by software.

The Frame Error FE bit is located in SCON.7. This bit is normally used as SM0 in the standard 8051 family. However, in the N79E352(R) it serves a dual function and is called SM0/FE. There are actually two separate flags, one for SM0 and the other for FE. The flag that is actually accessed as SCON.7 is determined by SMOD0 (PCON.6) bit. When SMOD0 is set to 1, then the FE flag is indicated in SM0/FE. When SMOD0 is set to 0, then the SM0 flag is indicated in SM0/FE.

The FE bit is set to 1 by hardware but must be cleared by software. Note that SMOD0 must be 1 while reading or writing to FE. If FE is set, then any following frames received without any error will not clear the FE flag. The clearing has to be done by software.

15.6 Multiprocessor Communications

Multiprocessor communications makes use of the 9th data bit in modes 2 and 3. In the N79E352(R), the RI flag is set only if the received byte corresponds to the Given or Broadcast address. This hardware feature eliminates the software overhead required in checking every received address, and greatly simplifies the software programmer task.

In the multiprocessor communication mode, the address bytes are distinguished from the data bytes by transmitting the address with the 9th bit set high. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte. This ensures that they will be interrupted only by the reception of a address byte. The Automatic address recognition feature ensures that only the addressed slave will be interrupted. The address comparison is done in hardware not software.

The addressed slave clears the SM2 bit, thereby clearing the way to receive data bytes. With SM2 = 0, the slave will be interrupted on the reception of every single complete frame of data. The unaddressed slaves will be unaffected, as they will be still waiting for their address. In Mode 1, the 9th bit is the stop bit, which is 1 in case of a valid frame. If SM2 is 1, then RI is set only if a valid frame is received and the received byte matches the Given or Broadcast address.

The Master processor can selectively communicate with groups of slaves by using the Given Address. All the slaves can be addressed together using the Broadcast Address. The addresses for each slave are defined by the SADDR and SADEN SFRs. The slave address is an 8-bit value specified in the SADDR SFR. The SADEN SFR is actually a mask for the byte value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the slave address in SADDR.

16.2 The I2C Control Registers:

The I2C has 1 control register (I2CON) to control the transmit/receive flow, 1 data register (I2DAT) to buffer the Tx/Rx data, 1 status register (I2STATUS) to catch the state of Tx/Rx, recognizable slave address register for slave mode use and 1 clock rate control block for master mode to generate the variable baud rate.

16.2.1 The Address Registers, I2ADDR

I2C port is equipped with one slave address register. The contents of the register are irrelevant when I2C is in master mode. In the slave mode, the seven most significant bits must be loaded with the MCU's own slave address. The I2C hardware will react if the contents of I2ADDR are matched with the received slave address.

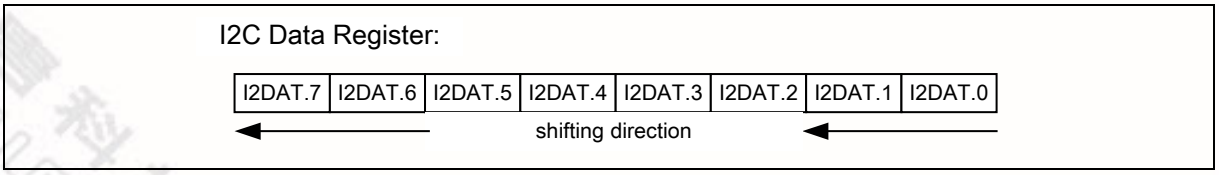
The I2C ports support the "General Call" function. If the GC bit is set the I2C port1 hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When GC bit is set, the I2C is in Slave mode, it can be received the general call address by 00H after Master send general call address to I2C bus, then it will follow status of GC mode. If it is in Master mode, the AA bit must be cleared when it will send general call address of 00H to I2C bus.

16.2.2 The Data Register, I2DAT

This register contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from or write to this 8-bit directly addressable SFR while it is not in the process of shifting a byte. This occurs when the bus is in a defined state and the serial interrupt flag (SI) is set. Data in I2DAT remains stable as long as SI bit is set. While data is being shifted out, data on the bus is simultaneously being shifted in; I2DAT always contains the last data byte present on the bus. Thus, in the event of arbitration lost, the transition from master transmitter to slave receiver is made with the correct data in I2DAT.

I2DAT and the acknowledge bit form a 9-bit shift register, the acknowledge bit is controlled by the hardware and cannot be accessed by the CPU. Serial data is shifted through the acknowledge bit into I2DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2DAT, the serial data is available in I2DAT, and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. Serial data is shifted out from I2DAT on the falling edges of SCL clock pulses, and is shifted into I2DAT on the rising edges of SCL clock pulses.



PRIORITY BITS		INTERRUPT PRIORITY LEVEL
IPXH	IPX	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

Table 18- 2: Four-level interrupts priority

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in registers IE or EIE. The IE register also contains a global disable bit, EA, which disables all interrupts at once.

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the IP0, IP0H, IP1, and IP1H registers. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. So, if two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve simultaneous requests of the same priority level.

Table below summarizes the interrupt sources, flag bits, vector address, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from Power Down mode.

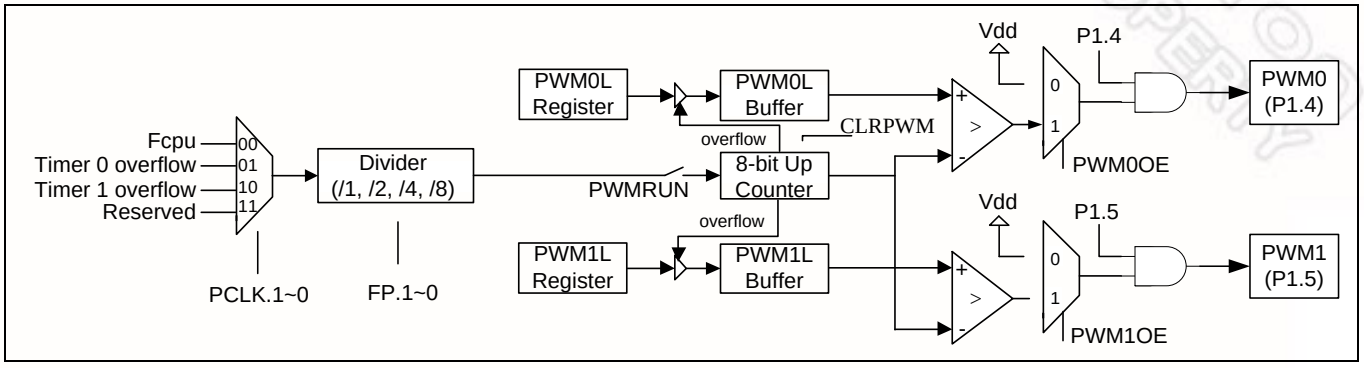
Source	Flag	Vector address	Enable bit	Flag cleared by	Priority bit	Arbitration ranking	Power-down wakeup
External Interrupt 0	IE0	0003H	EX0 (IE.0)	Hardware, Software	IP0H.0, IP0.0	1(highest)	Yes
Brownout Detect	BOF	002BH	EBO (EIE.6)	Hardware	IP1H.6, IP1.6	2	Yes
Watchdog Timer	WDIF	0053H	EWDI (EIE.4)	Software	IP1H.4, IP1.4	3	Yes
Timer 0 Overflow	TF0	000BH	ET0 (IE.1)	Hardware, Software	IP0H.1, IP0.1	4	No
I2C Interrupt	SI + TIF	0033h	EI2 (EIE.0)	Software	IP1H.0, IP1.0	5	No
External Interrupt 1	IE1	0013H	EX1 (IE.2)	Hardware, Software	IP0H.2, IP0.2	6	Yes
KBI	KBF	003BH	EKB (EIE.1)	Software	IP1H.1, IP1.1	7	Yes
Timer 1 Overflow	TF1	001BH	ET1 (IE.3)	Hardware, Software	IP0H.3, IP0.3	8	No
UART	RI + TI	0023H	ES (IE.4)	Software	IP0H.4, IP0.4	9	No

21. PULSE WIDTH MODULATED OUTPUTS (PWM)

The N79E352(R) contains two Pulse Width Modulated (PWM) channels which generate pulses of programmable length and interval. The output for PWM0 is on P1.4 and PWM1 on P1.5. After chip reset the internal output of the each PWM channel is a “1” (if PRHI=1). The PWM block diagram is shown as below figure. The interval between successive outputs is controlled by a 8–bit up-counter which uses the selectable clock sources. The clock sources supported are cpu clock, timer 0 overflow and timer 1 overflow, selectable by PWMCON3.PCLK.1~0 bits. The clock sources can be further divided with programmable PWMCON3.FP1~0 bits. When the counter reaches overflow, it is reloaded with zero.

The width of each PWM output pulse is determined by the value in the appropriate Compare registers, PWMnL (n=0,1). When the counter described above matches compare register value the PWM output is forced low. It remains low until the counter value overflow. The number of clock pulses that the PWMn output is low is given by:

t_{LO} = (FFh – PWMn+1)



A compare value of all zeroes, 00H, causes the output to remain permanently high. A compare value of all ones, FFH, results in the PWM output remaining permanently low.

The overall functioning of the PWM module is controlled by the contents of the PWMCON1 and PWMCON3 registers. The operation of most of the control bits are straightforward. The transfer Compare registers to the buffer registers is controlled by 8-bit counter overflow, while PWMCON1.7 (PWMRUN) allows the PWM to be either in the run or idle state. It has a CLRPWM bit to clear 8-bt up counter.

When the PWMRUN is cleared, the PWM outputs take on the state they had just prior to the bit being cleared. In general this state is not known. In order to place the outputs in a known state when PWMRUN is cleared the Compare registers can be written to either the “all 1” or “all 0” so the output will have the output desired when the counter is halted.

Note:

During PWM initial run, user is recommended to configure proper PWMn and/or PWM output pin (default high) follow by setting PWMRUN and CLRPWM bits, prior to enable PWMnOE. This is to avoid unexpected PWM output.

FS1	Internal Oscillator Output
0	11.0592MHz
1	22.1184MHz (default)

Internal Oscillator Selection Table

CBOV.1-0: Brownout level selection bits

These bits are used to select brownout voltage level.

CBOV.1	CBOV.0	Brownout Voltage
1	x	Brownout voltage is 2.6V
0	1	Brownout voltage is 3.8V
0	0	Brownout voltage is 4.5V

C1: MOVC inhibit enable bit

MOVC inhibit enable bit	MOVC access
0	The MOVC instruction in external memory cannot access the code in internal memory.
1	No restriction.

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.