



Lecture 9

Timer Operations and Programming

- ◆ מבוא.
- ◆ סיכום בנושא טיימרים.
- ◆ סדר פעולות בתכנון טיימרים.
- ◆ סיכום על אוגרים מיוחדים שאחראים על עבודת טיימרים.
 - ◆ Timer 0: 8-bit auto-reload mode (mode 2)
 - Programming sequence (timer 0 in mode 2)
 - Generating Interrupts on timer 0 overflow
 - ◆ Timer 3: 16-bit auto-reload mode (mode 0)
 - Timer 3 programming sequence
 - Generating interrupts on timer 3 overflow
 - ◆ Timer 2: 16-bit auto-reload mode (mode 1)
 - Timer 2: programming example
 - ◆ Timer 2: 16-bit counter/timer with capture (mode 0)

- ◆ שימוש בטיימרים: השהיות, ספירת אירועים, יצירת תדר לתקשורת טורית (baud rate)
- ◆ כאשר משתמשים בטיימרים ניתן לתכנן אותו בתור:
 - שימוש בדגל מילוי יתר (timer overflow flag);
 - לייצר פסיקה.
- משתמשים בדגל או פסיקה כדי לסנכרן קוד לביצוע פעולות כמו בדיקת מצב פורטים או לנשתמש ב- DAC או ADC.
- ניתן גם להשתמש לייצור גל ריבועי, משולש, שן ... בתדרים שונים.
- ◆ שימוש בתור מונה הוא בעיקר ספירה של כמות אירועים ולא השהייה בין אירועים.
- אירוע – זה פנייה חיצונית כלשהי שמשנה מצב הפורט מ"גבוה" ל"נמוך" בפורט הנבחר.
- ◆ טיימרים גם משחקים תפקיד של מחולל של תדר סנכרון לפורטים של תקשורת טורית (UART0 ו- UART1).
- "Baud rate" - זה תדר עבודה ביחידות סיביות בשנייה.
- ישנה טבלת תדרים קבוע. אנו נשתמש בתדרים 19200, 9600, 4800, 2400 סיביות בשנייה.

♦ למיקרו בקר C8051F020 ישנם 5 טיימרים.

Mode	Timer 0 & 1	Timer 2	Timer 3	Timer 4
0	13-bit counter/timer	16-bit counter/timer with capture	16-bit timer with auto-reload	16-bit counter/timer with capture
1	16-bit counter/timer	16-bit counter/timer with auto-reload		16-bit counter/timer with auto-reload
2	8-bit counter/timer with auto-reload (Baud rate Generator for UART0 and UART1, Timer 1 only)	Baud rate generator for UART0		Baud rate generator for UART1
3	Two 8-bit counter/timers (Timer 0 only)			



◆ לטיימרים בסיסיים (0 ו-1) (כמו 8051 קלאסי).

- צעד 1. בוחרים תדר השעון בעזרת סיביות CKCON.3 לטיימר 0 או CKCON.4 לטיימר 1. ניתן לבחור בין 2 אפשרויות – שעון של מערכת או שעון של מערכת מחולק ב-12.
- צעד 2. בוחרים צורת הטיימר – 2 סיביות לטיימר 0 ו-2 סיביות לטיימר 1 בתוך אוגר TMOD. (T0M[1:0] T1M[1:0])
- צעד 3. רושמים ערך התחלתי שממנו נתחיל לספור עד שטיימר מגיע למילוי יתר. עושים את זה בעזרת אוגרים מיועדים לכך. (TL0, TL1, TH0, TH1)
- צעד 4 (לא חייב). מאפשרים פסיקה של טיימרים ופסיקות באופן כללי. (ET0 או ET1 ו- EA באוגר IE).
- צעד 5. הפעלת סיביות TR0 או TR1 באוגר TCON.

◆ לטיימר 3.

- צעד 1. מכניסים ערך למילוי חוזר לתוך אוגרים (TMR3RLL , TMR3RLH).
- צעד 2. מכניסים ערך התחלתי לתוך אוגרים (TMR3RL , TMR3RH)
- צעד 3. בוחרים מקור של שעון באוגר (T3XCLK), תדר השעון (T3M) ומאתחלים סיביות בקרה (TR3).
- צעד 4 (לא חייבים): מאפשרים פסיקה של טיימר ומנגנון הפסיקות. (EA באוגר IE, ו- ET3 באוגר EIE2).
- צעד 5: מפעילים טיימר 3 ע"י הפעלת סיבית (TMR3CN).

◆ "מילוי חוזר" אומר שחומרה משחזרת ערך ששמור באחד האוגרים באופן אוטומטי ברגע שיש מילוי יתר מערך 0xFFFF לערך 0x0000.

◆ לטיימרים 2 ו- 4.

- צעד 1. בוחרים שעון של מערכת ע"י אוגר (CKCON).
- צעד 2. מכניסים ערך למילוי אוטומטי לאוגרים המיועדים לכך (RCAP2L, RCAP2H, RCAP4L, RCAP4H) במידה ומשתמשים במצב "מילוי חוזר" (auto-reload mode).
- צעד 3. כותבים ערך התחלתי לאוגרים מיועדים לכך (TL2, TL4, TH2, TH4).
- צעד 4. בוחרים צורת העבודה (C/Tx, CP/RLx) ומגדירים סיביות בקרה באוגר (TRx).
- צעד 5. (לא חובה) מאפשרים פסיקות של טיימר ומנגנון הפסיקות. (ET2 ו- EA באוגר IE, ET4 באוגר EIE2).
- צעד 6. מדליקים טיימר (T2CON, T4CON).

סיכום אוגרים מיוחדים בהקשר לטיימרים.

Timer SFR	Affected Timers	Purpose	Address	Bit Addressable
CKCON	0, 1, 2 and 4	Clock Control	8EH	No
TCON	0 and 1	Timer Control	88H	Yes
TMOD		Timer Mode	89H	No
TL0		Timer 0 Low Byte	8AH	No
TL1		Timer 1 Low Byte	8BH	No
TH0		Timer 0 High Byte	8CH	No
TH1		Timer 1 High Byte	8DH	No
T2CON		2	Timer 2 Control	C8H
RCAP2L	Timer 2 Low Byte Capture		CAH	No
RCAP2H	Timer 2 High Byte Capture		CBH	No
TL2	Timer 2 Low Byte		CCH	No
TH2	Timer 2 High Byte		CDH	No



סיכום אוגרים מיוחדים בהקשר לטיימרים.

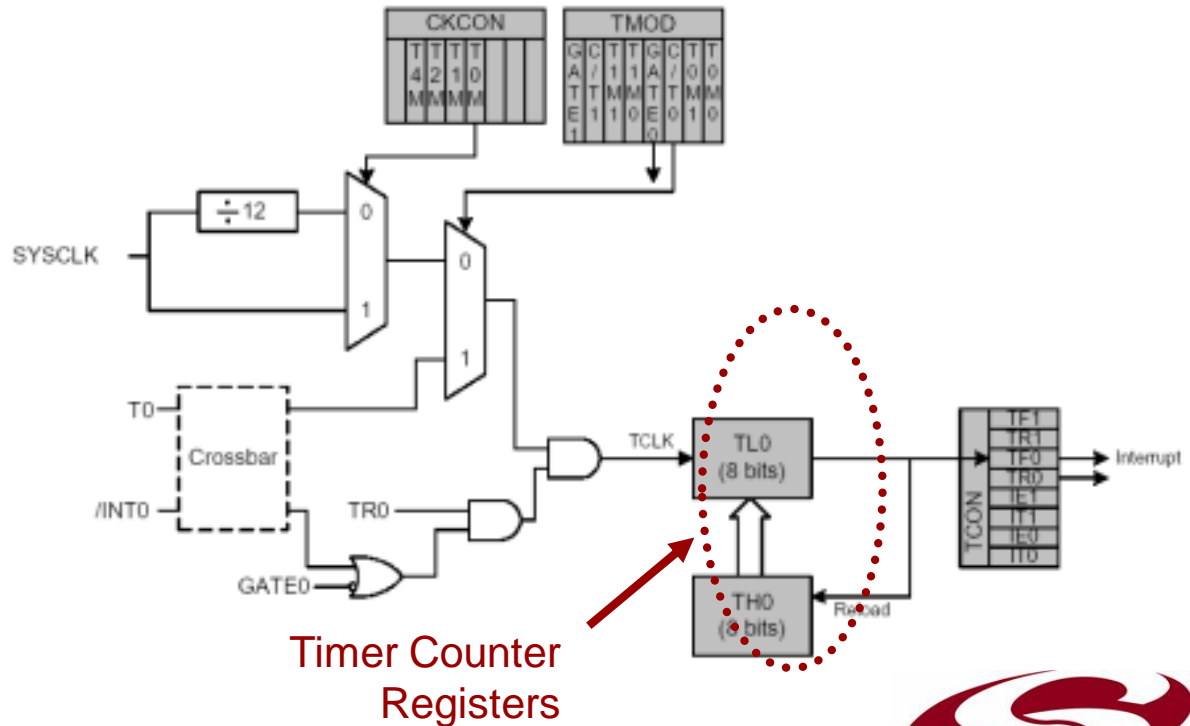
Timer SFR	Affected Timers	Purpose	Address	Bit Addressable
TMR3CN	3	Timer 3 Control	91H	No
TMR3RLL		Timer 3 Low Byte Reload	92H	No
TMR3RLH		Timer 3 High Byte Reload	93H	No
TMR3L		Timer 3 Low Byte	94H	No
TMR3H		Timer 3 High Byte	95H	No
T4CON	4	Timer 4 Control	C9H	No
RCAP4L		Timer 4 Low Byte Capture	E4H	No
RCAP4H		Timer 4 High Byte Capture	E5H	No
TL4		Timer 4 Low Byte	F4H	No
TH4		Timer 4 High Byte	F5H	No



טיימר 0: מצב טעינה חוזרת (טעינה אוטומטית), מצב 2.

- ◆ מצב מגדיר טיימר 0 וטיימר 1 במצב של טיימר או מונה 8 סיביות עם טעינה אוטומטית חוזרת עם ארך התחלתי מוגדר מראש.
- ◆ בית תחתון (TLx) עובד בתור טיימר 8 סיביות ובית עליון (THx) שומר ערך שלאחר וטיימר מתאפס מעלה אותו לתוך (TLx).
- ◆ כאשר ערך באוגר TLx משתנה מ-FFH ל-00H דגל של טיימר עולה ל-1 וערך של THx אוטומטי ניטען לתוך TLx.

- ◆ מצב זה מתאים לתהליכים כמו מחולל גלים או אירועים עם קצב קבוע ונקבע מראש, לאחר שהגדרנו TMOD ו-THx.
- ◆ חייבים להכניס ערך לתוך TLx לפני שאנו מאפשרים טיימר ומפעילים אותו בפעם הראשונה.
- ◆ ניתן להשתמש בטיימר 1 בתור מחולל 8 סיביות ל- UART0 ו-1 (baud rate generator) במצב 2.



Timer 0: Programming Step 1

Bit	Symbol	Description
7	-	Unused. Read=000b; Write=Don't care.
6	T_4M	Timer 4 Clock Select 0: Timer 4 uses the system clock divided by 12. 1: Timer 4 uses the system clock.
5	T_2M	Timer 2 Clock Select 0: Timer 2 uses the system clock divided by 12. 1: Timer 2 uses the system clock.
4	T_1M	Timer 1 Clock Select 0: Timer 1 uses the system clock divided by 12. 1: Timer 1 uses the system clock.
3	T_0M	Timer 0 Clock Select 0: Timer 0 uses the system clock divided by 12. 1: Timer 0 uses the system clock
2-0	Reserved	Read=000b. Must Write=000b

בוחרים צורת השעון בעזרת אוגר **CKCON** (Clock Control) Register

לטיימר 0 אנו משנים ביט מספר 3 (T_0M).



Timer 0—Programming Step 1

```
CKCON |= 0x04; //-- TOM = 1; Timer 0 uses SysClock  
CKCON &= 0xF0; //-- TOM = 0; Timer 0 uses SysClock/12
```

◆ מאוד חשוב שברגע שאתם מגדירים מצב של טיימר מסויים לא לפגוע בטיימרים אחרים.

◆ מה קורה כאשר כותבים משפט הבא? $CKCON = 0x04$ אנו משנים הגדרות של טיימרים אחרים.



Timer 0—Programming Step 2

◆ בוחרים צורת העבודה ע"י שינוי אוגר **TMOD** (Timer Mode). לטיימר 0 יש לגעת רק בסיביות 0 עד 3 ואין לשנות סיביות מ-4 עד 7. אוגר זה לא ניתן לגישה בתור סיביות בודדות.

```
//-- Timer 0 in Mode 2  
//-- Set T0M[1:0] to 10b  
TMOD &= 0xFC;  
TMOD |= 0x02;
```

Bit	Symbol	Description
7	GATE1	Timer 1 Gate Control 0: Timer 1 enabled when TR1(TCON.6)=1 irrespective of /INT logic level 1: Timer 1 enabled only when TR1=1 AND /INT=logic 1
6	C/T1	Counter/Timer 1 Select 0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4). 1: Counter Function: Timer 1 incremented by high- to-low transition on external input pin (T1).
5-4	T1M1- T1M0	Timer 1 Mode Select
3	GATE0	Timer 0 Gate Control 0: Timer 0 enabled when TR0(TCON.4)=1 irrespective of /INT logic level 1: Timer 0 enabled only when TR0=1 AND /INT=logic 1
2	C/T0	Counter/Timer 0 Select 0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3). 1: Counter Function; Timer 0 incremented by high- to-low transition on external input pin (T0).
1-0	T0M1-T0M0	Timer 0 Mode Select

Timer 0/Timer 1 Mode Select Bits

T_xM1	T_xM0	Mode	Description
0	0	0	13 bit Counter/Timer
0	1	1	16 bit Counter/Timer
1	0	2	8 bit Counter/Timer with Auto-reload
1	1	3	Timer 1: Inactive Timer 0: Two 8 bit Counter/Timers

x = 0 or 1 for Timer0 or Timer1

Timer 0—Programming Step 3

◆ כותבים ערך התחלתי לתוך האוגר **.TL0**.

```
TL0 = 0xFF;    //-- start value
```

◆ כותבים ערך למילוי חוזר לתוך אוגר **.TH0**.

```
TH0 = 0x80;    //-- reload value
```



Timer 0—Programming Step 5

- ◆ מפעילים טיימר 0 ע"י הפעלת ביט מס' 4 בתוך אוגר TCON (Timer Control)
- ◆ ניתן לגשת בנפרד לכל ביט של אוגר זה.

```
//--start Timer 0  
//-- (TCON.4 = 1)  
  
TR0 = 1;
```

Bit	Symbol	Description
7	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 (ISR). 0: No Timer 1 overflow detected 1: Timer 1 has overflowed
6	TR1	Timer 1 Run Control 0: Timer 1 disabled 1: Timer 1 enabled
5	TF0	Timer 0 Overflow Flag Same as TF1 but applies to Timer 0 instead. 0: No Timer 0 overflow detected 1: Timer 0 has overflowed
4	TR0	Timer 0 Run Control 0: Timer 0 disabled 1: Timer 0 enabled
3	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 ISR if IT1=1. This flag is the inverse of the /INT1 input signal's logic level when IT1=0
2	IT1	Interrupt 1 Type Select 0: /INT1 is level triggered 1: /INT1 is edge triggered
1	IE0	External Interrupt 0 Same as IE1 but applies to IT0 instead.
0	IT0	Interrupt 0 Type Select 0: /INT0 is level triggered 1: /INT0 is edge triggered

שימוש בפסיקות של טיימר 0 (מילוי יתר).

- ◆ כאשר טיימר TLx מתמלא מ- FFH ל- 00H, דגל מילוי יתר באוגר TCON עולה ל-1.
- ◆ לטיימר 0 יש לבדוק סיבית מספר 5 באוגר **TCON**. (TCON.5)
- ◆ כדי לבדוק מילוי יתר של טיימר יש 2 אפשרויות:
 - בעזרת שיטת POLLING (משאל) של ביט מילוי יתר.
 - בעזרת אפשר פסיקה של מילוי יתר של טיימר.
- מאפשרים פסיקות בעזרת אוגר פסיקות **IE** (interrupt enable).



אוגר (IE) Interrupt Enable

◆ ברגע שאפשרנו הפעלת פסיקה, שגרת טיפול בפסיקות ISR מופעלת בצורה אוטומטית כל פעם שטיימר מתמלא.

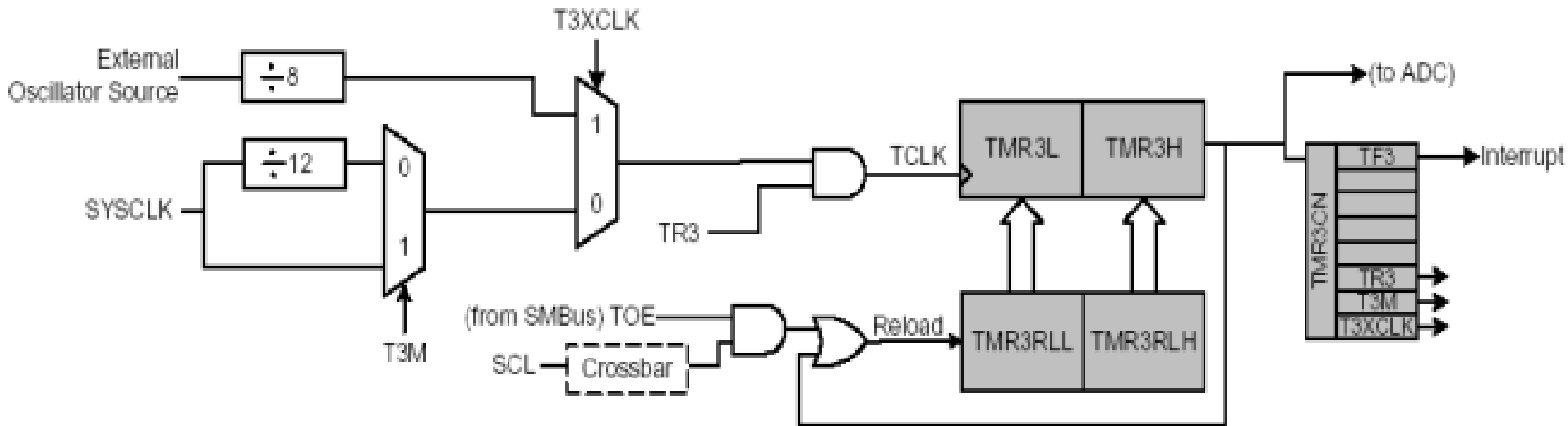
```
ET0 = 1; // Enable interrupt
        // request generated
        // by Timer 0 overflow
        // flag, TF0 (TCON.5)
```

Bit	Symbol	Description
7	EA	Enable All Interrupts 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	IEGF0	General Purpose Flag 0 This is a general purpose flag for use under software control.
5	ET2	Enable Timer 2 Interrupt 0: Disable Timer 2 Interrupt. 1: Enable interrupt requests generated by TF2 (T2CON.7).
4	ES0	Enable UART0 Interrupt 0: Disable UART0 Interrupt. 1: Enable UART0 Interrupt.
3	ET1	Enable Timer 1 Interrupt 0: Disable Timer 1 Interrupt. 1: Enable interrupt requests generated by TF1 (TCON.7).
2	EX1	Enable External Interrupt 1 0: Disable external interrupt 1. 1: Enable interrupt request generated by the /INT1 pin.
1	ET0	Enable Timer 0 Interrupt 0: Disable Timer 0 Interrupt. 1: Enable interrupt requests generated by TF0 (TCON.5).
0	EX0	Enable External Interrupt 0 0: Disable external interrupt 0. 1: Enable interrupt request generated by the /INT0 pin.



טיימר 3 במצב מילוי אוטומטי 16 סיביות. (מצב 0).

- ◆ טיימר 3 תמיד מוגדר בתור טיימר עם מילוי אוטומטי, כאשר ערך רצוי נמצא באוגרים TMR3RLL ו-TMR3RLH.
- ◆ ניתן להגדיר טיימר 3 אך ורק בעזרת אוגר TMR3CN.



טיימר 3 במצב מילוי אוטומטי 16 סיביות. (מצב 0).

- ◆ ניתן להגדיר תדר העבודה של טיימר ניתן לחבר למקור תדר חיצוני מחולק ב-8, או שעון פנימי (שיכול להיות ללא שינוי או לאחר חלוקה ב-12 ביט **T3M** אחראי על בחירה הזאת).
- ◆ כאשר **T3XCLK** שווה ל-1 טיימר 3 מופעל ע"י מקור תדר חיצוני (מחולק ב-8) ללא השפעה של בחירת מקור תדר פנימי.
- ◆ כאשר **T3XCLK** שווה ל-0, טיימר 3 מופעל ע"י מחולל פנימי (תדר המערכת) וביט **T3M** מגדיר האם תהיה חלוקה ב-12 או לא.
- ◆ ניתן להשתמש בטיימר 3 כדי להפעיל ADC (ממיר אנלוגי לדיגיטאלי).



Timer 3 Programming Sequence—Step 1a

צריכים לבחור צורת השעון
בעזרת ביט T3XCLK
(TMR3CN.0)

```
//-- Stop Timer 3,  
/-- Clear TF3  
/-- use SYSCLK/12  
/-- timebase  
TMR3CN = 0x00;
```

```
//-- use External  
/-- Oscillator  
TMR3CN |= 0x01;
```

Bit	Symbol	Description
7	TF3	Timer 3 Overflow Flag Set by hardware when Timer 3 overflows from FFFFH to 0000H. When the Timer 3 interrupt is enabled, setting this bit causes the CPU vectors to the Timer 3 ISR. This bit is not automatically cleared by hardware and must be cleared by software.
6-3	UNUSED	Read=0000b, Write=don't care
2	TR3	Timer 3 Run Control 0: Timer 3 disabled 1: Timer 3 enabled
1	T3M	Timer 3 Clock Select 0: Counter/Timer 3 uses the system clock divided by 12. 1: Counter/Timer 3 uses the system clock.
0	T3XCLK	Timer 3 External Clock Select 0: Timer 3 clock source defined by bit T3M (TMR3CN.1) 1: Timer 3 clock source is the external oscillator input divided by 8. T3M is ignored.



Timer 3 Programming Sequence—Step 1b-2

- ◆ נכתוב ערך מתמלא לתוך אוגרים (TMR3RLL+TMR3RLH) או אוגר .TMR3RL
- ◆ אם משתמשים בשעון פנימי (SYSCLK), בוחרים חלוקת השעון ב-12 או שעון רגיל של המערכת. לצורך זה משתמשים בסיבית בשם **T3M** (TMR3CN.1).

```
TMR3CN |= 0x02;           //-- use SYSCLK (NOT SYSCLK/12) timebase  
  
TMR3RL = 0xffff6;
```

Timer 3 Programming Sequence—Step 3 & 4

- ◆ מעתיקים ערך התחלתי לתוך אוגרים (TMR3L+TMR3H) או אוגר TMR3. ערך זה לא חייב להיות זהה לערך ששמנו בתוך אוגר TMR3RL בסעיף הקודם.

```
TMR3 = 0xffff; //-- set to reload immediately
```

- ◆ מפעילים טיימר ע"י עליית ביט TR3. (TMR3CN.2) שימו לב שאוגר TMR3CN לא נתון לגישה בצורת סיביות וכדי לבצע שינוי יש להשתמש בפעולות OR ו-AND, NOT ו-XOR.

```
TMR3CN |= 0x04; //-- Start Timer 3
```



Extended Interrupt Enable 2 (EIE2) SFR

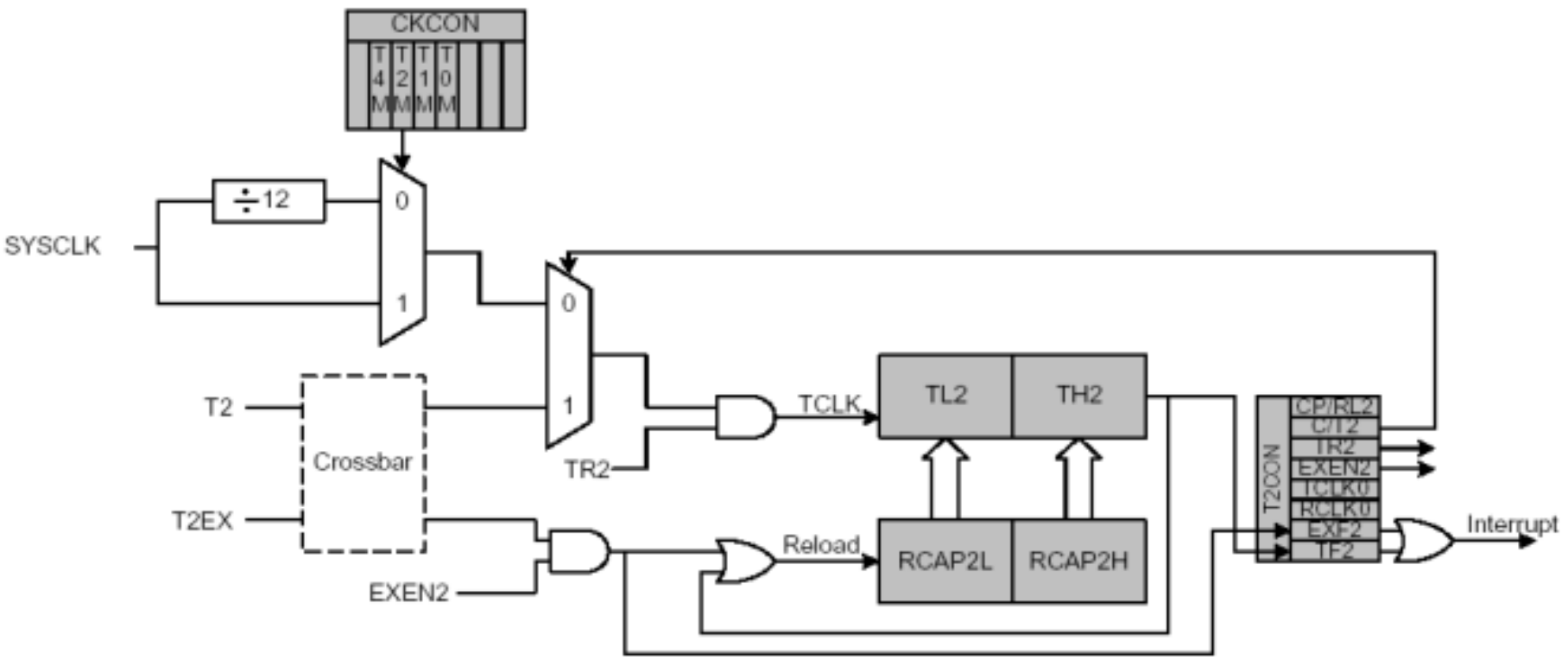
- ◆ מעלים ביט **EIE2.0** (ET3) ל-"1" כדי לאפשר בקשה לפסיקה של טיימר 3 (Timer 3 (TF3) overflow).
- ◆ כל פעם שטיימר מתמלא, שגרת טיפול בפסיקה ISR, מתבצעת באופן אוטומטי.

Bit	Symbol	Description
7	EXVLD	Enable External Clock Source Valid (XTLVLD) Interrupt 0: Disable XTLVLD interrupt. 1: Enable interrupt requests generated by XTLVLD (OXCXCN.7)
6	ES1	Enable UART1 Interrupt 0: Disable UART1 Interrupt. 1: Enable UART1 Interrupt.
5	EX7	Enable External Interrupt 7 0: Disable external interrupt 7. 1: Enable interrupt request generated by the External Interrupt 7 input pin.
5	EX6	Enable External Interrupt 6 0: Disable external interrupt 6. 1: Enable interrupt request generated by the External Interrupt 6 input pin.
3	EADC1	Enable ADC1 End of Conversion Interrupt 0: Disable ADC1 End of Conversion interrupt. 1: Enable interrupt requests generated by the ADC1 End of Conversion Interrupt.
2	ET4	Enable Timer 4 Interrupt 0: Disable Timer 4 Interrupt. 1: Enable interrupt requests generated by TF4 (T4CON.7).
1	EADC0	Enable ADC0 End of Conversion Interrupt 0: Disable ADC0 End of Conversion interrupt. 1: Enable interrupt requests generated by the ADC0 End of Conversion Interrupt.
0	ET3	Enable Timer 3 Interrupt 0: Disable Timer 3 Interrupt. 1: Enable interrupt requests generated by TF3 (TMR3CN.7).

```
EIE2 |= 0x01; // Enable  
           // Timer 3  
           // Interrupt  
           // requests
```


טיימר 2 מצב 1 – טיימר 16 סיביות במצב מילוי אוטומטי.

- ♦ ערך שאנו רוצים לשחזר מכניסים לתוך אוגרים (RCAP2L+RCAP2H) או לתוך אוגר RCAP2.



טיימר 2 מצב 1 – טיימר 16 סיביות במצב מילוי אוטומטי.

◆ צעדי תכנון.

- צעד 1: בוחרים תדר המקור בעזרת (CKCON).
- צעד 2: מכניסים ערך למילוי חוזר לתוך האוגרים (RCAP2L, RCAP2H).
- צעד 3: כותבים ערך התחלתי לתוך האוגר (TL2, TH2).
- צעד 4: בוחרים מצב המערכת ע"י רגליים (C/T2, CP/RL2), אחרי זה מגדירים סיבית בקרה (TR2) כדי להפעיל טיימר (T2CON).



T2CON—Timer 2 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: (bit addressable) 0xC8

Bit7: TF2: Timer 2 Overflow Flag.
Set by hardware when Timer 2 overflows. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software. TF2 will not be set when RCLK0 and/or TCLK0 are logic 1.

Bit6: EXF2: Timer 2 External Flag.
Set by hardware when either a capture or reload is caused by a high-to-low transition on the T2EX input pin and EXEN2 is logic 1. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 Interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit5: RCLK0: Receive Clock Flag for UART0.
Selects which timer is used for the UART0 receive clock in modes 1 or 3.
0: Timer 1 overflows used for receive clock.
1: Timer 2 overflows used for receive clock.

Bit4: TCLK0: Transmit Clock Flag for UART0.
Selects which timer is used for the UART0 transmit clock in modes 1 or 3.
0: Timer 1 overflows used for transmit clock.
1: Timer 2 overflows used for transmit clock.

Bit3: EXEN2: Timer 2 External Enable.
Enables high-to-low transitions on T2EX to trigger captures or reloads when Timer 2 is not operating in Baud Rate Generator mode.
0: High-to-low transitions on T2EX ignored.
1: High-to-low transitions on T2EX cause a capture or reload.

Bit2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit1: C/T2: Counter/Timer Select.
0: Timer Function: Timer 2 incremented by clock defined by T2M (CKCON.5).
1: Counter Function: Timer 2 incremented by high-to-low transitions on external input pin (T2).

Bit0: CP/RL2: Capture/Reload Select.
This bit selects whether Timer 2 functions in capture or auto-reload mode. EXEN2 must be logic 1 for high-to-low transitions on T2EX to be recognized and used to trigger captures or reloads. If RCLK0 or TCLK0 is set, this bit is ignored and Timer 2 will function in auto-reload mode.
0: Auto-reload on Timer 2 overflow or high-to-low transition at T2EX (EXEN2 = 1).
1: Capture on high-to-low transition at T2EX (EXEN2 = 1).

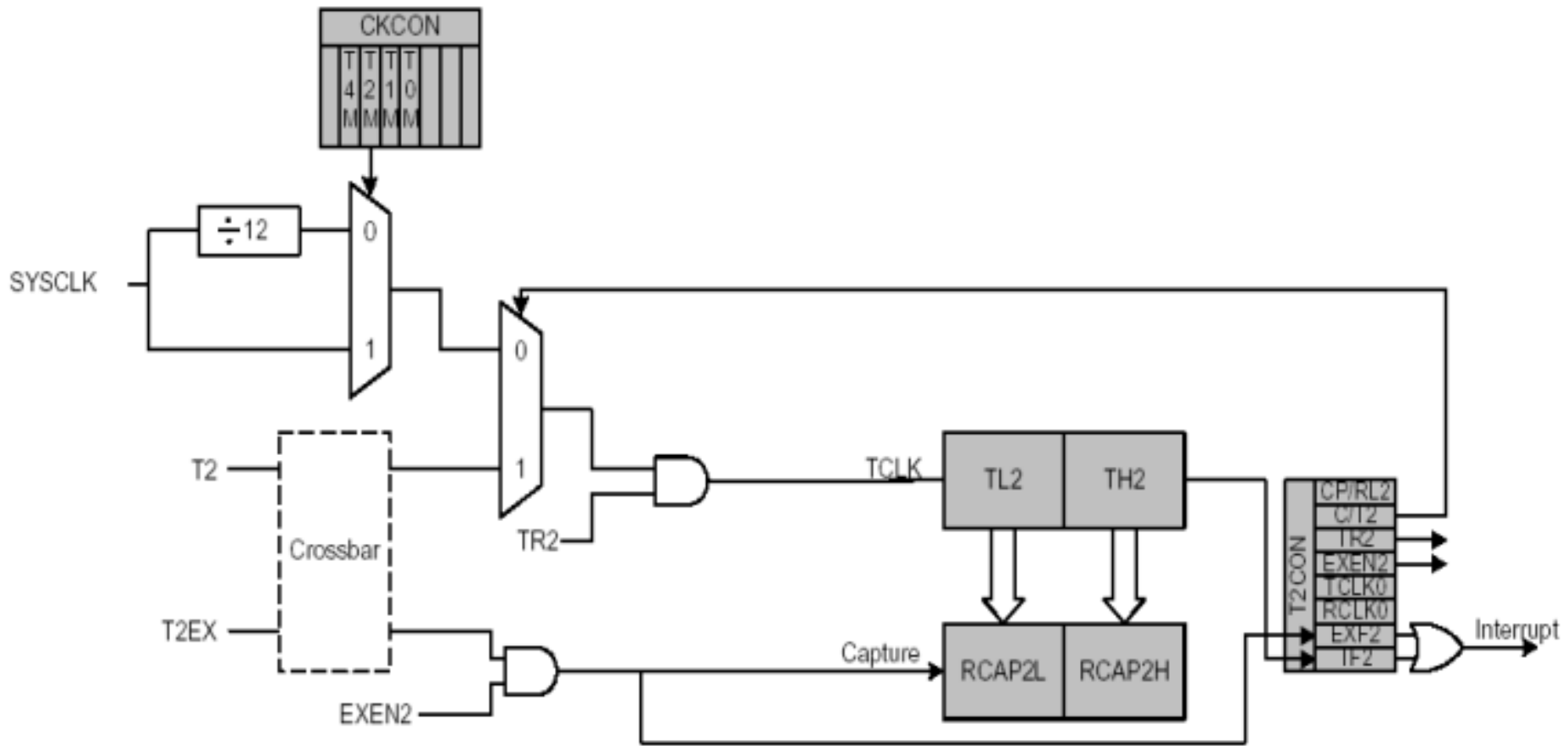


Mode Configuration for Timer 2

RCLK0	TCLK0	CP/RL2	TR2	Mode
0	0	1	1	16 Bit Counter/Timer with Capture
0	0	0	1	16 Bit Counter/Timer with Auto-reload
0	1	X	1	Baud Rate Generator for UART0
1	0	X	1	Baud Rate Generator for UART0
1	1	X	1	Baud Rate Generator for UART0
X	X	X	0	Off

```
//-- Configure Timer2 to auto-reload and generate an interrupt at interval
//-- specified by <counts> using SYSClk/12 as its time base.
void Init_Timer2 (unsigned int counts)
{
    CKCON = 0x00;    // Define clock (T2M). Timer 2
                    // uses system clock DIV BY 12
    // CKCON |= 0x20; if you want to use system clock
    T2CON = 0x00;
    // T2CON.1 = 0 --> T2 set for Timer function
    // (C/T2) i.e.incremented by clock defined by T2M
    // T2CON.0 = 0 --> Allow Auto-reload on Timer2 overflow (CP/RL2)
    // T2CON.3 = 0 --> High-to-Low transitions on
    // T2EX ignored (EXEN2)
    // T2CON.2 = 0 --> Disable Timer2
    RCAP2 = -counts; // Init reload values in the Capture registers
    T2 = 0xFFFF;    // count register set to reload
                    // immediately when the first clock occurs
    IE |= 0x20;     // IE.5, Enable Timer 2 interrupts (ET2)
    T2CON |= 0x04;  // Start Timer2 by setting TR2 (T2CON.2) to 1
}
}
```

- ◆ במצב זה טיימר או מונה עובד בתור טיימר רגיל 16 סיביות וסופר עד FFFFH ומתמלא כאשר מתאפס. ברגע שהוא מתמלא הוא מעלה סיבית TF2.
- ◆ ניתן להשתמש במנגנון פסיקות במידה ופסיקה מאופשרת.
- ◆ הבדל עיקרי בין מצב זה למצבים אחרים בכך שניתן לספור אירועים בירידת פולס ברגל T2EX.



- ◆ כדי להפעיל מונה סיבית **EXEN2 (T2CON.3)** צריך להיות "1".
- ◆ ירידת שעון ברגל **T2EX** מפעילה צעדים הבאים:
 - ערך בגודל של 16 סיביות לטיימר 2 נמצאת באוגרים (**TH2, TL2**) נשמר לתוך האוגרים (**RCAP2H, RCAP2L**).
 - דגל חיצוני של טיימר 2 (**EXF2**) עולה ל- "1".
 - ניתן לייצר פסיקה של טיימר 2 במידה ומנגנון פסיקות מאופשר.



S I L I C O N L A B S

www.silabs.com/MCU