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	8051/52
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	38
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w79e225apg

PIN DESCRIPTION, continued

SYMBOL	TYPE	INITIAL STATE	DESCRIPTIONS
P4.0-P4.3	I/O S H	High	**PORT 4**: 4-bit multipurpose programmable I/O port with alternate functions. The Port 4 has four different operation modes. P4.0, STADC P4.1, T2EX, IC2 P4.2 P4.3 **Note:** P4.2 & P4.3 are not supported in PLCC44 pins package.
P5.0-P5.1	I/O S	Tri-state	**PORT 5**: 2-bit, bit-directional I/O port. This port is not bit addressable. The alternate functions are described below: P5.0, PWM6 P5.1, PWM7 **Note:** P5.0 & P5.1 are not supported in PLCC44 pins package.

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

5.1 Port 4

Port 4, SFR P4 at address A5H, is a 4-bit multipurpose programmable I/O port which functions are I/O and chip-select function. It has four different operation modes:

- Mode 0 - P4.0 ~ P4.3 is 4-bit bi-directional I/O port which is the same as port 1. The default Port 4 is a general I/O function.
- Mode1 - P4.0 ~ P4.3 are read data strobe signals which are synchronized with $\overline{RD}$ signal at specified addresses. These read data strobe signals can be used as chip-select signals for external peripherals.
- Mode2 - P4.0 ~ P4.3 are write data strobe signals which are synchronized with $\overline{WR}$ signal at specified addresses. These write data strobe signals can be used as chip-select signals for external peripherals.
- Mode3 - P4.0 ~ P4.3 are read/write data strobe signals which are synchronized with $\overline{RD}$ or $\overline{WR}$ signal at specified addresses. These read/write data strobe signals can be used as chip-select signals for external peripherals.

When Port 4 is configured with the feature of chip-select signals, the chip-select signal address range depends on the contents of the SFR P4xAH, P4xAL, P4CONA and P4CONB. P4xAH and P4xAL contain the 16-bit base address of P4.x. P4CONA and P4CONB contain the control bits to configure the Port 4 operation mode.

Continued

SYMBOL	DEFINITION	ADDRESS	MSB BIT_ADDRESS, LSB SYMBOL								RESET
FSPLT	FAULT SAMPLING TIME REGISTER	C5H	SCMP1	SCMP0	SFP1	SFP0	SFCEN	SFCST	SFCDIR	LSBD	0000 0000B
PMR	POWER MANAGEMENT REGISTER	C4H	-	-	-	-	-	ALEOFF	-	DME0	xxxx x0x0B
T3CON	TIMER 3 CONTROL	C3H	TF3	-	-	-	-	TR3	-	CMP/$\overline{RL3}$	0xxx x0x0B
T3MOD	TIMER 3 MODE CONTROL	C2H	ENLD	ICEN2	ICEN1	ICEN0	T3CR	-	-	-	0000 0xxxB
SBUF1	SERIAL BUFFER 1	C1H	SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0	xxxx xxxxB
SCON1	SERIAL CONTROL 1	C0H	(BF) SM0_1/FE_1	(BE) SM1_1	(BD) SM2_1	(BC) REN_1	(BB) TB8_1	(BA) RB8_1	(B9) TI_1	(B8) RI_1	0000 0000B
PWM4H	PWM 4 HIGH BITS REGISTER	BFH	-	-	-	-	PWM4.11	PWM4.10	PWM4.9	PWM4.8	xxxx 0000B
PWMEN	PWM OUTPUT ENABLE REGISTER	BEH	PWM7EN	PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	0000 0000B
PIO	PWM PIN OUTPUT SOURCE SELECT	BDH	PIO7	PIO6	PIO5	PIO4	PIO3	PIO2	PIO1	PIO0	0000 0000B
POVD	PWM OUTPUT STATE REGISTERS	BCH	POVD.7	POVD.6	POVD.5	POVD.4	POVD.3	POVD.2	POVD.1	POVD.0	0000 0000B
POVM	PWM OUTPUT OVERRIDE CONTROL REGISTERS	BBH	POVM.7	POVM.6	POVM.5	POVM.4	POVM.3	POVM.2	POVM.1	POVM.0	0000 0000B
SADEN1	SLAVE ADDRESS MASK 1	BAH	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0	0000 0000B
SADEN	SLAVE ADDRESS MASK	B9H	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0	0000 0000B
IP	INTERRUPT PRIORITY	B8H	(BF)	(BE) PADC	(BD) PT2	(BC) PS	(BB) PT1	(BA) PX1	(B9) PT0	(B8) PX0	0000 0000B
IPH	INTERRUPT HIGH PRIORITY	B7H	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x000 0000B
EIP1H	EXTENDED INTERRUPT HIGH PRIORITY 1	B6H	-	-	PNVMIH	PCPTFH	PT3H	PBKFH	PPWMH	PSPIH	xx00 0000B
RCAP3H	RELOAD CAPTURE 3 HIGH REGISTER	B5H	RCAP3H.7	RCAP3H.6	RCAP3H.5	RCAP3H.4	RCAP3H.3	RCAP3H.2	RCAP3H.1	RCAP3H.0	0000 0000B
RCAP3L	RELOAD CAPTURE 3 LOW REGISTER	B4H	RCAP3L.7	RCAP3L.6	RCAP3L.5	RCAP3L.4	RCAP3L.3	RCAP3L.2	RCAP3L.1	RCAP3L.0	0000 0000B
P5	PORT 5	B1H	-	-	-	-	-	-	PWM7	PWM6	xxxx xx11B
P3	PORT 3	B0H	(B7) RD	(B6) WR	(B5) T1/ IC1/QEB	(B4) T0/ ICO/QEA	(B3) /INT1	(B2) /INT0	(B1) TXD	(B0) RXD	1111 1111B
SFRCN	F/W FLASH CONTROL	AFH	-	WFWIN	NOE	NCE	CTRL3	CTRL2	CTRL1	CTRL0	x011 1111B
SFRFD	F/W FLASH DATA	AEH	D7	D6	D5	D4	D3	D2	D1	D0	xxxx xxxxB
SFRAH	F/W FLASH HIGH ADDRESS	ADH	A15	A14	A13	A12	A11	A10	A9	A8	0000 0000B
SFRAL	F/W FLASH LOW ADDRESS	ACH	A7	A6	A5	A4	A3	A2	A1	A0	0000 0000B
SADDR1	SLAVE ADDRESS 1	AAH	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0	0000 0000B
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	(AF) EA	(AE) EADC	(AD) ET2	(AC) ES	(AB) ET1	(AA) EX1	(A9) ET0	(A8) EX0	0000 0000B

WD1, WD0: Mode Select bits:

These bits determine the time-out periods for the Watchdog Timer. The reset time-out period is 512 clocks more than the interrupt time-out period.

WD1	WD0	INTERRUPT TIME-OUT	RESET TIME-OUT
0	0	2^{17}	2^{17} + 512
0	1	2^{20}	2^{20} + 512
1	0	2^{23}	2^{23} + 512
1	1	2^{26}	2^{26} + 512

MD2, MD1, MD0: Stretch MOVX select bits:

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

CLOCK CONTROL 1

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CCDIV1	CCDIV0

Mnemonic: CKCON1 Address: 8Fh

BIT	NAME	FUNCTION
7-2	-	Reserved.
1-0	CCDIV	Timer 3 clock select.

CCDIV1	CCDIV0	Timer 3 clock
0	0	Fosc
0	1	Fosc/4
1	0	Fosc/16
1	1	Fosc/32

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

BIT	NAME	FUNCTION
7-0	POVD	PWM Override Data represents the value of PWM[7:0] respectively in override mode. 1 = Output on PWM I/O pin is ACTIVE when the corresponding PWM output override bit is cleared. 0 = Output on PWM I/O pin is INACTIVE when the corresponding PWM output override bit is cleared.

PWM PIN OUTPUT SOURCE SELECT

Bit:	7	6	5	4	3	2	1	0
	PIO7	PIO6	PIO5	PIO4	PIO3	PIO2	PIO1	PIO0

Mnemonic: PIO Address: BDh

BIT	NAME	FUNCTION
7-0	PIO.x	Select pin output source from PWM or I/O register; x=0~7; PIOn is effective only when option bit PWMOE/PWMEE/PWM6E/PWM7E is in enabled status. Reset value=0; 1 = Correspondent I/O pin with high source/sink current. 0 = PWMn output; n=0~7 with high source/sink current.

PWM OUTPUT ENABLE REGISTER

Bit:	7	6	5	4	3	2	1	0
	PWM7EN	PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN

Mnemonic: PWMEN Address: BEh

BIT	NAME	FUNCTION
6,4,2,0	PWMeEN	Set high to enable even PWM output; e = 0,2,4,6; Reset value=0; 1 = Enable PWM output. 0 = Disable PWM output.
7,5,3,1	PWMoEN	Set high to enable odd PWM output; o = 1,3,5,7; Reset value=0; 1 = Enable PWM output. 0 = Disable PWM output.

PWM 4 HIGH BITS REGISTER

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	PWM4.11	PWM4.10	PWM4.9	PWM4.8

Mnemonic: PWM4H Address: BFh

BIT	NAME	FUNCTION
7~4	-	Reserved
3~0	PWM4.11 ~PWM4.8	The PWM 4 Register bit 11~8.

SERIAL PORT CONTROL 1

Bit:	7	6	5	4	3	2	1	0
	SM0_1/FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1

Mnemonic: SCON1 Address: C0h

BIT	NAME	FUNCTION
7	SM0_1/ FE_1	Serial Port 1 mode select bit 0 or Framing Error Flag: This bit is controlled by the SMOD0 bit in the PCON register. (SM0) See table below. (FE) This bit indicates an invalid stop bit. It must be manually cleared by software.
6	SM1_1	Serial Port 1 mode select bit 1. See table below.
5	SM2_1	Serial Port Clock or Multi-Processor Communication. (Mode 0) This bit controls the serial port clock. If set to zero, the serial port runs at a divide-by-12 clock of the oscillator. This is compatible with the standard 8051/52. If set to one, the serial clock is a divide-by-4 clock of the oscillator. (Mode 1) If SM2_1 is set to one, RI_1 is not activated if a valid stop bit is not received. (Modes 2 / 3) This bit enables multi-processor communication. If SM2_1 is set to one, RI_1 is not activated if RB8_1, the ninth data bit, is zero.
4	REN_1	Receive enable: 1: Enable serial reception. 0: Disable serial reception.
3	TB8_1	(Modes 2 / 3) This is the 9th bit to transmit. This bit is set by software.
2	RB8_1	(Mode 0) No function. (Mode 1) If SM2_1 = 0, RB8_1 is the stop bit that was received. (Modes 2 / 3) This is the 9th bit that was received.
1	TI_1	Transmit interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or at the beginning of the stop bit in the other modes during serial transmission. This bit must be cleared by software.
0	RI_1	Receive interrupt flag: This flag is set by the hardware at the end of the 8th bit in mode 0 or halfway through the stop bits in the other modes during serial reception. However, SM2_1 can restrict this behavior. This bit can only be cleared by software.

SM1_1, SM0_1: Mode Select bits:

SM0_1	SM1_1	MODE	DESCRIPTION	LENGTH	BAUD RATE
0	0	0	Synchronous	8	Tclk divided by 4 or 12
0	1	1	Asynchronous	10	Variable
1	0	2	Asynchronous	11	Tclk divided by 32 or 64
1	1	3	Asynchronous	11	Variable

The ADCI and ADCS control the ADC conversion as below:

ADCI	ADCS	ADC STATUS
0	0	ADC not busy; A conversion can be started.
0	1	ADC busy; Start of a new conversion is blocked.
1	0	Conversion completed; Start of a new conversion requires ADCI = 0.
1	1	This is an internal temporary state that user can ignore it.

ADC CONVERTER RESULT HIGH REGISTER

Bit:	7	6	5	4	3	2	1	0
	ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2

Mnemonic: ADCH Address: E2h

BIT	NAME	FUNCTION
7-0	ADC[9:2]	8 MSB of 10 bit A/D conversion result. ADCH is a read only register.

ADC CONVERTER RESULT LOW REGISTER

Bit:	7	6	5	4	3	2	1	0
	ADCLK.1	ADCLK.0	-	-	-	-	ADC.1	ADC.0

Mnemonic: ADCL Address: E3h

BIT	NAME	FUNCTION
7-6	ADCLK	ADC Clock Frequency Select. The 10 bit ADC needs a clock to drive the converting that the clock frequency may not over 4MHz. ADCLK[1:0] controls the frequency of the clock to ADC block: ADCLK.1 / ADCLK.0 / ADC Clock Frequency 0 / 0 / Crystal clock / 4 (Default) 0 / 1 / Crystal clock / 8 1 / 0 / Crystal clock / 16 1 / 1 / Reserved
1-0	ADC	2 LSB of 10-bit A/D conversion result. Both bits are read only.

PWM DEAD-TIME CONTROL REGISTER 1

Bit:	7	6	5	4	3	2	1	0
	PDTC1.7	PDTC1.6	PDTC1.5	PDTC1.4	PDTC1.3	PDTC1.2	PDTC1.1	PDTC1.0

Mnemonic: PDTC1 Address: E5h

Continued

OP-CODE	HEX CODE	BYTES	W79E22X SERIES MACHINE CYCLE	W79E22X SERIES CLOCK CYCLES	8032 CLOCK CYCLES	W79E22X SERIES VS. 8032 SPEED RATIO
JBC bit, rel	10	3	4	16	24	1.5
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
CJNE R7, #data, rel	BF	3	4	16	24	1.5
DJNZ R0, rel	D8	2	3	12	24	2
DJNZ R1, rel	D9	2	3	12	24	2
DJNZ R5, rel	DD	2	3	12	24	2
DJNZ R2, rel	DA	2	3	12	24	2
DJNZ R3, rel	DB	2	3	12	24	2
DJNZ R4, rel	DC	2	3	12	24	2
DJNZ R6, rel	DE	2	3	12	24	2
DJNZ R7, rel	DF	2	3	12	24	2
DJNZ direct, rel	D5	3	4	16	24	1.5

Table 8-1: Instruction Set for W79E22X SERIES

9. POWER MANAGEMENT

The W79E22X SERIES provides idle mode and power-down mode to control power consumption. These modes are discussed in the next two sections.

9.1 Idle Mode

The user can put the device into idle mode by writing 1 to the bit PCON.0. The instruction that sets the idle bit is the last instruction that will be executed before the device goes into Idle Mode. In the Idle mode, the clock to the CPU is halted, but not to the Interrupt, Timer, Watchdog timer, PWM, ADC and Serial ports blocks. This forces the CPU state to be frozen; the Program counter, the Stack Pointer, the Program Status Word, the Accumulator and the other registers hold their contents. The ALE and $\overline{\text{PSEN}}$ pins are held high during the Idle state. The port pins hold the logical states they had at the time Idle was activated. The Idle mode can be terminated in two ways. Since the interrupt controller is still active, the activation of any enabled interrupt can wake up the processor. This will automatically clear the Idle bit, terminate the Idle mode, and the Interrupt Service Routine (ISR) will be executed. After the ISR, execution of the program will continue from the instruction which put the device into Idle mode.

The Idle mode can also be exited by activating the reset. The device can be put into reset either by applying a high on the external RST pin, a Power on reset condition or a Watchdog timer reset. The external reset pin has to be held high for at least two machine cycles i.e. 8 clock periods to be recognized as a valid reset. In the reset condition the program counter is reset to 0000h and all the SFRs are set to the reset condition. Since the clock is already running there is no delay and execution starts immediately. In the Idle mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt which will wake up the device. The software must reset the Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the device is exiting from an Idle mode with a reset, the instruction following the one which put the device into Idle mode is not executed. So there is no danger of unexpected writes.

9.2 Power Down Mode

The device can be put into Power Down mode by writing 1 to bit PCON.1. The instruction that does this will be the last instruction to be executed before the device goes into Power Down mode. In the Power Down mode, all the clocks are stopped and the device comes to a halt. All activity is completely stopped and the power consumption is reduced to the lowest possible value. In this state the ALE and $\overline{\text{PSEN}}$ pins are pulled low (if PWDNH=0). The port pins output the values held by their respective SFRs.

The device will exit the Power Down mode with a reset or by an external interrupt pin enabled (external interrupts 0 and 1). An external reset can be used to exit the Power down state. The high on RST pin terminates the Power Down mode, and restarts the clock. The program execution will restart from 0000h. In the Power down mode, the clock is stopped, so the Watchdog timer cannot be used to provide the reset to exit Power down mode.

The device can be waken up from the Power Down mode by forcing an external interrupt pin activation, provided the corresponding interrupt is enabled, while the global enable (EA) bit is set. If these conditions are met, then either a low-level or a falling-edge at external interrupt pin will re-start the oscillator. The device will then execute the interrupt service routine for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after one which put the device into Power Down mode and continues from there.

12.1.5 Mode 3

Mode 3 is used when an extra 8-bit timer is needed. It has different effect on Timer 0 and Timer 1. TL0 and TH0 become two separate 8 bits counters. TL0 uses the Timer 0 control bits C/$\overline{T}$, GATE, TR0, $\overline{INT0}$ and TF0, and it can be used to count clock cycles (clock/12 or clock/4) or falling edges on pin T0, as determined by C/$\overline{T}$ (TMOD.2). TH0 becomes a clock-cycle counter (clock/12 or clock/4) and takes over the Timer 1 enable bit TR1 and overflow flag TF1. In contrast, mode 3 simply freezes Timer 1. If Timer 0 is in mode 3, Timer 1 can still be used in modes 0, 1 and 2, but it no longer has control over TR1 and TF1. Therefore when Timer 0 is in Mode 3, Timer 1 can only count oscillator cycles, and it does not have an interrupt or flag. With these limitations, baud rate generation is its most practical application, but other time-base functions may be achieved by reading the registers.

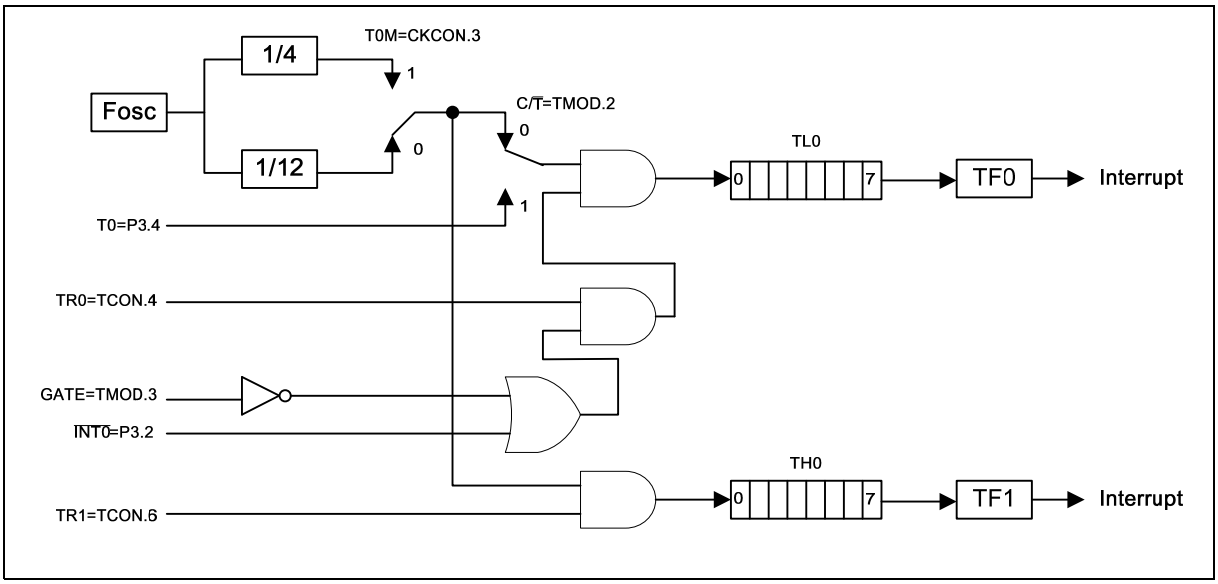


Figure 12-3: Timer/Counter Mode 3

12.2 Timer/Counter 2

Timer/Counter 2 is a 16-bit up/down-counter equipped with a capture/reload capability. The clock source for Timer/Counter 2 may be the external T2 pin (C/$\overline{T2}$ = 1) or the crystal oscillator (C/$\overline{T2}$ = 0), divided by 12 or 4. The clock is enabled and disabled by TR2. Timer/Counter 2 runs in one of four operating modes, each of which is discussed below.

14.11 PWM Power-down/Wakeup Procedures

The following flow diagrams describe the possible pwm procedures users require to take care prior to the product power-down/wake-up. The power-down procedure below will result in PWM output a low state after power-down. To output a high state, users may set PWMn to FFFh and initial state set to high through option bit (EPOL/OPOL).

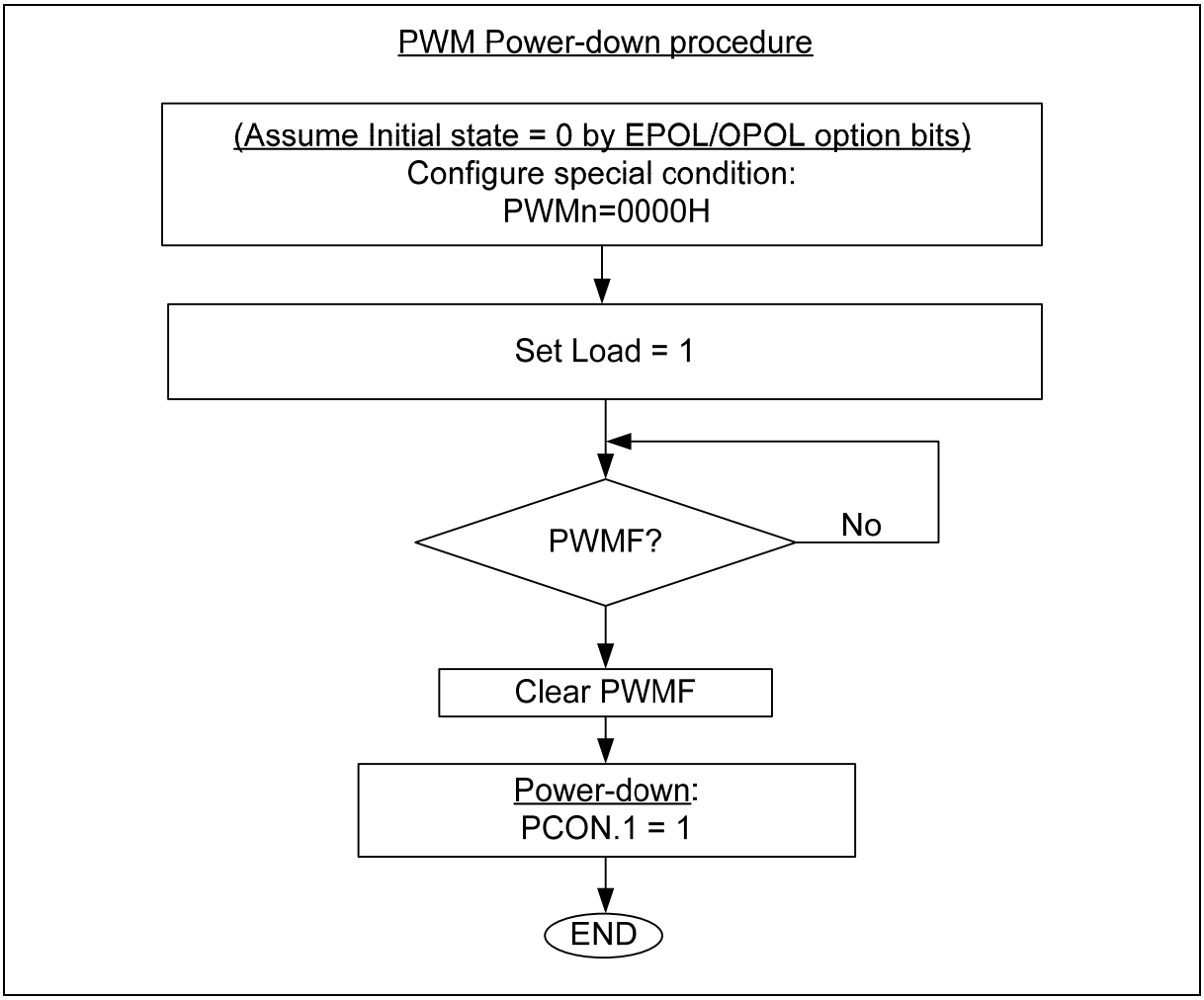


Figure 14-27: Example of PWM power-down procedure (pwm output low state)

Figure 16-1 Serial Port Mode 0

The serial port receives data when REN is 1 and RI is zero. The shift clock (TxD) is activated, and the serial port latches data on the rising edge of the shift clock. The external device should, therefore, present data on the falling edge of the shift clock. This process continues until all eight bits have been received. The RI flag is set in C1 following the last rising edge of the shift clock, which stops reception until RI is cleared by the software.

16.2 Mode 1

In Mode 1, full-duplex asynchronous communication is used. Frames consist of ten bits transmitted on TXD and received on RXD. The ten bits consist of a start bit (0), eight data bits (LSB first), and a stop bit (1). When receiving, the stop bit goes into RB8 in SCON. The baud rate in this mode is 1/16 or 1/32 of the Timer 1 overflow, and since Timer 1 can be set to a wide range of values, a wide variation of baud rates is possible.

Transmission begins with a write to SBUF but is synchronized with the divide-by-16 counter, not the write to SBUF. The start bit is put on TxD at C1 following the first roll-over of the divide-by-16 counter, and the next bit is placed at C1 following the next rollover. After all eight bits are transmitted, the stop bit is transmitted. The TI flag is set in the next C1 state, or the tenth rollover of the divide-by-16 counter after the write to SBUF.

Reception is enabled when REN is high, and the serial port starts receiving data when it detects a falling edge on RxD. The falling-edge detector monitors the RxD line at 16 times the selected baud rate. When a falling edge is detected, the divide-by-16 counter is reset to align the bit boundaries with the rollovers of the counter. The 16 states of the counter divide the bit time into 16 slices. Bit detection is done on a best-of-three basis using samples at the 8th, 9th and 10th counter states. If the first bit after the falling edge is not 0, the start bit is invalid, reception is aborted immediately, and the serial port resumes looking for a falling edge on RxD. If a valid start bit is detected, the rest of the bits are shifted into SBUF. After shifting in eight data bits, the stop bit is received. Then, if;

1. RI is 0, and
2. SM2 is 0 or the received stop bit is 1,

the stop bit goes into RB8, the eight data bits go into SBUF, and RI is set. Otherwise, the received frame is lost. In the middle of the stop bit, the receiver resumes looking for a falling edge on RxD.

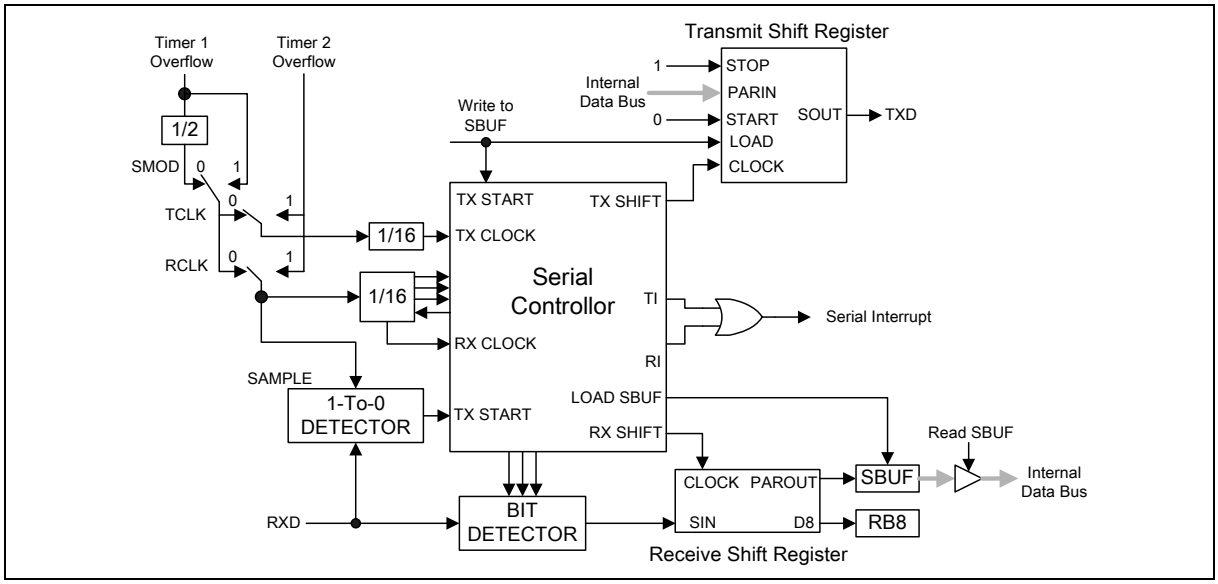


Figure 16-2 Serial Port Mode 1

17. I2C SERIAL PORTS

The I2C bus uses two wires (SCL and SDA) to transfer information between devices connected to the bus. The main features of the I2C bus are:

– Bi-directional data transfer between masters and slaves.

– Multi-master bus (no central master).

– Arbitration between simultaneously transmitting masters without corruption of serial data on the bus.

– Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.

– Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer.

– The I2C bus may be used for test and diagnostic purposes.

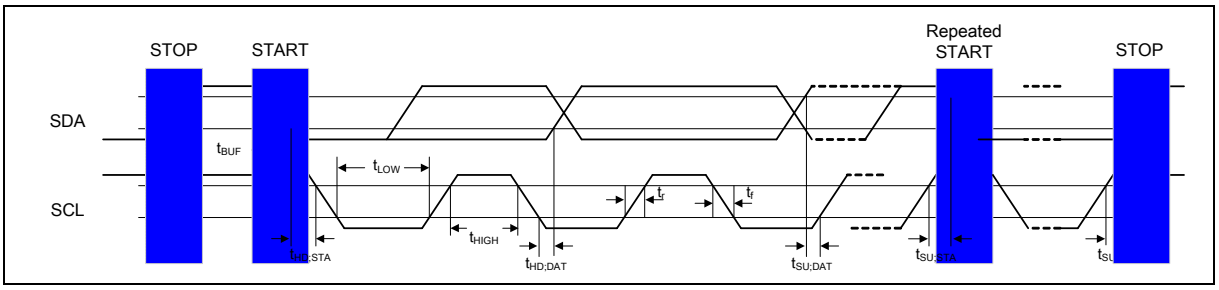


Figure 17-1: I2C Bus Timing

The device's on-chip I2C logic provides the serial interface that meets the I2C bus standard mode specification. The I2C logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (I2STATUS) reflects the status of the I2C bus.

The I2C port, SCL and SDA are at P2.6 and P2.7. When the I/O pins are used as I2C port, user must set the pins to logic high in advance. When I2C port is enabled by setting ENS to high, the internal states will be controlled by I2CON and I2C logic hardware. Once a new status code is generated and stored in I2STATUS, the I2C interrupt flag (SI) will be set automatically. If both EA and EI2C are also in logic high, the I2C interrupt is requested. The 5 most significant bits of I2STATUS stores the internal state code, the lowest 3 bits are always zero and the content keeps stable until SI is cleared by software.

17.1 SIO Port

The SIO port is a serial I/O port, which supports all transfer modes from and to the I2C bus. The SIO port handles byte transfers autonomously. To enable this port, the bit ENS1 in I2CON should be set to '1'. The CPU interfaces to the SIO port through the seven special function registers. The detail description of these registers can be found in the I2C Control registers section. The SIO H/W interfaces to the I2C bus via two pins: SDA (P2.7, serial data line) and SCL (P2.6, serial clock line). Pull up resistor is needed for Pin P2.6 and P2.7 for I2C operation as these are 2 open drain pins.

17.2 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.

SYMBOL	DEFINITION	ADDRESS	MSB		BIT_ADDRESS, SYMBOL				LSB		RESET
I2TIMER	I2C Timer Counter Register	EFH	-	-	-	-	-	ENTI	DIV4	TIF	xxxx x000B
I2CLK	I2C Clock Rate	EEH	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0	0000 0000B
I2STATUS	I2C Status Register	EDH	I2STATUS.7	I2STATUS.6	I2STATUS.5	I2STATUS.4	I2STATUS.3	-	-	-	1111 1000B
I2DAT	I2C Data	ECH	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0	0000 0000B
I2ADDR	I2C Slave Address	EAH	ADDR.7	ADDR.6	ADDR.5	ADDR.4	ADDR.3	ADDR.2	ADDR.1	GC	0000 0000B
I2CON	I2C Control Register	E9H	-	ENS	STA	STO	SI	AA	I2CIN	-	x000 000xB
I2CSADEN	I2C Maskable Slave Address	F6H	I2CSADEN.7	I2CSADEN.6	I2CSADEN.5	I2CSADEN.4	I2CSADEN.3	I2CSADEN.2	I2CSADEN.1	I2CSADEN.0	1111 1110B

Table 17-1: Control Registers of I2C Ports

17.2.1 Slave 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 device is in slave mode which can receive the General Call address(00H) sent by Master on the I2C bus. This special slave mode is referred to as GC mode.

17.2.2 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. Data in I2DAT remains stable as long as SI is set. The MSB is shifted out first.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 which shifts in or out an 8-bit byte, followed by an acknowledge bit. The acknowledge bit is controlled by the hardware and cannot be accessed by the CPU. Serial data is shifted 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.

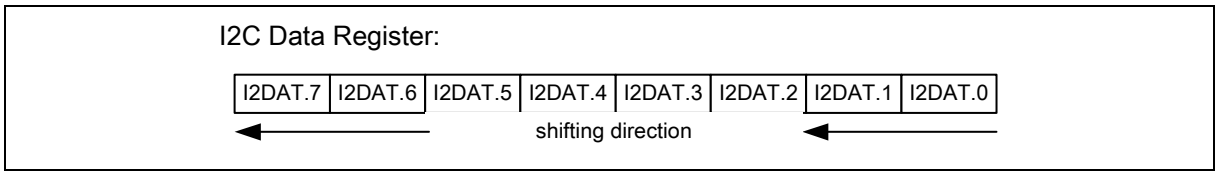
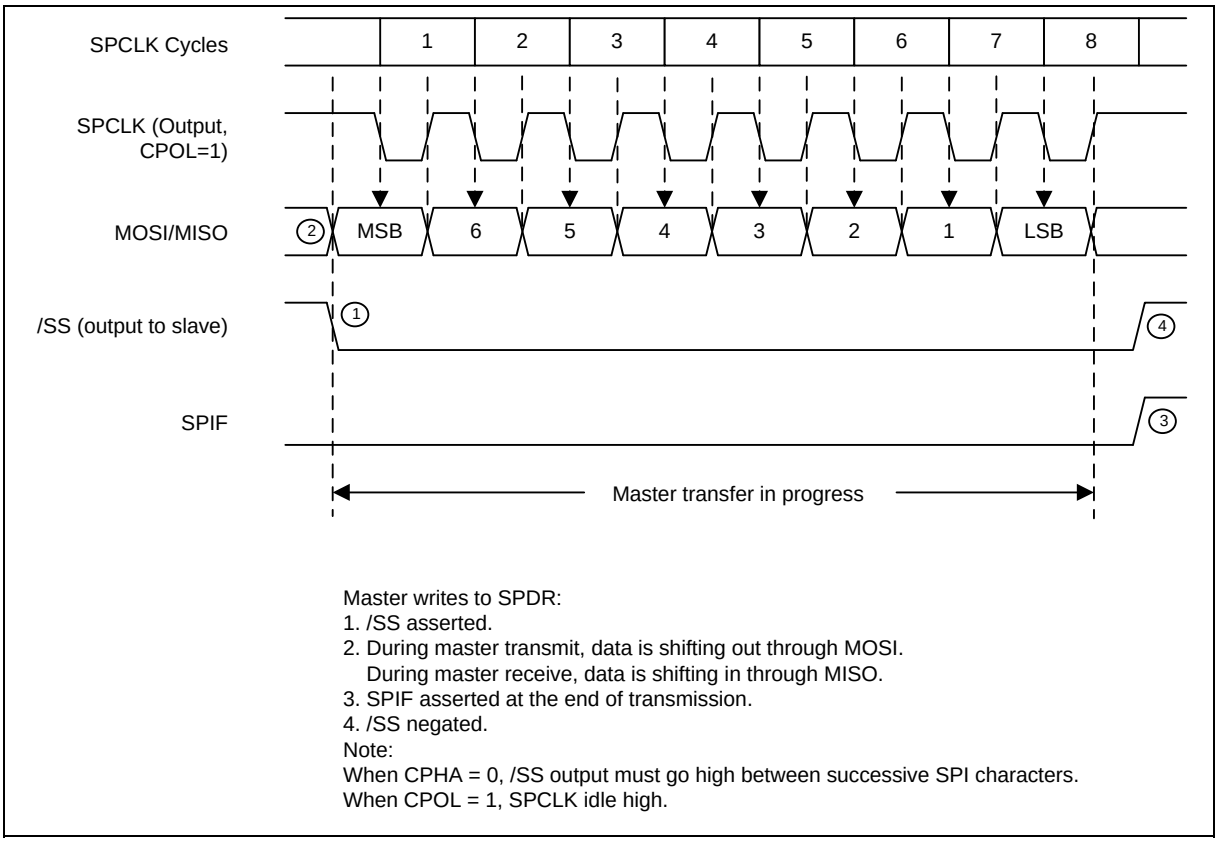


Figure 17-2: I2C Data Shift

17.2.3 Control Register, I2CON

The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by hardware: the SI bit is set when the I2C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when ENS = "0".

ENS I2C serial function block enable bit. When ENS=1 the I2C serial function enables. The port latches of SDA1 and SCL1 must be set to logic high.



Master writes to SPDR:
1. /SS asserted.
2. During master transmit, data is shifting out through MOSI.
   During master receive, data is shifting in through MISO.
3. SPIF asserted at the end of transmission.
4. /SS negated.

Note:
When CPHA = 0, /SS output must go high between successive SPI characters.
When CPOL = 1, SPCLK idle high.

Figure 18-3: Master Mode Transmission (CPOL = 1, CPHA = 0)

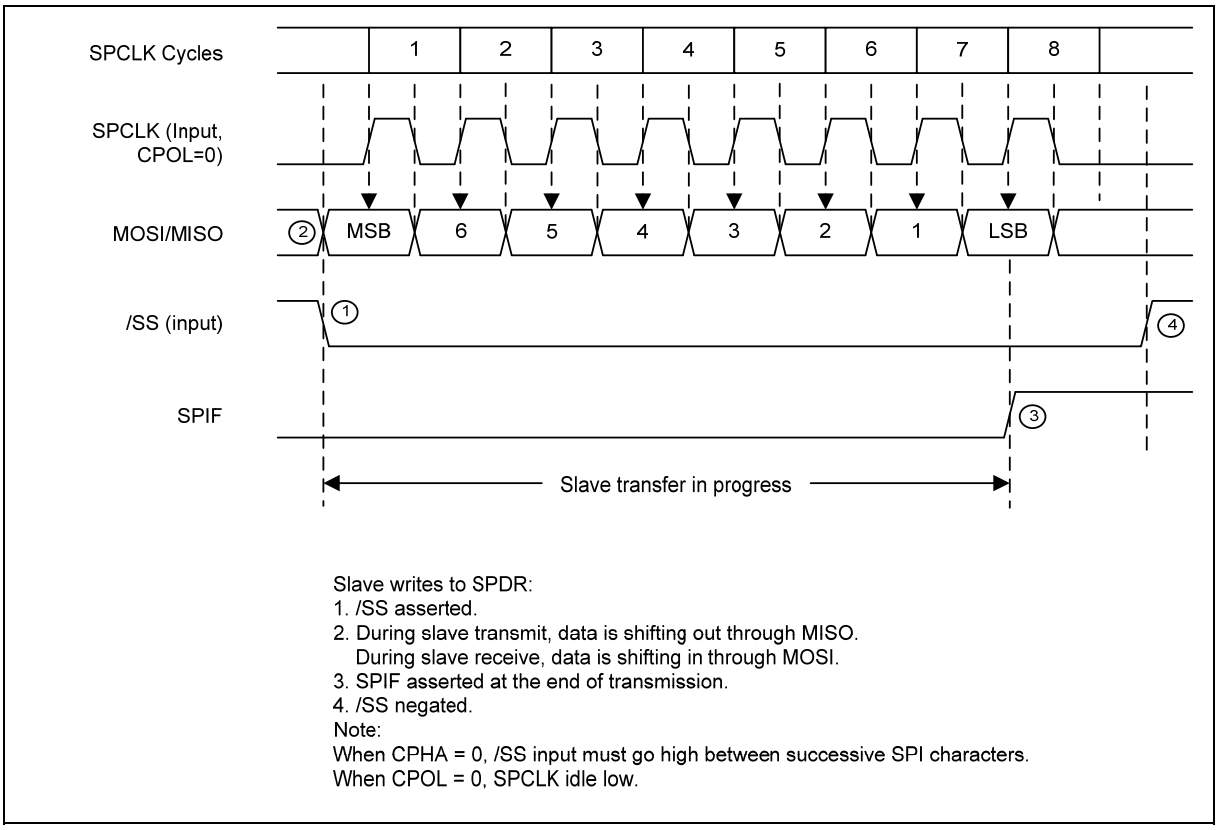


Figure 18-6: Slave Mode Transmission (CPOL = 0, CPHA = 0)

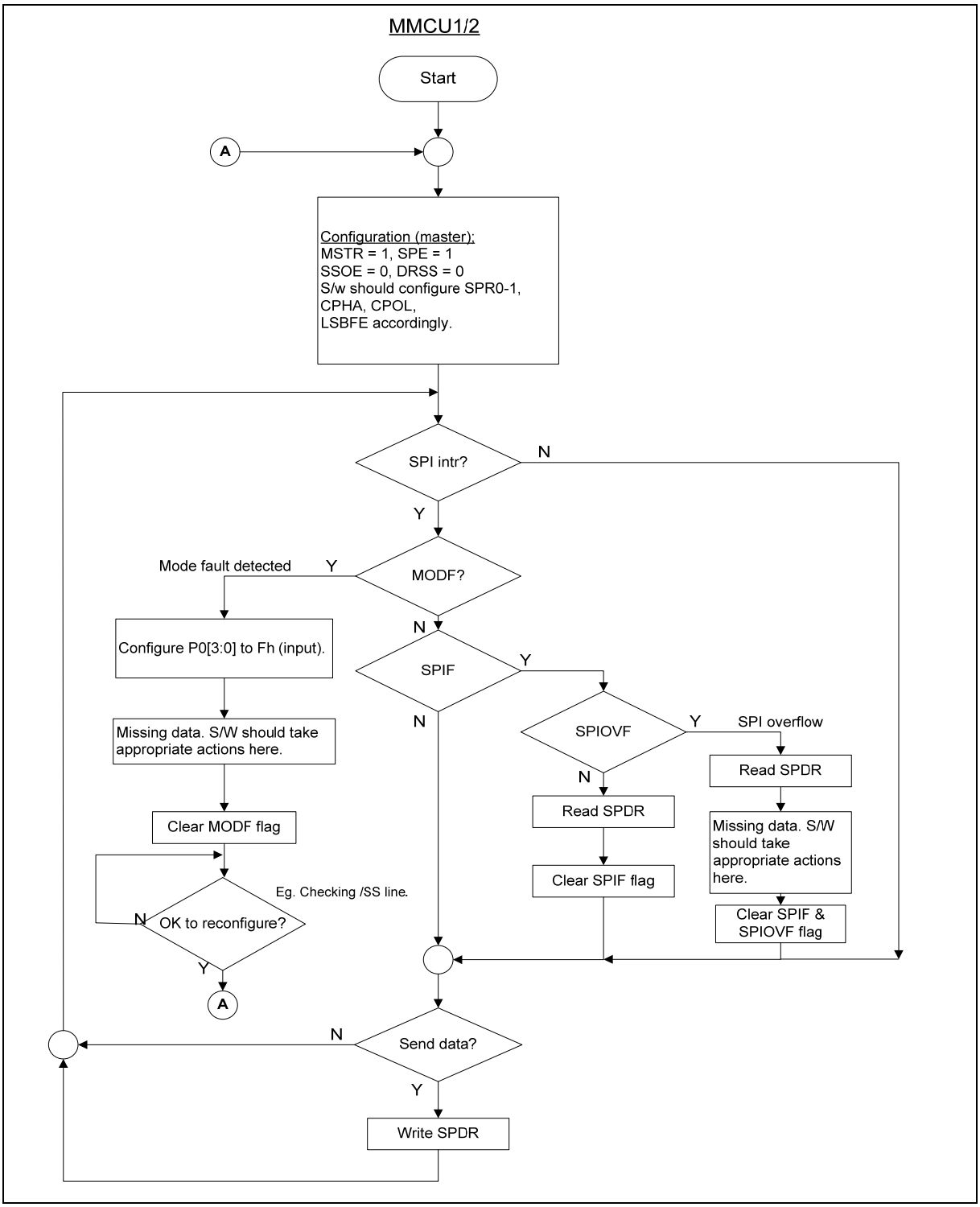


Figure 18-13: SPI multi-master slave s/w flow diagram

19. ANALOG-TO-DIGITAL CONVERTER

The ADC contains a digital-to-analog converter (DAC) that converts the contents of a successive approximation register to a voltage (V_{DAC}), which is compared to the analog input voltage (Vin). The output of the comparator is then fed back to the successive approximation control logic that controls the successive approximation register. This is illustrated in the figure below.

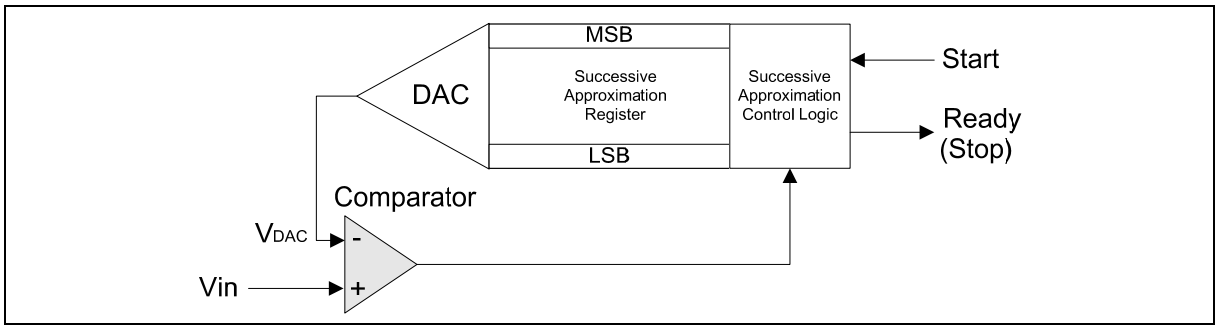


Figure 19-1: Successive Approximation ADC

19.1 Operation of ADC

A conversion can be initiated by software only or by either hardware or software. The software only start mode is selected when control bit ADCCON.5 (ADCEX) =0. A conversion is then started by setting control bit ADCCON.3 (ADCS) to 1. The hardware or software start mode is selected when ADCCON.5 =1, and a conversion may be started by setting ADCCON.3 = 1 as above or by applying a rising edge to external pin STADC (P4.0). When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

User sets ADCS to start converting then ADCS remains high while ADC is converting signal and will be automatically cleared by hardware when ADC conversion is completed. The end of the 10-bit conversion is flagged by control bit ADCCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCL.1 (ADC.1) and ADCL.0 (ADC.0). The user may ignore the two least significant bits in ADCL and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 ADC clock input cycles.

Control bits from ADCCON.0 to ADCCON.2 are used to control an analog multiplexer which selects one of eight analog channels. An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

The device supports maximum 8 analog input ports. 8 analog input ports share the I/O pins from P1.0 to P1.7. These I/O pins are switched to analog input ports by setting the bits of ADC Input Pin Select Register (DDIO) to logic 1.

24.4 The ADC Converter DC ELECTRICAL CHARACTERISTICS

(V_{DD}–V_{SS} = 3.0~5V±10%, T_A = -40~85°C, Fosc = 20MHz, unless otherwise specified.)

PARAMETER	SYMBOL	SPECIFICATION			
		MIN.	TYP.	MAX.	UNIT
Analog input	AVin	V_{SS}-0.2		V_{DD}+0.2	V
ADC clock	ADCCLK	200KHz	-	5MHz	Hz
Conversion time	t_C		$52t_{ADC}$[1]		us
Differential non-linearity	DNL	-1	-	+1	LSB
Integral non-linearity	INL	-2	-	+2	LSB
Offset error	Ofe	-1	-	+1	LSB
Gain error	Ge	-1	-	+1	%
Absolute voltage error	Ae	-3	-	+3	LSB

Notes:1. t_{ADC}: The period time of ADC input clock.

24.5 I2C Bus Timing Characteristics

PARAMETER	SYMBOL	STANDARD MODE I2C BUS		UNIT
		MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	100	kHz
bus free time between a STOP and START condition	t_{BUF}	4.7	-	uS
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{Hd;STA}$	4.0	-	uS
Low period of the SCL clock	t_{LOW}	4.7	-	uS
HIGH period of the SCL clock	t_{HIGH}	4.0	-	uS
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	-	uS
Data hold time	$t_{HD;DAT}$	5.0	-	uS
Data set-up time	$t_{SU;DAT}$	250	-	nS
Rise time of both SDA and SCL signals	t_r	-	1000	nS
Fall time of both SDA and SCL signals	t_f	-	300	nS
Set-up time for STOP condition	$t_{SU;STO}$	4.0	-	uS
Capacitive load for each bus line	C_b	-	400	pF

```
TA              EQU             C7H
SFRAL           EQU             ACH
SFRAH           EQU             ADH
SFRFD           EQU             AEH
SFRCN           EQU             AFH

ORG 000H
LJMP 100H                               ; JUMP TO MAIN PROGRAM
;*****************************************************************************
;* 1. TIMER0 SERVICE VECTOR ORG = 0BH
;*****************************************************************************
ORG 000B

CLR   TR0                               ; TR0 = 0, STOP TIMER0
MOV TL0, R6
MOV TH0, R7
RETI
;*****************************************************************************
;*  4KB LDFlash MAIN PROGRAM
;*****************************************************************************
ORG 100H

MAIN_4K:
MOV TA, #AAH
MOV TA, #55H
MOV CHPCON, #03H                        ; CHPCON = 03H, ENABLE IN-SYSTEM PROGRAMMING.

MOV SFRCN, #0H
MOV TCON, #00H                          ; TCON = 00H, TR = 0 TIMER0 STOP
MOV TMOD, #01H                          ; TMOD = 01H, SET TIMER0 A 16BIT TIMER
MOV IP, #00H                            ; IP = 00H
MOV IE, #82H                            ; IE = 82H, TIMER0 INTERRUPT ENABLED
MOV R6, #F0H
MOV R7, #FFH
MOV TL0, R6
MOV TH0, R7
MOV TCON, #10H                          ; TCON = 10H, TR0 = 1, GO
MOV PCON, #01H                          ; ENTER IDLE MODE

UPDATE_APFlash:
MOV TCON, #00H                          ; TCON = 00H, TR = 0 TIM0 STOP
MOV IP, #00H                            ; IP = 00H
MOV IE, #82H                            ; IE = 82H, TIMER0 INTERRUPT ENABLED
```

```
READ_VERIFY_APFlash:
        MOV SFRAL,R2                ; SFRAL = LOW ADDRESS
        MOV TCON,#10H               ; TCON = 10H, TR0 = 1,GO
        MOV PCON,#01H
        INC R2
        MOVX A,@DPTR
        INC DPTR
        CJNE A,SFRFD,ERROR_APFlash
        CJNE R2,#0H,READ_VERIFY_APFlash
        INC R1
        MOV SFRAH,R1
        CJNE R1,#0H,READ_VERIFY_APFlash

;*******************************************************************************
;* PROGRAMMING COMPLETLY, SOFTWARE RESET CPU
;*******************************************************************************
        MOV TA, #AAH
        MOV TA, #55H
        MOV CHPCON, #83H            ; SOFTWARE RESET. CPU will restart from APFlash0

ERROR_APFlash:
        DJNZ R4, UPDATE_APFlash     ; IF ERROR OCCURS, REPEAT 3 TIMES.
.                                   ; IN-SYST PROGRAMMING FAIL, USER'S PROCESS TO
DEAL WITH IT.
                        .
                        .
                        .
```