**מדינת ישראל**
משרד החינוך

סוג הבחינה: בגרות לבתי־ספר על־יסודיים
מועד הבחינה: אביב תשפ"ד, 2024
סמל השאלון: 711911

**אין להעביר את הנוסחאון לנבחן אחר**

# נוסחאון במיקרו־בקר C8051F380 לכיתה י"ג (37 עמודים)

במטרה לשמור אחידות במרכיבים הפנימיים של הבקר, מיפוי הרגליים, האוגרים הפנימיים, שמות האוגרים והמשתנים, שאלות המבחן והנתונים במסמך זה מותאמים למיקרו־בקר C8051F380 ממשפחת Silicon Labs המבוסס על הבקר 8051 .

דף הנוסחאות מותאם למהדר Keil uVision . חלקים מדף נוסחאות זה מתאימים גם למהדרים אחרים.

שמירה על צורה אחידה למתן שמות אוגרים וטיפוסים של משתנים, יש להצהיר על ספריות הבאות:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
```

פונקצייה לאתחול המעבד:

```
void Init_Device(void);
```

אין צורך לכתוב את מילות האתחול לאוגרים לביטול ה־watchdog וקביעת תדר הבקר, אלא להניח שהן קיימות כחלק ממימוש הקוד בפונקצייה Init_Device .

יש להניח שפונקציות השהייה במילי־שניות ומיקרו־שניות קיימות ואין צורך להצהיר עליהן או לממש אותן.

```
void delay_ms(unsigned int ms);
void delay_us(unsigned int us);
```

המשך בעמוד 2

**Data Types (טיפוסי נתונים)**

| Name | Description | תיאור | Size | Range |
|---|---|---|---|---|
| bit | One Bit | ביט בודד | 1 bit | 0 to 1 |
| char<br>or<br>S8 | Character or small integer. | תו בודד או בייט | 1 byte | –128 to 127 |
| unsigned char<br>or<br>U8 | Unsigned small integer | בייט אחד ללא סימן | 1 byte | 0 to 255 |
| int<br>or<br>S16 | Integer | מספר שלם | 2 bytes | –32768 to 32767 |
| unsigned int<br>or<br>U16 | Unsigned integer | מספר שלם ללא סימן | 2 bytes | 0 to 65535 |
| long<br>or<br>S32 | 32-bit integer | מספר שלם ארוך | 4 bytes | –2147483648 to 2147483647 |
| unsigned long<br>or<br>U32 | 32-bit Unsigned integer | מספר שלם ארוך ללא סימן | 4 bytes | 0 to 4294967295 |
| float | Floating point number | מספר ממשי | 4 bytes | +/– 1.175494E–38 to<br>+/– 3.402823E+38 |
| sbit | Special Bit | ביט מיוחד | 1 bit | 0 to 1 |
| sfr | Special Function Registers | בית מיוחד | 1 byte | 0 to 255 |
| sfr16 | 16 bits special Function Registers | בית כפול מיוחד | 2 bytes | 0 to 65535 |

דוגמאות:

```
unsigned char a;
int b, c;
sfr P1=0x90;
sbit P1-7 = P1^7;
```

המשך בעמוד 3

**Memory Areas (גישה לזיכרון)**

| Name | Description | Example |
|---|---|---|
| data | places the variable in directly addressable RAM in the micro core | unsigned int data dnum; |
| xdata | places the variable in external RAM | unsigned char xdata xnum_at_0x8000; * |
| idata | places the variable in indirectly addressable memory within the micro core | int idata inum; |
| code | places the variable in program memory | unsigned char code cnum=0xAA; |

* _at_ – places the variable in absolute address.

**Preprocessor-directives (הנחיות לקדם־מהדר)**

| Description | Syntax | Example |
|---|---|---|
| macro definitions | #define identifier replacement | #define LED P 1_7 |

identifier - מזהה ; replacement - תחליף

**Operators (אופרטורים)**

| Description | תיאור | Operator |
|---|---|---|
| Assignment | השמה | = |

**Initalization of variables (אתחול משתנים)**

```
unsigned char  d=0;
d=75;        // decimal  number
d=0x4B;      // hexadecimal  number
```

**Arithmetic Operators (אופרטורים חשבוניים)**

| Description | תיאור | Operator |
|---|---|---|
| addition | חיבור | + |
| subtraction | חיסור | − |
| multiplication | כפל | * |
| division | חילוק | / |
| modulo | שארית | % |

המשך בעמוד 4

**Relational and equality operators (אופרטורים להשוואה ויחסים)**

| Description | תיאור | Operator |
|---|---|---|
| Equal to | שווה | = = |
| Not equal to | שונה | ! = |
| Greater than | גדול מ־ | > |
| Less than | קטן מ־ | < |
| Greater than or equal to | גדול מ־/ שווה | > = |
| Less than or equal to | קטן מ־/ שווה | < = |

**Logical operators (אופרטורים לוגיים בין ביטויים)**

| Description | תיאור | Operator |
|---|---|---|
| NOT | היפוך | ! |
| AND | וגם | && |
| OR | או | \| \| |

**Bitwise Operators (אופרטורים על סיביות)**

| Description | תיאור | Operator |
|---|---|---|
| AND | וגם | & |
| Inclusive OR | או כולל | \| |
| Exclusive OR (XOR) | או מוציא | ^ |
| Byte inversion | היפוך בית | ~ |
| Bit inversion | היפוך סיבית | ! |
| Shift Left | הזזה שמאלה | << |
| Shift Right | הזזה ימינה | >> |

המשך בעמוד 5

**Conditional Structures (מבני בקרה – משפטי תנאי)**

| Description | Syntax | Example |
|---|---|---|
| if | if (condition)<br>statements | if (d == 100) {<br>P1 = 0xFF;<br>} |
| if else | if (condition)<br>statement1<br>else<br>statement2 | if (d == 100)<br>P1 = 0xFF;<br>else<br>P1=0; |
| if else if else | if (condition)<br>statement1<br>else if (condition)<br>statement2<br>else<br>statement3 | if (d > 0)<br>P1=4;<br>else if (d < 0)<br>P1=2;<br>else<br>P1=1; |

* condition (תנאי)  * statement (הצהרה)

**Iteration Structures (מבני בקרה - לולאות)**

| Description | Syntax | Example |
|---|---|---|
| while loop | while (expression) statements | while (n>0) {<br>P1=n;<br>n– –;<br>} |
| do-while loop | do statements while (condition); | do {<br>n=P1;<br>} while (n != 0); |
| for loop | for (initialization; condition;<br>increase) statements ; | for (i=0; i<10; i++) {<br>P1=i;<br>} |

* condition (תנאי)  * statement (הצהרה)

המשך בעמוד 6

**Arrays (מערכים)**

| Description | Syntax | Example |
|---|---|---|
| הגדרת מערך חד מימדי<br>0 1 2 3 4<br>arr<br>char | type name [elements]; | char arr[5]; |
| אתחול והצבת ערכים במערך<br>0 1 2 3 4<br>arr 3 5 7 –1 14<br>int | type name [elements] =<br>{value1,..valueu}; | int arr[5] =<br>{3,5,7,–1, 14}; |
| הגדרת מערך דו מימדי<br>0 1 2 3 4<br>arr 0 1 2<br>arr [1] [3] | type name [elements, elements]; | int arr[3][5]; |

* elements (פרטים)  * value (ערך)

**Structure of a program (מבנה כללי של תוכנית)**

```
#include "compiler_defs.h"

#include "C8051F380_defs.h"


void Init_Device(void);

void delay_ms(unsigned int ms);

void delay_us(unsigned int us);
void main()


{

        Init_Device();

        while(1)

        {

                //Your code.

        }
```

המשך בעמוד 7

**Functions (פונקציות)**

| Description | Syntax | Example |
|---|---|---|
| Functions with no type and no argument | void name (void)<br>{<br>statements;<br>}| void OutData(void)<br>{<br>P1=0xAB;<br>}<br>void main()<br>{<br>OutData();<br>} |
| Functions with no type | void name ( parameter1, parameter2, ...)<br>{<br>statements;<br>} | void OutData(unsigned char a, unsigned char b)<br>{<br>P0=a;<br>P1=b;<br>}<br>void main()<br>{<br>OutData(0xF,0xF0);<br>} |
| Functions with type and argument | type name ( parameter1, parameter2, ...)<br>{<br>statements;<br>} | unsigned char InOutData(unsigned char a)<br>{<br>P1=a;<br>return P0;<br>}<br>void main()<br>{<br>unsigned char r;<br>r = InOutData(63);<br>P2=r;<br>} |

parameter – ערך המועבר לפונקציה ; statement – הצהרה ; type – טיפוס ; argument – טיעון

המשך בעמוד 8

**Interrput Service Routines (שגרות לטיפול בפסיקות)**

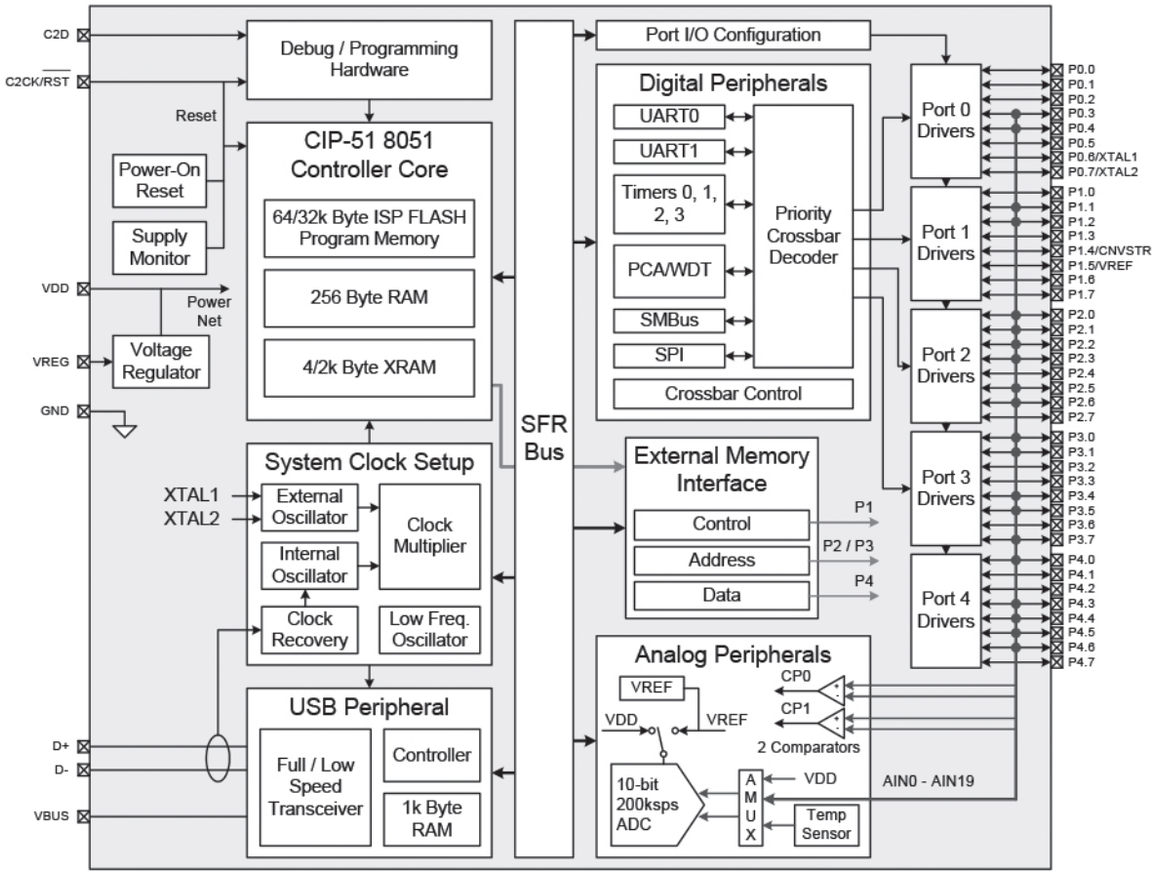
| Interrupt Source | Interrupt Vector | Priority Order | Pending Flag | Bit addressable? | Cleared by HW? | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|
| Reset | 0x0000 | Top | None | N/A | N/A | Always Enabled | Always Highest |
| External Interrupt 0 ($\overline{INT0}$) | 0x0003 | 0 | IE0 (TCON.1) | Y | Y | EX0 (IE.0) | PX0 (IP.0) |
| Timer 0 Overflow | 0x000B | 1 | TF0 (TCON.5) | Y | Y | ET0 (IE.1) | PT0 (IP.1) |
| External Interrupt 1 ($\overline{INT1}$) | 0x0013 | 2 | IE1 (TCON.3) | Y | Y | EX1 (IE.2) | PX1 (IP.2) |
| Timer 1 Overflow | 0x001B | 3 | TF1 (TCON.7) | Y | Y | ET1 (IE.3) | PT1 (IP.3) |
| UART0 | 0x0023 | 4 | RI0 (SCON0.0)<br>TI0 (SCON0.1) | Y | N | ES0 (IE.4) | PS0 (IP.4) |
| Timer 2 Overflow | 0x002B | 5 | TF2H (TMR2CN.7)<br>TF2L (TMR2CN.6) | Y | N | ET2 (IE.5) | PT2 (IP.5) |
| SPI0 | 0x0033 | 6 | SPIF (SPI0CN.7)<br>WCOL (SPI0CN.6)<br>MODF (SPI0CN.5)<br>RXOVRN (SPI0CN.4) | Y | N | ESPI0 (IE.6) | PSPI0 (IP.6) |
| SMB0 | 0x003B | 7 | SI (SMB0CN.0) | Y | N | ESMB0 (EIE1.0) | PSMB0 (EIP1.0) |
| USB0 | 0x0043 | 8 | Special | N | N | EUSB0 (EIE1.1) | PUSB0 (EIP1.1) |
| ADC0 Window Compare | 0x004B | 9 | AD0WINT (ADC0CN.3) | Y | N | EWADC0 (EIE1.2) | PWADC0 (EIP1.2) |
| ADC0 Conversion Complete | 0x0053 | 10 | AD0INT (ADC0CN.5) | Y | N | EADC0 (EIE1.3) | PADC0 (EIP1.3) |

דוגמה:

```
void int2sub () interrupt 2 {
      // Interrupt code
}
```

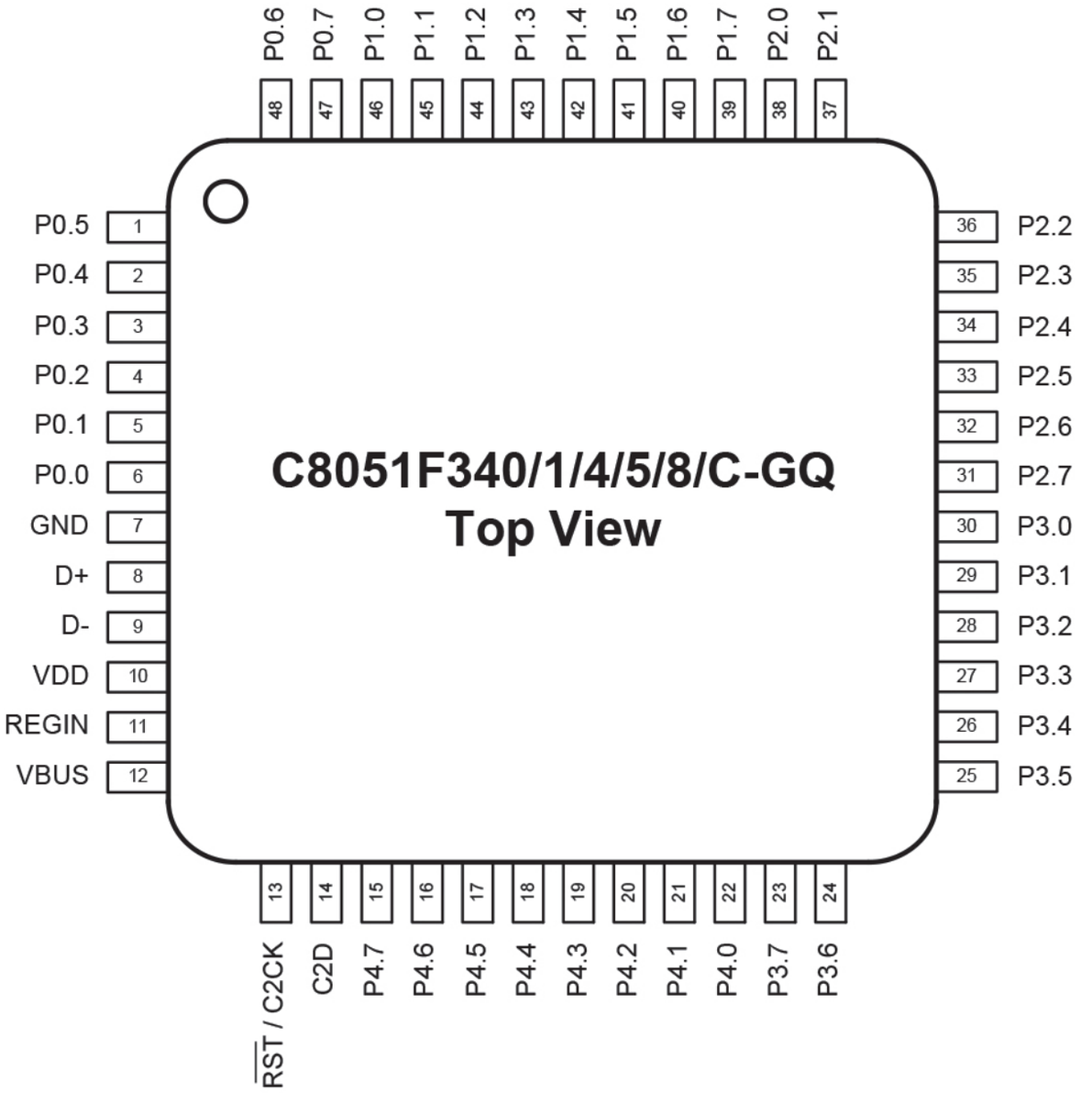
המשך בעמוד 9

**C8051F380 Block Diagram**



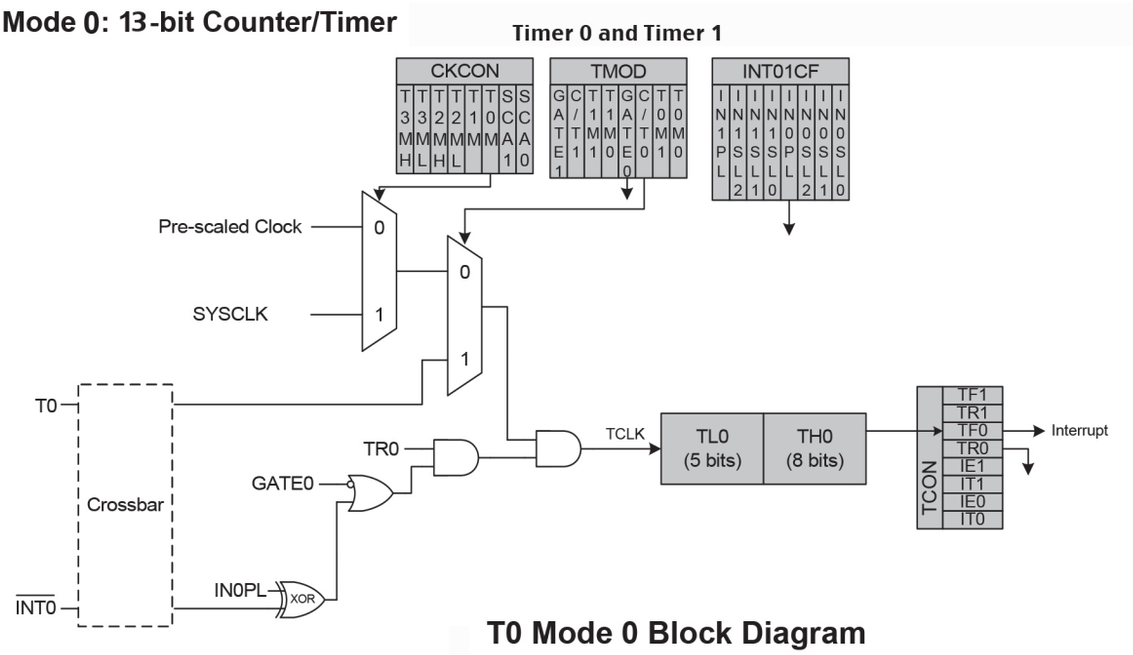

המשך בעמוד 10

**TQFP – 48 Pinout Diagram (Top View)**




המשך בעמוד 11

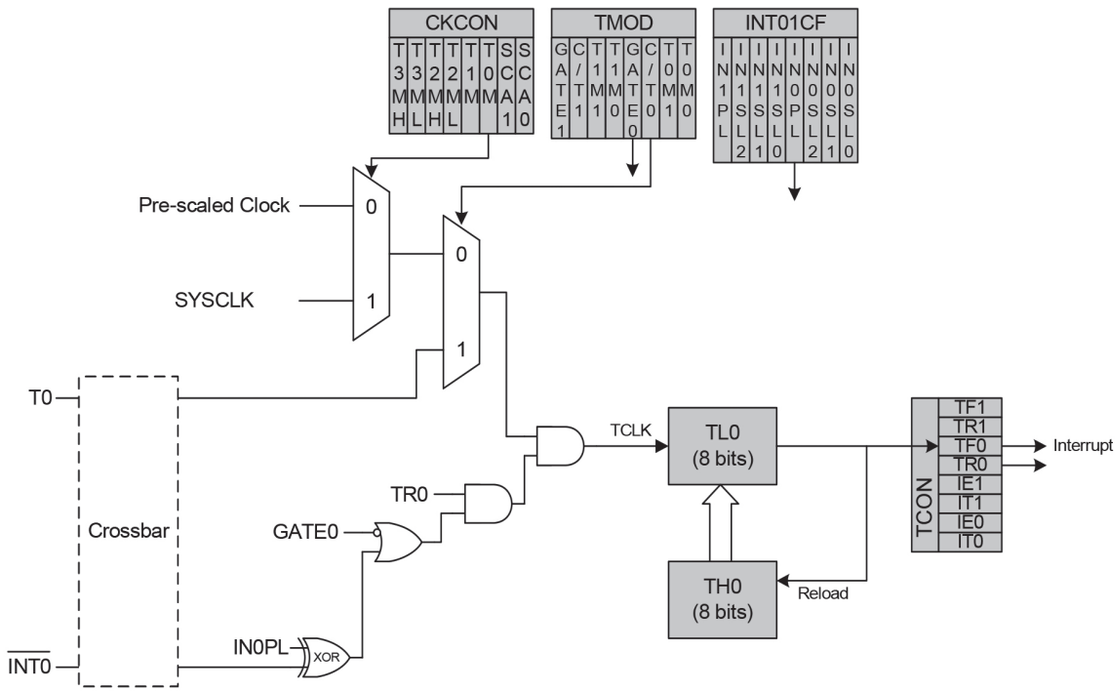
## Mode 0: 13-bit Counter/Timer




**T0 Mode 0 Block Diagram**

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

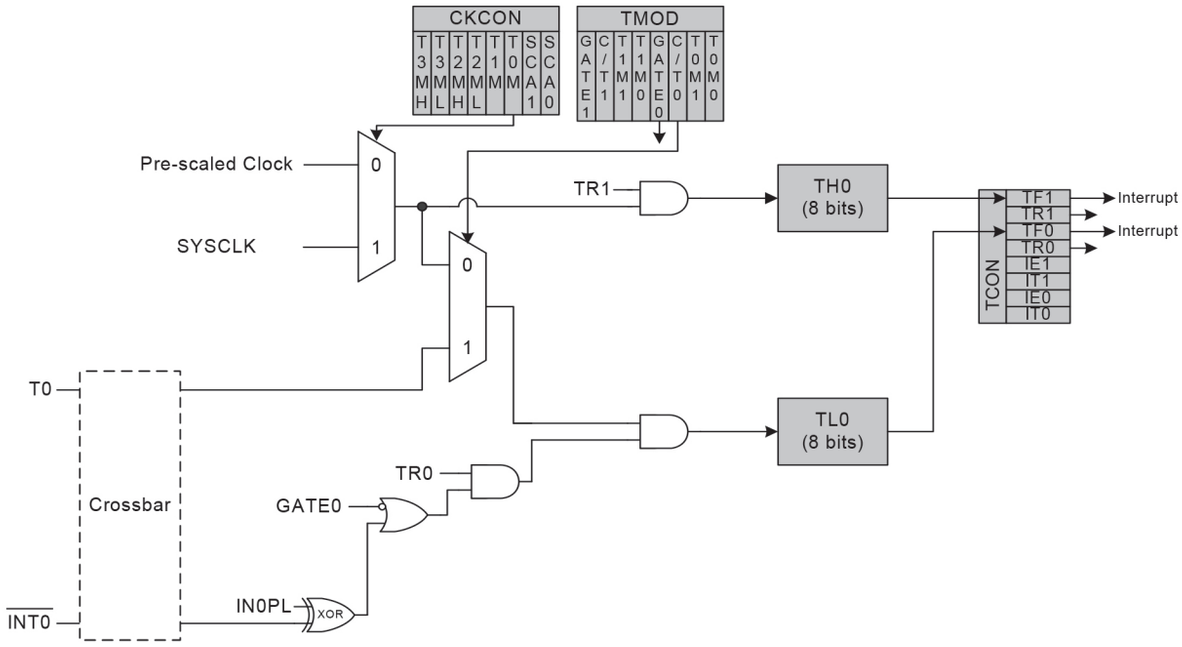
## Mode 2: 8-bit Counter/Timer with Auto-Reload




**T0 Mode 2 Block Diagram**

המשך בעמוד 12

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




**T0 Mode 3 Block Diagram**

המשך בעמוד 13

**CKCON: Clock Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| T3MH | T3ML | T2MH | T2ML | T1M | T0M | SCA1 | SCA0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x8E |

Bit7: T3MH: Timer 3 High Byte Clock Select.
This bit selects the clock supplied to the Timer 3 high byte if Timer 3 is configured in split 8-bit timer mode. T3MH is ignored if Timer 3 is in any other mode.
0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 high byte uses the system clock.

Bit6: T3ML: Timer 3 Low Byte Clock Select.
This bit selects the clock supplied to Timer 3. If Timer 3 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 low byte uses the system clock.

Bit5: T2MH: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit4: T2ML: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Timer 1 uses the system clock.

Bit2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic 1.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Counter/Timer 0 uses the system clock.

Bits1–0: SCA1-SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

* parameter (ערך המועבר לפונקציה)  * statement (הצהרה)

המשך בעמוד 14

**TMOD: Timer Mode**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| GATE1 | C/T1 | T1M1 | T1M0 | GATE0 | C/T0 | T0M1 | T0M0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x89 |

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{INT1}$ logic level.
1: Timer 1 enabled only when TR1 = 1 AND $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.3).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

| T1M1 | T1M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 13-bit counter/timer |
| 0 | 1 | Mode 1: 16-bit counter/timer |
| 1 | 0 | Mode 2: 8-bit counter/timer with auto-reload |
| 1 | 1 | Mode 3: Timer 1 inactive |

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{INT0}$ logic level.
1: Timer 0 enabled only when TR0 = 1 AND $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.2).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

| T0M1 | T0M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 13-bit counter/timer |
| 0 | 1 | Mode 1: 16-bit counter/timer |
| 1 | 0 | Mode 2: 8-bit counter/timer with auto-reload |
| 1 | 1 | Mode 3: Two 8-bit counter/timers |

המשך בעמוד 15

**TCON: Timer Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0x88 |

Bit7: TF1: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: TF0: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: IE1: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to '1' when $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured $\overline{INT1}$ interrupt will be edge or level sensitive. $\overline{INT1}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 9.13).
0: $\overline{INT1}$ is level triggered.
1: $\overline{INT1}$ is edge triggered.

Bit1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to '1' when $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured $\overline{INT0}$ interrupt will be edge or level sensitive. $\overline{INT0}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 9.13).
0: $\overline{INT0}$ is level triggered.
1: $\overline{INT0}$ is edge triggered.

המשך בעמוד 16

## SFR Definition 21.4. TL0: Timer 0 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x8A |

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

## SFR Definition 21.5. TL1: Timer 1 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x8B |

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

## SFR Definition 21.6. TH0: Timer 0 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x8C |

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

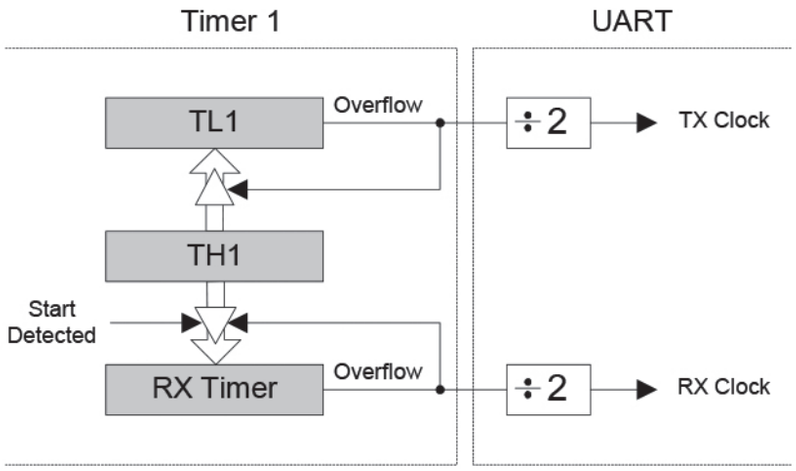
## SFR Definition 21.7. TH1: Timer 1 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x8D |

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

המשך בעמוד 17

**UART0**

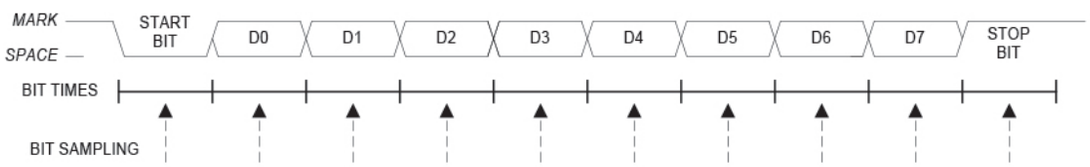


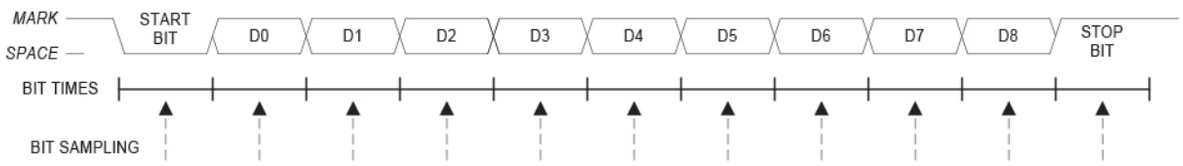

**UARTO Baud Rate Logic**

דוגמה:

A) $$\text{UARTBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

B) $$\text{T1\_Overflow\_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$




**9-Bit UART Timing Diagram**

המשך בעמוד 18

**SCON0: Serial Port 0 Control**

| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| S0MODE | - | MCE0 | REN0 | TB80 | RB80 | TI0 | RI0 | 01000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x98

Bit7: S0MODE: Serial Port 0 Operation Mode.
This bit selects the UART0 Operation Mode.
0: 8-bit UART with Variable Baud Rate.
1: 9-bit UART with Variable Baud Rate.

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: MCE0: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
S0MODE = 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level 1.
S0MODE = 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI0: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

המשך בעמוד 19

**SBUF0: Serial (UART0) Port Data Buffer**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x99

Bits7–0: SBUF0[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

| | Target Baud Rate (bps) | Actual Baud Rate (bps) | Baud Rate Error | Oscillator Divide Factor | Timer Clock Source | SCA1-SCA0 (pre-scale select* | T1M* | Timer 1 Reload Value (hex) |
|---|---|---|---|---|---|---|---|---|
| SYSCLK = 12 MHz | 230400 | 230769 | 0.16% | 52 | SYSCLK | XX | 1 | 0xE6 |
| | 115200 | 115385 | 0.16% | 104 | SYSCLK | XX | 1 | 0xCC |
| | 57600 | 57692 | 0.16% | 208 | SYSCLK | XX | 1 | 0x98 |
| | 28800 | 28846 | 0.16% | 416 | SYSCLK | XX | 1 | 0x30 |
| | 14400 | 14423 | 0.16% | 832 | SYSCLK / 4 | 01 | 0 | 0x98 |
| | 9600 | 9615 | 0.16% | 1248 | SYSCLK / 4 | 01 | 0 | 0x64 |
| | 2400 | 2404 | 0.16% | 4992 | SYSCLK / 12 | 00 | 0 | 0x30 |
| | 1200 | 1202 | 0.16% | 9984 | SYSCLK / 48 | 10 | 0 | 0x98 |
| SYSCLK = 24 MHz | 230400 | 230769 | 0.16% | 104 | SYSCLK | XX | 1 | 0xCC |
| | 115200 | 115385 | 0.16% | 208 | SYSCLK | XX | 1 | 0x98 |
| | 57600 | 57692 | 0.16% | 416 | SYSCLK | XX | 1 | 0x30 |
| | 28800 | 28846 | 0.16% | 832 | SYSCLK / 4 | 01 | 0 | 0x98 |
| | 14400 | 14423 | 0.16% | 1664 | SYSCLK / 4 | 01 | 0 | 0x30 |
| | 9600 | 9615 | 0.16% | 2496 | SYSCLK / 12 | 00 | 0 | 0x98 |
| | 2400 | 2404 | 0.16% | 9984 | SYSCLK / 48 | 10 | 0 | 0x98 |
| | 1200 | 1202 | 0.16% | 19968 | SYSCLK / 48 | 10 | 0 | 0x30 |
| SYSCLK = 48 MHz | 230400 | 230769 | 0.16% | 208 | SYSCLK | XX | 1 | 0x98 |
| | 115200 | 115385 | 0.16% | 416 | SYSCLK | XX | 1 | 0x30 |
| | 57600 | 57692 | 0.16% | 832 | SYSCLK / 4 | 01 | 0 | 0x98 |
| | 28800 | 28846 | 0.16% | 1664 | SYSCLK / 4 | 01 | 0 | 0x30 |
| | 14400 | 14388 | 0.08% | 3336 | SYSCLK / 12 | 00 | 0 | 0x75 |
| | 9600 | 9615 | 0.16% | 4992 | SYSCLK / 12 | 00 | 0 | 0x30 |
| | 2400 | 2404 | 0.16% | 19968 | SYSCLK / 48 | 10 | 0 | 0x30 |

המשך בעמוד 20

**IE: Interrupt Enable**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| EA | ESPI0 | ET2 | ES0 | ET1 | EX1 | ET0 | EX0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xA8 |

Bit7: EA: Enable All Interrupts.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit6: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit5: ET2: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.

Bit1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

המשך בעמוד 21

**TCON: Timer Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0x88 |

Bit7: TF1: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: TF0: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: IE1: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to '1' when $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured $\overline{INT1}$ interrupt will be edge or level sensitive. $\overline{INT1}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 9.13).
0: $\overline{INT1}$ is level triggered.
1: $\overline{INT1}$ is edge triggered.

Bit1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to '1' when $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured $\overline{INT0}$ interrupt will be edge or level sensitive. $\overline{INT0}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 9.13).
0: $\overline{INT0}$ is level triggered.
1: $\overline{INT0}$ is edge triggered.

המשך בעמוד 23

**IT01CF: $\overline{\text{INT0}}$ / $\overline{\text{INT1}}$ Configuration**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| IN1PL | IN1SL2 | IN1SL1 | IN1SL0 | IN0PL | IN0SL2 | IN0SL1 | IN0SL0 | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xE4 |

Note: Refer to SFR Definition 21.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: $\overline{\text{INT1}}$ Polarity
0: $\overline{\text{INT1}}$ input is active low.
1: $\overline{\text{INT1}}$ input is active high.

Bits6–4: IN1SL2–0: $\overline{\text{INT1}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

| IN1SL2–0 | $\overline{\text{INT1}}$ Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

Bit3: IN0PL: $\overline{\text{INT0}}$ Polarity
0: $\overline{\text{INT0}}$ interrupt is active low.
1: $\overline{\text{INT0}}$ interrupt is active high.

Bits2–0: INT0SL2–0: $\overline{\text{INT0}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar. $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

| IN0SL2–0 | $\overline{\text{INT0}}$ Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

**IP: Interrupt Priority**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | PSPI0 | PT2 | PS0 | PT1 | PX1 | PT0 | PX0 | 10000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xB8 |

Bit7: UNUSED. Read = 1, Write = don't care.
Bit6: PSPI0: Serial Peripheral Interface (SPI0) Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority level.
1: SPI0 interrupt set to high priority level.
Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority level.
1: Timer 2 interrupts set to high priority level.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupts set to high priority level.
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupts set to high priority level.
Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.
Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.
Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

המשך בעמוד 24

**Interrput Service Routines (שגרות לטיפול בפסיקות)**

| Interrupt Source | Interrupt Vector | Priority Order | Pending Flag | Bit Address? | Cleared by HW? | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|
| Reset | 0×0000 | Top | None | N/A | N/A | Always Enabled | Always Highest |
| External Interrupt 0 ($\overline{INT0}$) | 0×0003 | 0 | IE0(TCON.1) | Y | Y | EX0(IE.0) | P×0(IP.0) |
| Timer 0 Overflow | 0×000B | 1 | TF0(TCON.5) | Y | Y | ET0(IE.1) | PT0(IP.1) |
| External Interrupt 1 ($\overline{INT1}$) | 0×0013 | 2 | IE1(TCON.3) | Y | Y | EX1(IE.2) | PX1(IP.2) |
| Timer 1 Overflow | 0×001B | 3 | TF1(TCON.7) | Y | Y | ET1(IE.3) | PT1(IP.3) |
| UART0 | 0×0023 | 4 | RI0 (SCON.0)<br>TI0 (SCON.1) | Y | N | ES0(IE.4) | PS0(IP.4) |
| Timer 2 Overflow | 0×002B | 5 | TF2H(TMR2CN.7)<br>TF2L(TMR2CN.6) | Y | N | ET2(IE.5) | PT2(IP.5) |
| SPI0 | 0×0033 | 6 | SPIF (SPI0CN.7)<br>WCOL (SPI0CN.6)<br>MODF (SPI0CN.5)<br>RXOVRN (SPI0CN.4) | Y | N | ESPI0(IE.6) | PSPI0(IP.6) |
| SMB0 | 0×003B | 7 | SI (SMB0CN.0) | Y | N | ESMB0 (EIE1.0) | PSMB0 (EIP1.0) |
| USB0 | 0×0043 | 8 | Special | N | N | EUSB0 (EIE1.1) | PUSB0 (EIP1.1) |
| ADC0 Window Com-pare | 0×004B | 9 | AD0WINT (ADC0CN.3) | Y | N | EWADC0 (EIE1.2) | PWADC0 (EIP1.2) |
| ADC0 Conversion Complete | 0×0053 | 10 | AD0INT (ADC0CN.5) | Y | N | EADC0 (EIE1.3) | PADC0 (EIP1.3) |

דוגמה:

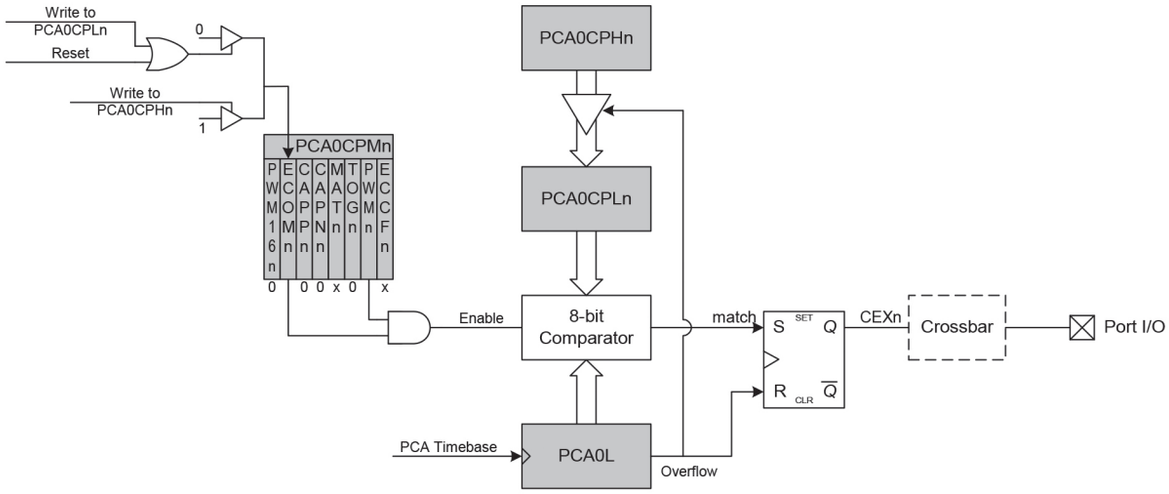
```
void int2sub () interrupt 2 {
      // Interrupt code
}
```

המשך בעמוד 25

### Bit Pulse Width Modulator Mode

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

**Equation 8-Bit PWM Duty Cycle**

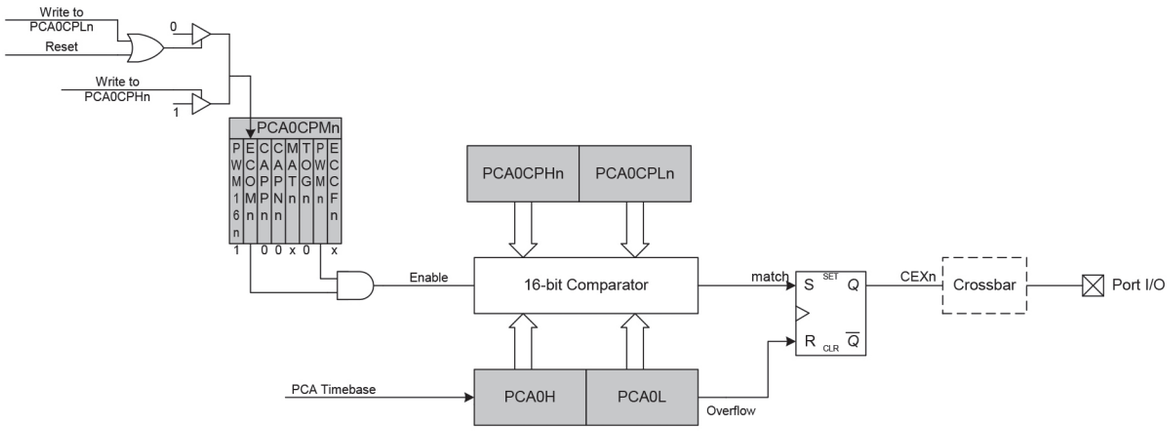



**PCA 8-Bit PWM Mode Diagram**

### 16-Bit Pulse Width Modulator Mode

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

**Equation 16-Bit PWM Duty Cycle**




**PCA 16-Bit PWM Mode**

**PCA0CN: PCA Control**

המשך בעמוד 26

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CF | CR | - | CCF4 | CCF3 | CCF2 | CCF1 | CCF0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xD8 |

Bit7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: CR: PCA Counter/Timer Run Control.
This bit enables/disables the PCA Counter/Timer.
0: PCA Counter/Timer disabled.
1: PCA Counter/Timer enabled.

Bit5: UNUSED. Read = 0b, Write = don't care.

Bit4: CCF4: PCA Module 4 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF4 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit3: CCF3: PCA Module 3 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF3 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: CCF2: PCA Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: CCF1: PCA Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: CCF0: PCA Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

המשך בעמוד 27

**PCA0CPM Bit Setting for PCA Capture/Compare Modules**

| PWM16 | ECOM | CAPP | CAPN | MAT | TOG | PWM | ECCF | Operation Mode |
|---|---|---|---|---|---|---|---|---|
| X | X | 1 | 0 | 0 | 0 | 0 | X | Capture triggered by positive edge on CEXn |
| X | X | 0 | 1 | 0 | 0 | 0 | X | Capture triggered by negative edge on CEXn |
| X | X | 1 | 1 | 0 | 0 | 0 | X | Capture triggered by transition on CEXn |
| X | 1 | 0 | 0 | 1 | 0 | 0 | X | Software Timer |
| X | 1 | 0 | 0 | 1 | 1 | 0 | X | High Speed Output |
| X | 1 | 0 | 0 | X | 1 | 1 | X | Frequency Output |
| 0 | 1 | 0 | 0 | X | 0 | 1 | X | 8-Bit Pulse Width Modulator |
| 1 | 1 | 0 | 0 | X | 0 | 1 | X | 16-Bit Pulse Width Modulator |

## SFR Definition 22.6. PCA0CPLn: PCA Capture Module Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xFB, 0xE9, 0xEB, 0xED, 0xFD |

PCA0CPLn Address: PCA0CPL0 = 0xFB (n = 0), PCA0CPL1 = 0xE9 (n = 1),
PCA0CPL2 = 0xEB (n = 2), PCA0CPL3 = 0xED (n = 3),
PCA0CPL4 = 0xFD (n = 4)

Bits7–0: PCA0CPLn: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

## SFR Definition 22.7. PCA0CPHn: PCA Capture Module High Byte

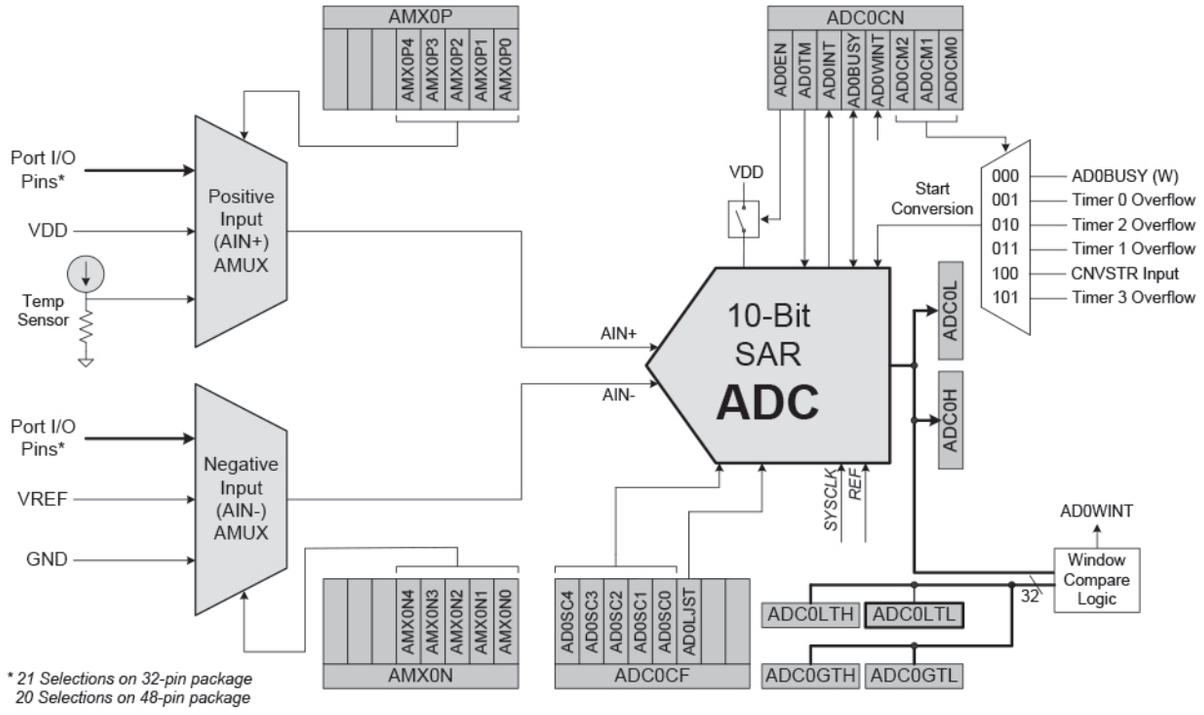
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xFC, 0xEA, 0xEC,0xEE, 0xFE |

PCA0CPHn Address: PCA0CPH0 = 0xFC (n = 0), PCA0CPH1 = 0xEA (n = 1),
PCA0CPH2 = 0xEC (n = 2), PCA0CPH3 = 0xEE (n = 3),
PCA0CPH4 = 0xFE (n = 4)

Bits7–0: PCA0CPHn: PCA Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

המשך בעמוד 28

10–Bit ADC




ADCO Functional Block Diagram

המשך בעמוד 29

**ADC0CF: ADC0 Configuration**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x99

Bits7–0: SBUF0[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

SFR Address = 0xBC; SFR Page = All Pages

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AD0LJST | - | - | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xBC |

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4-0. SAR Conversion clock requirements are given in Table 5.1.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bit2: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.

Bits1–0: UNUSED. Read = 00b; Write = don't care.

המשך בעמוד 30

**ADC0CN: ADC0 Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0EN | AD0TM | AD0INT | AD0BUSY | AD0WINT | AD0CM2 | AD0CM1 | AD0CM0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xE8 |

Bit7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: AD0TM: ADC0 Track Mode Bit.
0: Normal Track Mode: When ADC0 is enabled, tracking is continuous unless a conversion is in progress.
1: Low-power Track Mode: Tracking Defined by AD0CM2-0 bits (see below).

Bit5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
0: ADC0 has not completed a data conversion since the last time AD0INT was cleared.
1: ADC0 has completed a data conversion.

Bit4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
1: ADC0 conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM2-0 = 000b

Bit3: AD0WINT: ADC0 Window Compare Interrupt Flag.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bits2–0: AD0CM2–0: ADC0 Start of Conversion Mode Select.
When AD0TM = 0:
000: ADC0 conversion initiated on every write of '1' to AD0BUSY.
001: ADC0 conversion initiated on overflow of Timer 0.
010: ADC0 conversion initiated on overflow of Timer 2.
011: ADC0 conversion initiated on overflow of Timer 1.
100: ADC0 conversion initiated on rising edge of external CNVSTR.
101: ADC0 conversion initiated on overflow of Timer 3.
11x: Reserved.
When AD0TM = 1:
000: Tracking initiated on write of '1' to AD0BUSY and lasts 3 SAR clocks, followed by conversion.
001: Tracking initiated on overflow of Timer 0 and lasts 3 SAR clocks, followed by conversion.
010: Tracking initiated on overflow of Timer 2 and lasts 3 SAR clocks, followed by conversion.
011: Tracking initiated on overflow of Timer 1 and lasts 3 SAR clocks, followed by conversion.
100: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
101: Tracking initiated on overflow of Timer 3 and lasts 3 SAR clocks, followed by conversion.
11x: Reserved.

המשך בעמוד 31

## SFR Definition 5.1. AMX0P: AMUX0 Positive Channel Select

| R | R | R | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | AMX0P4 | AMX0P3 | AMX0P2 | AMX0P1 | AMX0P0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xBB |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–0: AMX0P4–0: AMUX0 Positive Input Selection

| AMX0P4-0 | ADC0 Positive Input (32-pin Package) | ADC0 Positive Input (48-pin Package) |
|---|---|---|
| 00000 | P1.0 | P2.0 |
| 00001 | P1.1 | P2.1 |
| 00010 | P1.2 | P2.2 |
| 00011 | P1.3 | P2.3 |
| 00100 | P1.4 | P2.5 |
| 00101 | P1.5 | P2.6 |
| 00110 | P1.6 | P3.0 |
| 00111 | P1.7 | P3.1 |
| 01000 | P2.0 | P3.4 |
| 01001 | P2.1 | P3.5 |
| 01010 | P2.2 | P3.7 |
| 01011 | P2.3 | P4.0 |
| 01100 | P2.4 | P4.3 |
| 01101 | P2.5 | P4.4 |
| 01110 | P2.6 | P4.5 |
| 01111 | P2.7 | P4.6 |
| 10000 | P3.0 | RESERVED |
| 10001 | P0.0 | P0.3 |
| 10010 | P0.1 | P0.4 |
| 10011 | P0.4 | P1.1 |
| 10100 | P0.5 | P1.2 |
| 10101 - 11101 | RESERVED | RESERVED |
| 11110 | Temp Sensor | Temp Sensor |
| 11111 | $V_{DD}$ | $V_{DD}$ |

המשך בעמוד 32

## SFR Definition 5.2. AMX0N: AMUX0 Negative Channel Select

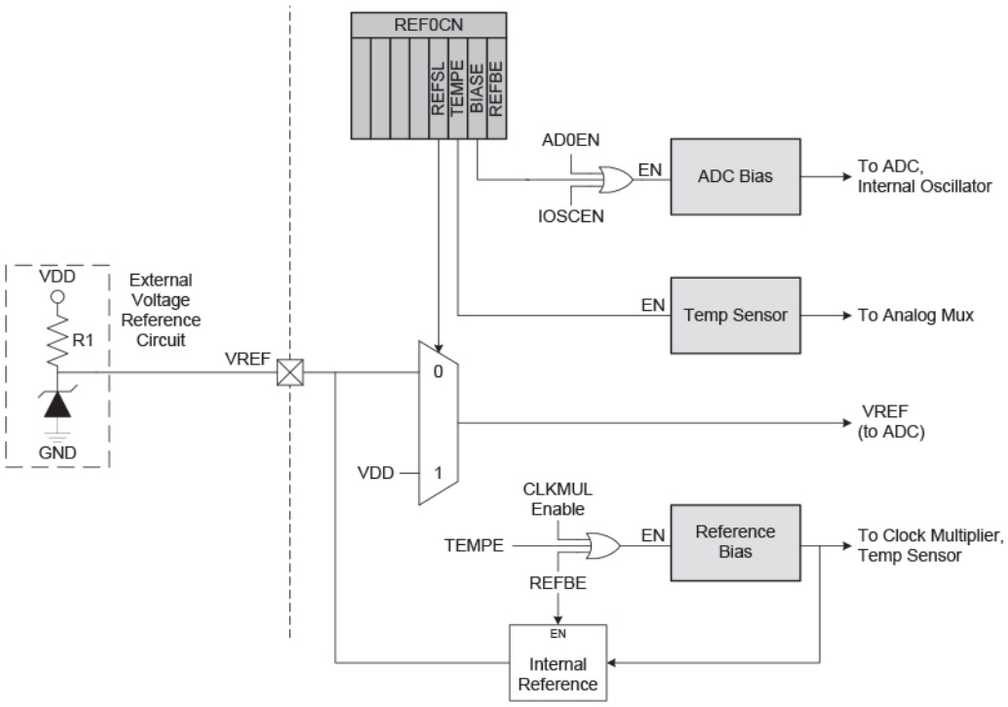
| R | R | R | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | AMX0N4 | AMX0N3 | AMX0N2 | AMX0N1 | AMX0N0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xBA |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–0: AMX0N4–0: AMUX0 Negative Input Selection.
Note that when GND is selected as the Negative Input, ADC0 operates in Single-ended mode. For all other Negative Input selections, ADC0 operates in Differential mode.

| AMX0N4-0 | ADC0 Negative Input (32-pin Package) | ADC0 Negative Input (48-pin Package) |
|---|---|---|
| 00000 | P1.0 | P2.0 |
| 00001 | P1.1 | P2.1 |
| 00010 | P1.2 | P2.2 |
| 00011 | P1.3 | P2.3 |
| 00100 | P1.4 | P2.5 |
| 00101 | P1.5 | P2.6 |
| 00110 | P1.6 | P3.0 |
| 00111 | P1.7 | P3.1 |
| 01000 | P2.0 | P3.4 |
| 01001 | P2.1 | P3.5 |
| 01010 | P2.2 | P3.7 |
| 01011 | P2.3 | P4.0 |
| 01100 | P2.4 | P4.3 |
| 01101 | P2.5 | P4.4 |
| 01110 | P2.6 | P4.5 |
| 01111 | P2.7 | P4.6 |
| 10000 | P3.0 | RESERVED |
| 10001 | P0.0 | P0.3 |
| 10010 | P0.1 | P0.4 |
| 10011 | P0.4 | P1.1 |
| 10100 | P0.5 | P1.2 |
| 10101 - 11101 | RESERVED | RESERVED |
| 11110 | VREF | VREF |
| 11111 | GND (Single-Ended Mode) | GND (Single-Ended Mode) |

**Voltage Reference Option**

המשך בעמוד 33




**Voltage Reference Functional Block Diagram**

**דוגמאות למתחי ייחוס**

- מתח ייחוס של VDD = 3.3V — REFOCN = 0×08;
- מתח ייחוס פנימי של 1.8V — REFOCN = 0×10;
- מתח VREF חיצוני דרך הדק P1.5 — REFOCN = 0×00;
- מתח פנימי של (1×Gain) — REFOCN = 0×81;
  או 2.4v (2×Gain) — REFOCN = 0×01;
  (הדק P1.5 צריך להיות פנוי)

המשך בעמוד 34

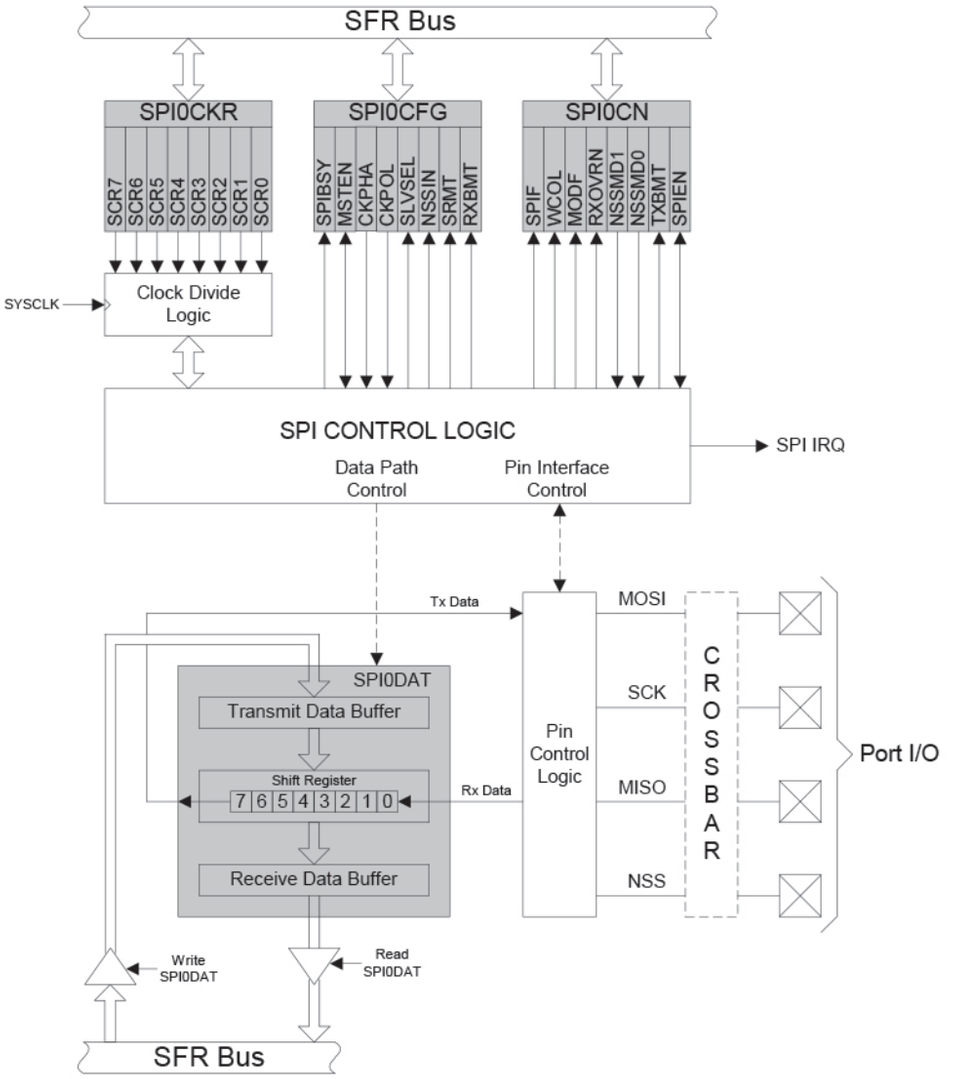
**REF0CN: Reference Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | REFSL | TEMPE | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xD1 |

Bits7–3: UNUSED. Read = 00000b; Write = don't care.
Bit3: REFSL: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: VREF pin used as voltage reference.
1: $V_{DD}$ used as voltage reference.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit1: BIASE: Internal Analog Bias Generator Enable Bit.
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer disabled.
1: Internal Reference Buffer enabled. Internal voltage reference driven on the VREF pin.

המשך בעמוד 35

**Enhanced Serial Peripheral Interface (SPI0)**




**SPI Block Diagram**

המשך בעמוד 36

**SPI0CFG: SPI0 Configuration**

| R | R/W | R/W | R/W | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SPIBSY | MSTEN | CKPHA | CKPOL | SLVSEL | NSSIN | SRMT | RXBMT | 00000111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA1

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0.
NOTE: RXBMT = 1 when in Master Mode.

***Note:** In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 20.1 for timing parameters.

המשך בעמוד 37

**SPI0N: SPI0 Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SPIF | WCOL | MODF | RXOVRN | NSSMD1 | NSSMD0 | TXBMT | SPIEN | 00000110 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xF8

Bit 7: SPIF: SPI0 Interrupt Flag.
This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 6: WCOL: Write Collision Flag.
This bit is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPI0 modes. It must be cleared by software.

Bit 5: MODF: Mode Fault Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 4: RXOVRN: Receive Overrun Flag (Slave Mode only).
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. This bit is not automatically cleared by hardware. It must be cleared by software.

Bits 3–2: NSSMD1–NSSMD0: Slave Select Mode.
Selects between the following NSS operation modes:
(See **Section "20.2. SPI0 Master Mode Operation" on page 224** and **Section "20.3. SPI0 Slave Mode Operation" on page 226**).
00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin.
01: 4-Wire Slave or Multi-Master Mode (Default). NSS is always an input to the device.
1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.

Bit 1: TXBMT: Transmit Buffer Empty.
This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.

Bit 0: SPIEN: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

המשך בעמוד 38

**SPI0CKR: SPI0 Clock Rate**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SCR7 | SCR6 | SCR5 | SCR4 | SCR3 | SCR2 | SCR1 | SCR0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA2

Bits 7–0: SCR7–SCR0: SPI0 Clock Rate.
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

**SPI0DAT: SPI0 Data**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA3

Bits 7–0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.