

Timers And Counters

בס"ד



מגישים:

בר אליס - 208545236

חיים עוזר – 316063569

קבוצה 3

מעבדה למיקרו בקרים תשפ"ב

Hi!

מהו טיימר?

- הטיימר הוא מונה בינארי אשר מונע על ידי שעון בתדירות ידועה.
- לטיימר משויכים רגיסטרים (אוגרים) שחלקם מאפשרים פסיקה (Interrupt Enable)
- את הזמן שעבר ניתן לחשב כך: $\Delta t = T \cdot N = \frac{1}{f} \cdot N$. כאשר N מתאר את מספר הספירות שהטיימר ספר.



שימושי הטיימר במיקרו בקר MSP430

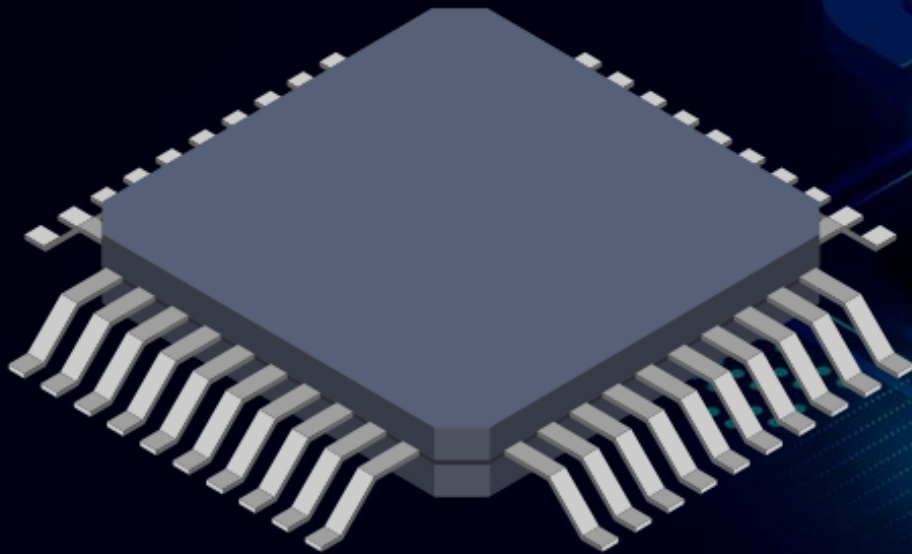
- לסנכרן פעולות בטווחי זמן מחזוריים.
- לאפשר עירור מחזורי ממצב שינה.
- ספירת אות המעבר.
- 4 דרגות מצב של מעבר למצב שינה (LPM) לצריכת זרם מזערי.



סוגי הטיימרים במיקרו בקר MSP430

במשפחת מיקרו בקר MSP430 קיימים ארבעה טיימרים.
כל אחד מהם בנוי לתת מענה לסוג משימות שונה.

- Watchdog Timer
- Basic Timer 1
- Timer_A
- Timer_B

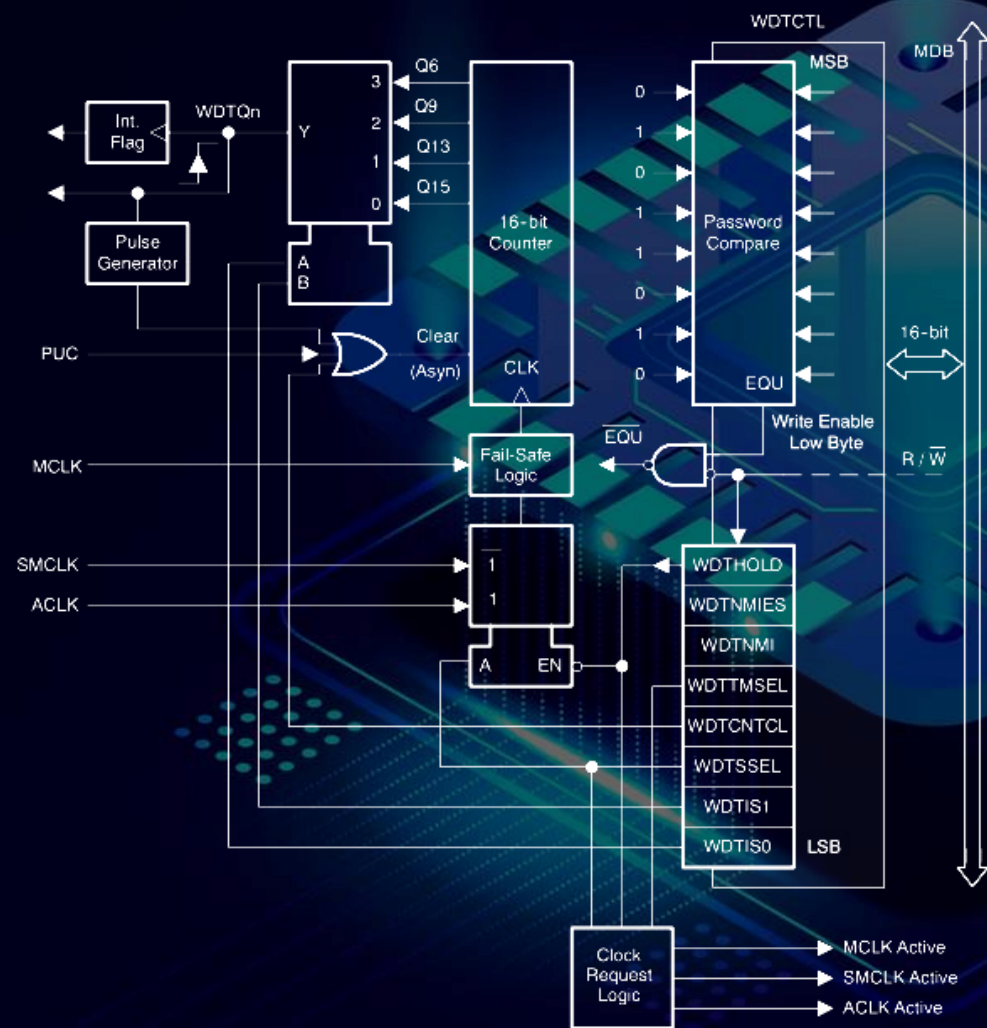


WatchDog Timer

Watchdog זהו טיימר שנועד למנוע תקלות חומרה באשר המערכת נכנסת ללופ אינסופי או באג מסויים. במקרים כאלו לאחר סיום ספירת הטיימר ברגיסטרים הוא מאתחל את מערכת. נהוג לנטרל אותו בתחילת הקוד.



WatchDog Timer - וק'צה



Basic Timer

הטיימר בנוי משני מונים של 8 סיביות שיכולים לתפקד ביחד למונה של 16 סיביות:

- שני האוגרים הם -BTCNT1, BTCNT2

- כאשר רוצים לעבוד במצב של טיימר של 16 סיביות יש להשתמש באוגר BTCTL

תפקידי Basic Timer:

- יצירת מרווח תדרים נמוך לתפעול ה-LCD.
- פונקציית RTC (Real Time Clock) - נותנת אפשרות לשעון דיגיטלי לשמור על מעקב מדויק של זמן גם כאשר אין הספקת חשמל ו/או צריכת חשמל מינימלית.
- אפשרות פסיקה (interrupt) עבור ה-LCD.

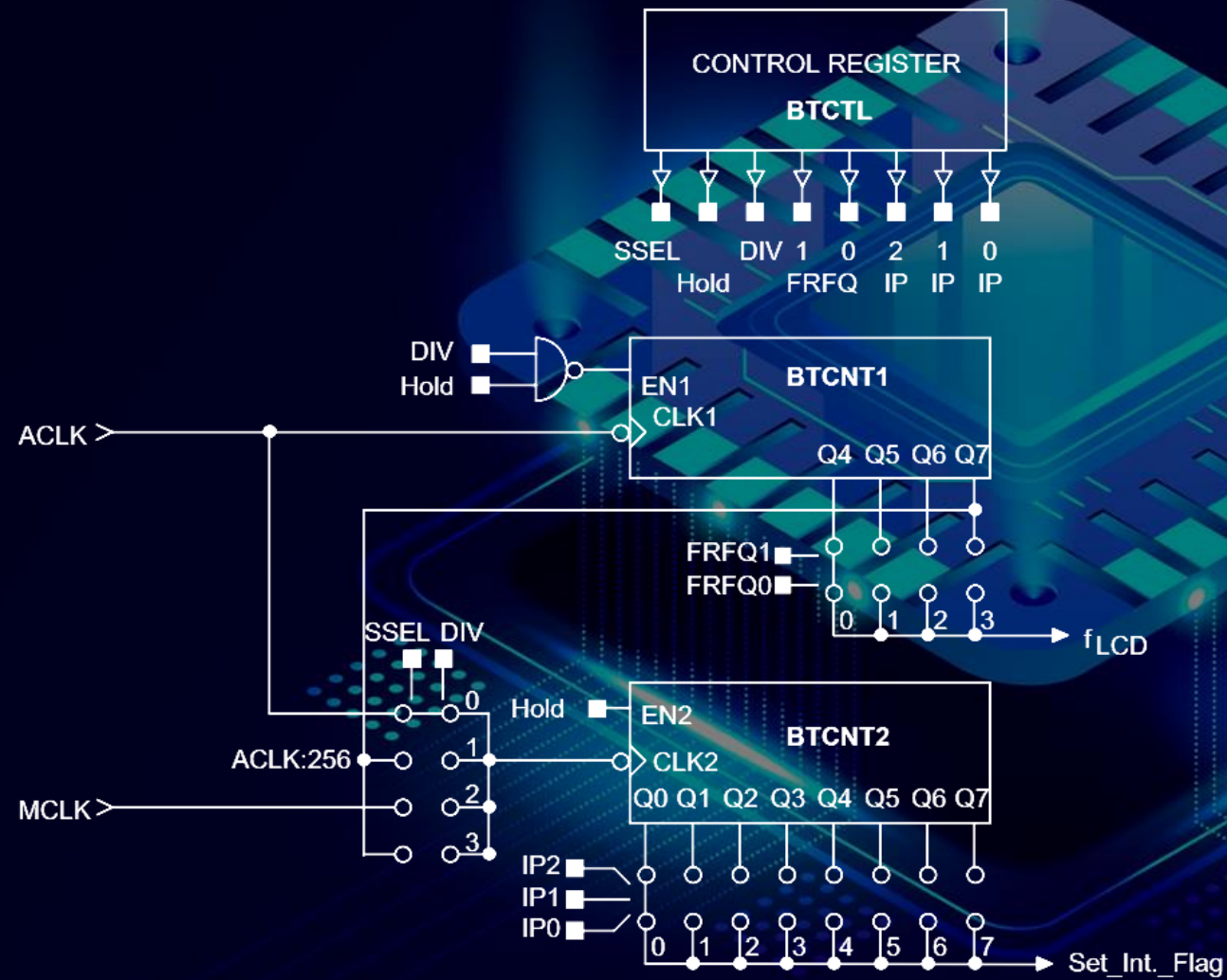
סיביות Basic Timer

- את שעון הייחוס מגדירים בעזרת BTSSEL
- ניתן לעצור את הטיימר בעזרת סיבית BTHOLD
- כאשר רוצים להשתמש בטיימר 16 סיביות משתמשים בסיבית BTDIV
- תדר f_{LCD} מוגדר בעזרת סיביות BTFRFQx כאשר ניתן לבחור את התדרים הבאים: $ACLK/256$, $ACLK/128$, $ACLK/64$, or $ACLK/32$
- סיביות פסיקה מוגדרות בעזרת מילת בקרה BTIPx

BTCTL, Basic Timer 1 Control Register

7	6	5	4	3	2	1	0
BTSSEL	BTHOLD	BTDIV	BTFRFQ1	BTFRFQ0	BTIP2	BTIP1	BTIP0

סקיזה - Basic Timer

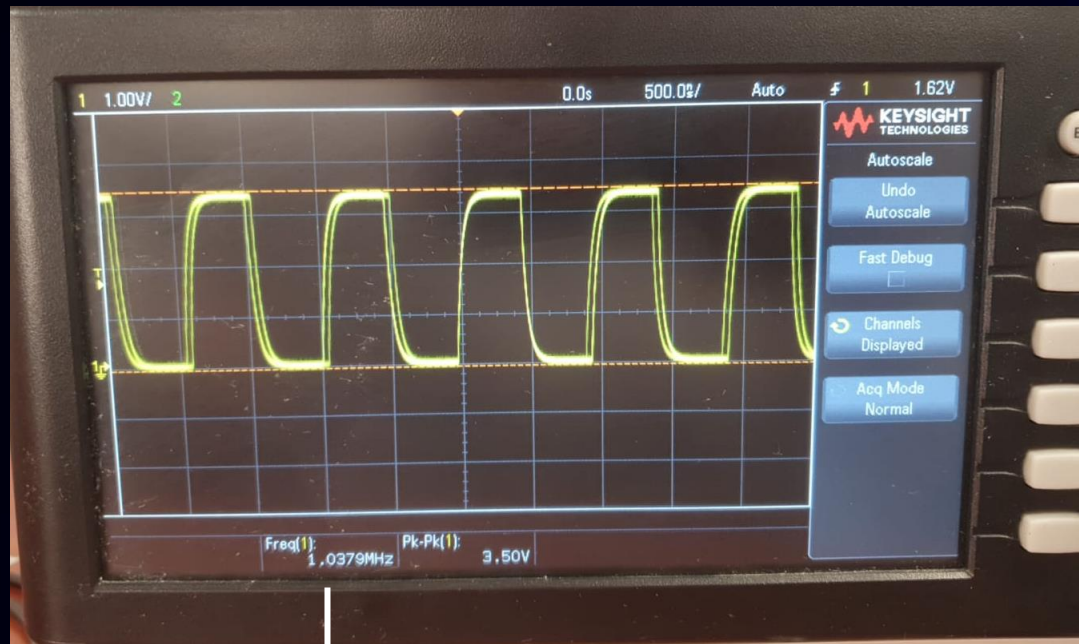


Timer_A

- הטיימר מכיל 16bit עם רגיסטרים עבור השוואה או לכידה.
- ניתן לעבוד במצב של פסיקה כמו בטיימר הבסיסי
- ניתן לבצע פסיקה גם במצב של OVERFLOW - מסיום המונה עד לספירה מחודשת.
- ניתן לבחור את סוג השעון – תדרי ACLK או MCLK (מדדנו במעבדה)
- Pulse Width Modulation (PWM) - טכניקה לשליטה על גורם זמן מחזור של האות.
- הטיימר מכיל 4 מצבי הפעלה – Up, Continuous, Up-Stop Mode, Down Mode

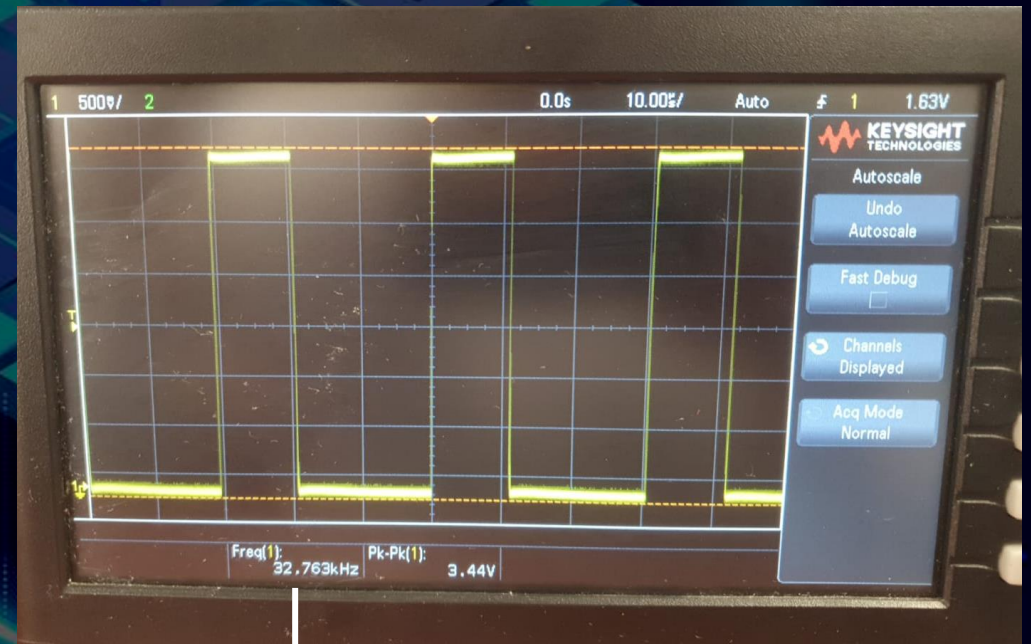
Timer_A

תדרי MCLK



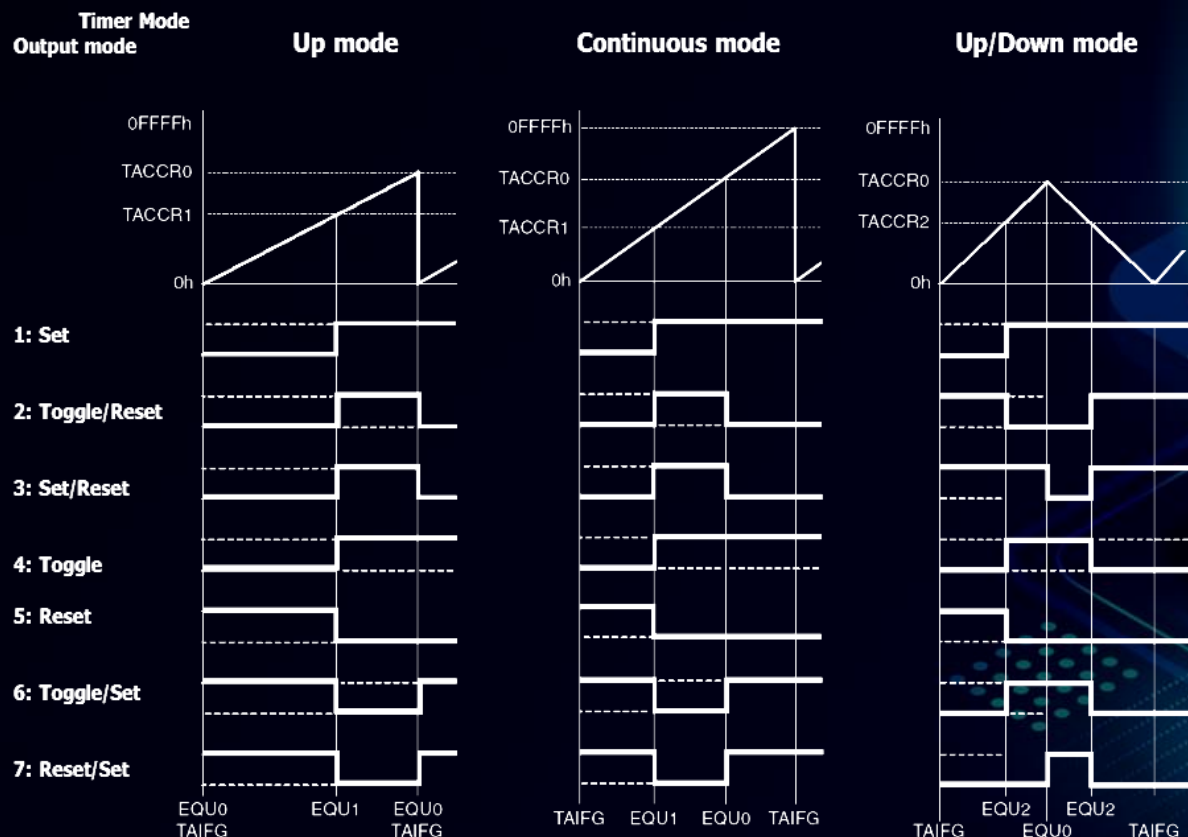
$$f_{MCLK} = 1\text{ MHz}$$

תדרי ACLK



$$f_{ACLK} = 32.768\text{ kHz}$$

Timer_A/B – מצבי פעולה



- Stop Mode - משמש להשהיית הטיימר

- Up Mode - משמש לספירה עד ערך מסוים השמור ברגיסטר והמקסימלי הוא עד-FFFFh (2^{16})

- Continuous Mode - הטיימר סופר באופן רציף עד לגודל הרגיסטר (2^{16}) כל הזמן.

- Up-Down Mode – משמש לספירה עד ערך מסוים השמור ברגיסטר וחזרה עד ל-0.

Timer_A/B – מצב לכידה והשוואה

מצב לכידה (Capture Mode) :

- מאופשר בפקודה CAP=1.
- ניתן להגדיר במצב של עלייה או ירידה או שניהם בהתאם למצבי הפעולה דרך סיביות MCx
- מצב זה מיועד לבצע מדידות בזמן מחזור של פעולות תוך כדי ניצול מינימלי של המעבד.

מצב השוואה (Compare Mode) :

- מאופשר בפקודה CAP=0.
- לעבודה ב PWM או לעבודה במרווחי זמן מסוימים.
- במצב Compare ניתן לייצר פסיקות (interrupts).

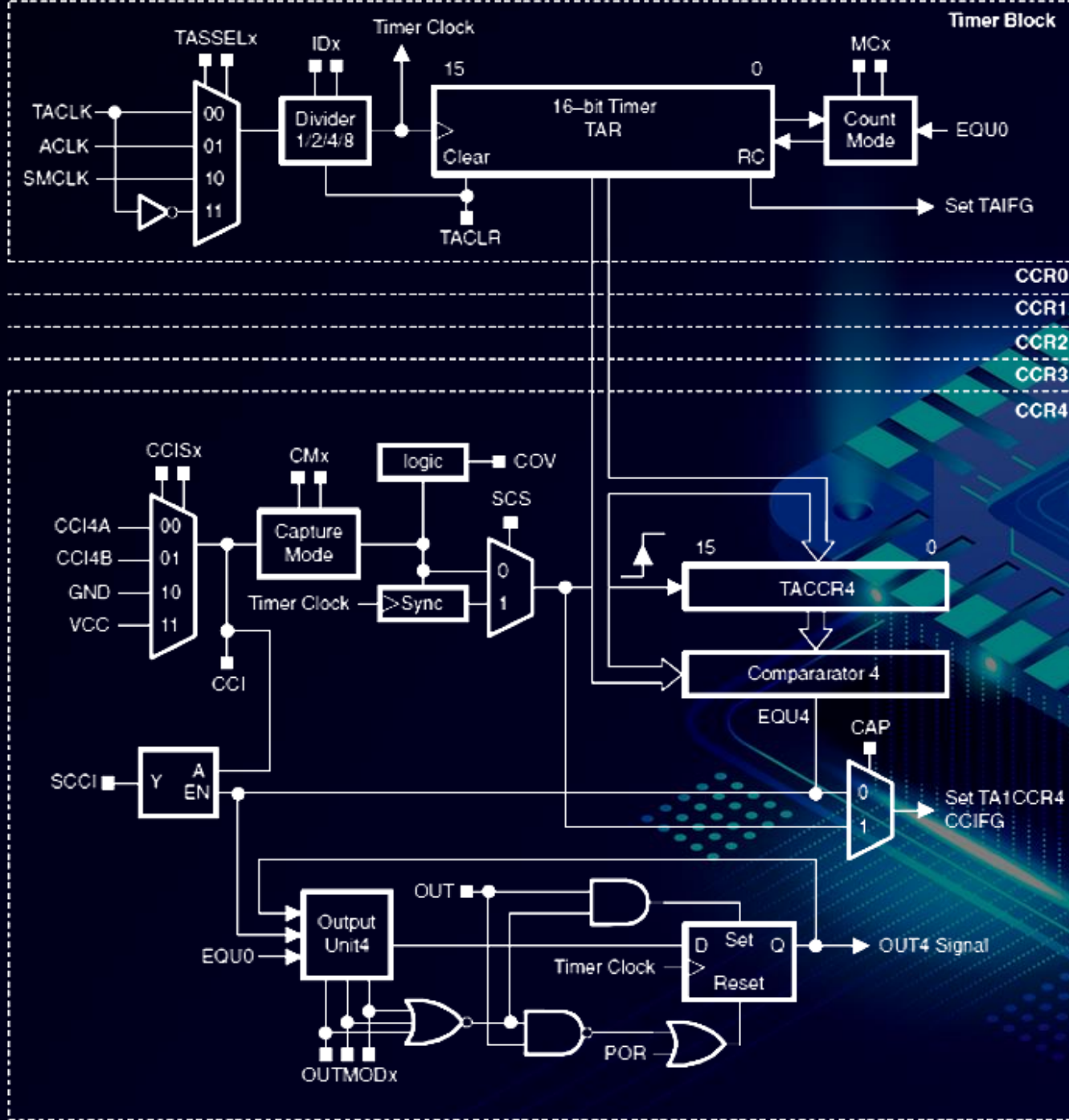
רגיסטר בקרה של Timer_A/B

- סיביות 10-15 ו-3 לא בשימוש
- TASSELx - בחירת השעון (Select)
- Idx – חלוקת אות השעון ל-2, 4 או 8.
- MCx – קובע את 4 מצבי הפעולה
- TACLx – מאפס את השעון ואת המנייה שלו (ספירה קדימה או אחורה)
- TAlx – סיביות פסיקה

TACTL, Timer_A Control Register

15				10		9		8	
Unused						TASSEL1		TASSEL0	
7		6		5		4		3	
ID1		ID0		MC1		MC0		Unused	
2		1		0		TACLR		TAIE	
TAIFG									

סקיצה - Timer A



Timer_B

Timer_B זהה כמעט לחלוטין ל-Timer_A מלבד הנושאים הבאים:

- ניתן לתכנת את גודל הסיביות ל- 8, 10, 12 או 16 בשונה מ-Timer_A שפועל ב-16 סיביות בלבד.
- ניתן להגדיר את היציאות של Timer_B כ high impedance (פקודה TBOUTH=1)
- לא קיים SSCI בטיימר B - סינכון כניסות לכידה/השוואה.

TBCTL, Timer_B Control Register

15	14	13	12	11	10	9	8
Unused	TBCLGRP1	TBCLGRP0	CNTL1	CNTL0	Unused	TBSSEL1	TBSSEL0
7	6	5	4	3	2	1	0
ID1	ID0	MC1	MC0	Unused	TBCLR	TBIE	TBIFG

סקיצה - Timer B

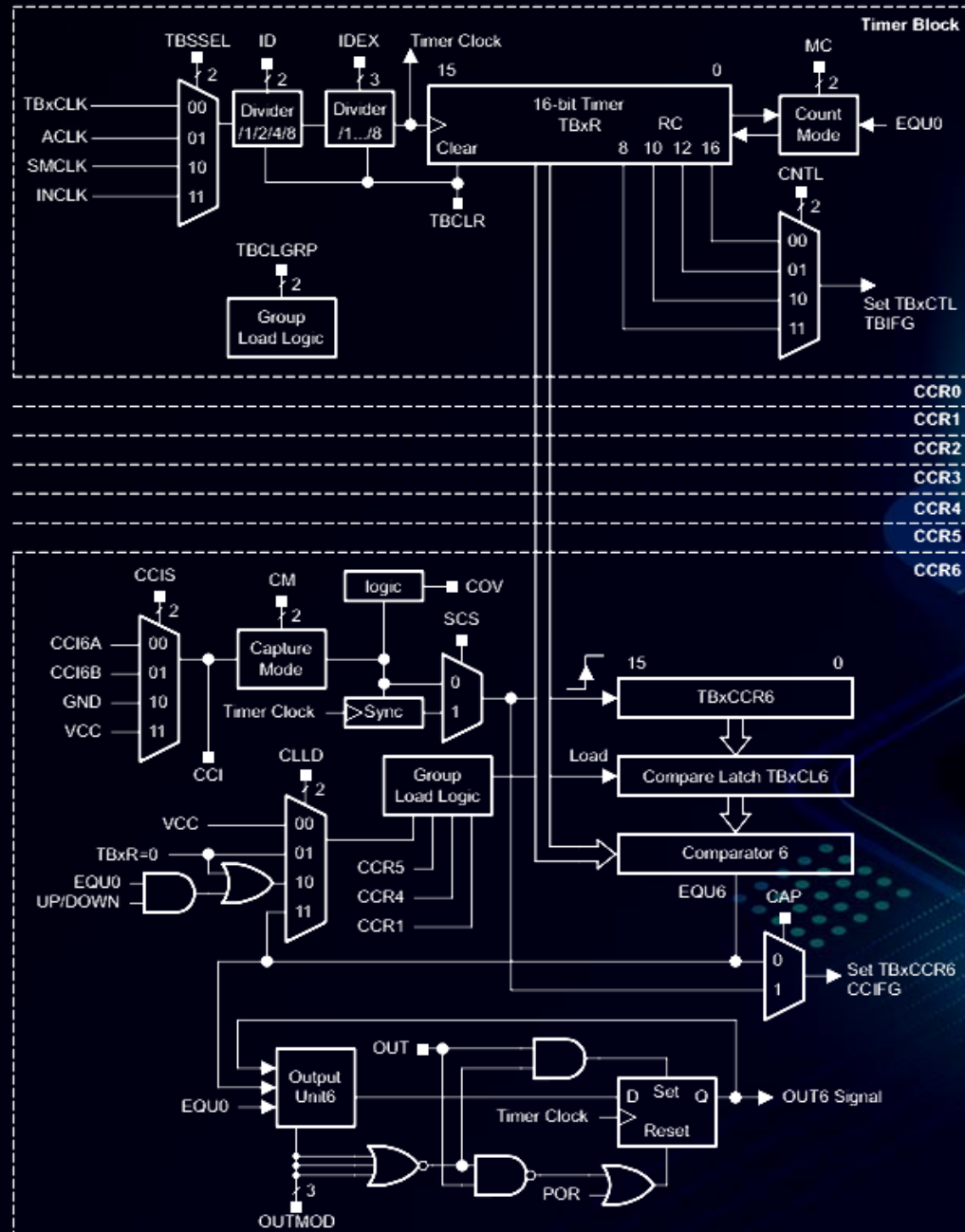


Figure 18-1. Timer_B Block Diagram

קוד – Basic Timer

הקוד הבא מהבהב את לד4 בתדירות של השעון ACLK בעזרת פסיקה.
כאשר ACLK מספק את מקור השעון ל BASIC TIMER.

```
#include <msp430.h>
```

```
int main(void)
```

```
{
```

```
WDTCTL = WDTPW + WDTHOLD; // Stop WDT
```

```
FLL_CTL0 |= XCAP18PF; // Set load cap for 32k xtal
```

```
P5DIR |= 0x02; // Set P5.1 as output
```

```
BTCTL = 0x23; // BTDIV + BT_fCLK2_DIV16 → ACLK/(256*16)
```

```
IE2 |= 0x80; // Enable Basic Timer interrupt – 1000 0000
```

```
__bis_SR_register(LPM3_bits + GIE); // Enter LPM3, enable interrupts
```

```
}
```

```
#pragma vector=BASICTIMER_VECTOR
```

```
__interrupt void basic_timer_ISR(void)
```

```
void __attribute__((interrupt(BASICTIMER_VECTOR))) basic_timer_ISR (void)
```

```
{
```

```
P5OUT ^= 0x02; // Toggle P5.1 with XOR
```

```
}
```

שלב ראשון – הגדרת תדר השעון הבסיסי בעזרת שעון ACLK

$$f_{LED4} = \frac{f_{ACLK}}{256} \cdot \frac{1}{16} = 8 [Hz]$$

קצב פסיקה

$$T_{LED4} = \frac{1}{f_{LED4}} = 125 [msec]$$

זמן מחזור הבהוב הלד –

BTCTL, Basic Timer 1 Control Register

7	6	5	4	3	2	1	0	
BTSEL	BTHOLD	BDIV	BTFRFQ1	BTFRFQ0	BTIP2	BTIP1	BTIP0	
0	0	1	0	0	0	1	1	= 23hex

IE2, Interrupt Enable Register 2

7	6	5	4	3	2	1	0	
BTIE								
1	0	0	0	0	0	0	0	= 80hex

שלב שני – קריאה להבהוב בעזרת הפסיקה שרשמנו

הלד יהבהב כל קריאה לפסיקה בתדר שבחרנו שהוא 8Hz.

```

#include <msp430.h>
int main(void)
{

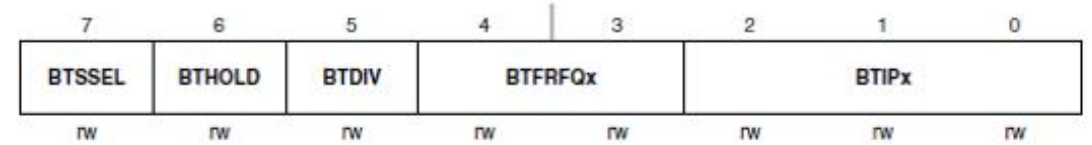
WDTCTL = WDTPW + WDT HOLD; // Stop WDT
FLL_CTL0 |= XCAP18PF; // Set load cap for 32k xtal
P5DIR |= 0x02; // Set P5.1 as output
BTCTL = 0x23; // BTDIV + BT_fCLK2_DIV16 → ACLK/(256*16)
IE2 |= 0x80; // Enable Basic Timer interrupt – 1000 0000

__bis_SR_register(LPM3_bits + GIE); // Enter LPM3, enable interrupts
}

#pragma vector=BASICTIMER_VECTOR
__interrupt void basic_timer_ISR(void)
void __attribute__((interrupt(BASICTIMER_VECTOR))) basic_timer_ISR (void)
{
P5OUT ^= 0x02; // Toggle P5.1 with XOR
}

```

BTCTL, Basic Timer1 Control Register



BTSSSEL	Bit 7	BTCNT2 clock select. This bit, together with the BTDIV bit, selects the clock source for BTCNT2. See the description for BTDIV.
BTHOLD	Bit 6	Basic Timer1 hold 0 BTCNT1 and BTCNT2 are operational 1 BTCNT1 is held if BTDIV=1 BTCNT2 is held
BTDIV	Bit 5	Basic Timer1 clock divide. This bit together with the BTSSSEL bit, selects the clock source for BTCNT2.

BTSSSEL	BTDIV	BTCNT2 Clock Source
0	0	ACLK
0	1	ACLK/256
1	0	SMCLK
1	1	ACLK/256

BTFRFQx	Bits 4–3	f _{LCD} frequency. These bits control the LCD update frequency. 00 f _{ACLK} /32 01 f _{ACLK} /64 10 f _{ACLK} /128 11 f _{ACLK} /256
----------------	----------	--

BTIPx	Bits 2–0	Basic Timer1 interrupt interval 000 f _{CLK2} /2 001 f _{CLK2} /4 010 f _{CLK2} /8 011 f _{CLK2} /16 100 f _{CLK2} /32 101 f _{CLK2} /64 110 f _{CLK2} /128 111 f _{CLK2} /256
--------------	----------	--