

Hardver alapok

(VIII BA01)

Időzítők gyakorlati alkalmazása

Pilászy György gpilaszy@iit.bme.hu

2018/19. tanév őszi félév

- Szoftveres időzítést már láttuk
- Hátrányai:
 - 100% CPU lefoglaltságot eredményez
 - Időzítése függ a processzor órajeléről
 - Ha van engedélyezve hardveres megszakítás, ami többször is lefut, akkor az időzítés pontatlanná válik
- Célszerű lenne külön hardveregységet alkalmazni a pontos időzítésekhez → **TIMER**

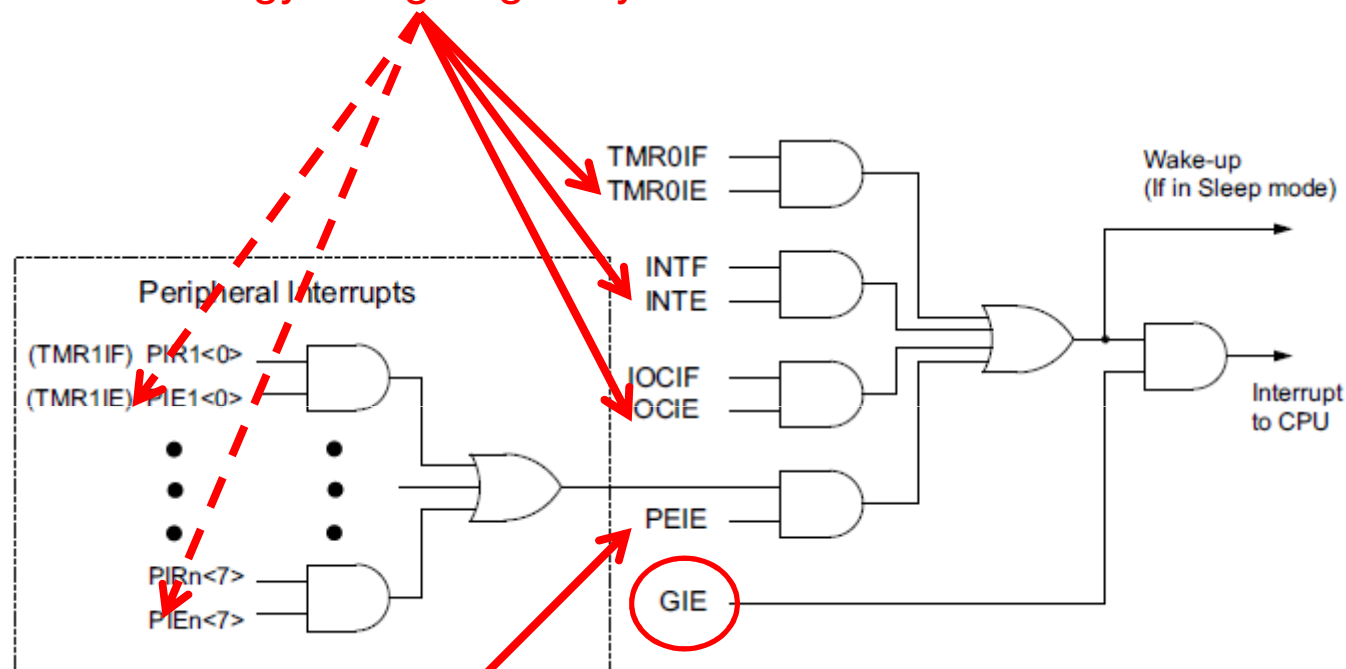


- Láttuk a szoftveres várakozó hurok és szoftveres késleltetés megvalósítását
- Mivel ez 100%-ban lefoglalja a CPU-t feleslegesen, így a CPU nem tud más tevékenységet végezni
- Megoldás: „automatikus szubrutin hívás”
- A gomb megnyomása, vagy az időzítés letelte indítson el egy rutint
→ megszakítási szubrutin

- Sok megszakítási forrás lehet (szinte minden periféria)
- Minden forráshoz külön engedélyezés (IE) és külön jelzőbit (IF)
- Egyetlen vektor → **0x0004** belépési pont
- Programból kell az okot kitalálni (melyik engedélyezett IF aktív)
- Belépéskor környezet mentése, visszatéréskor pedig a környezet visszaállítása → később még foglalkozunk vele...

Megszakítás logika

Minden forrás egyedileg engedélyezhető

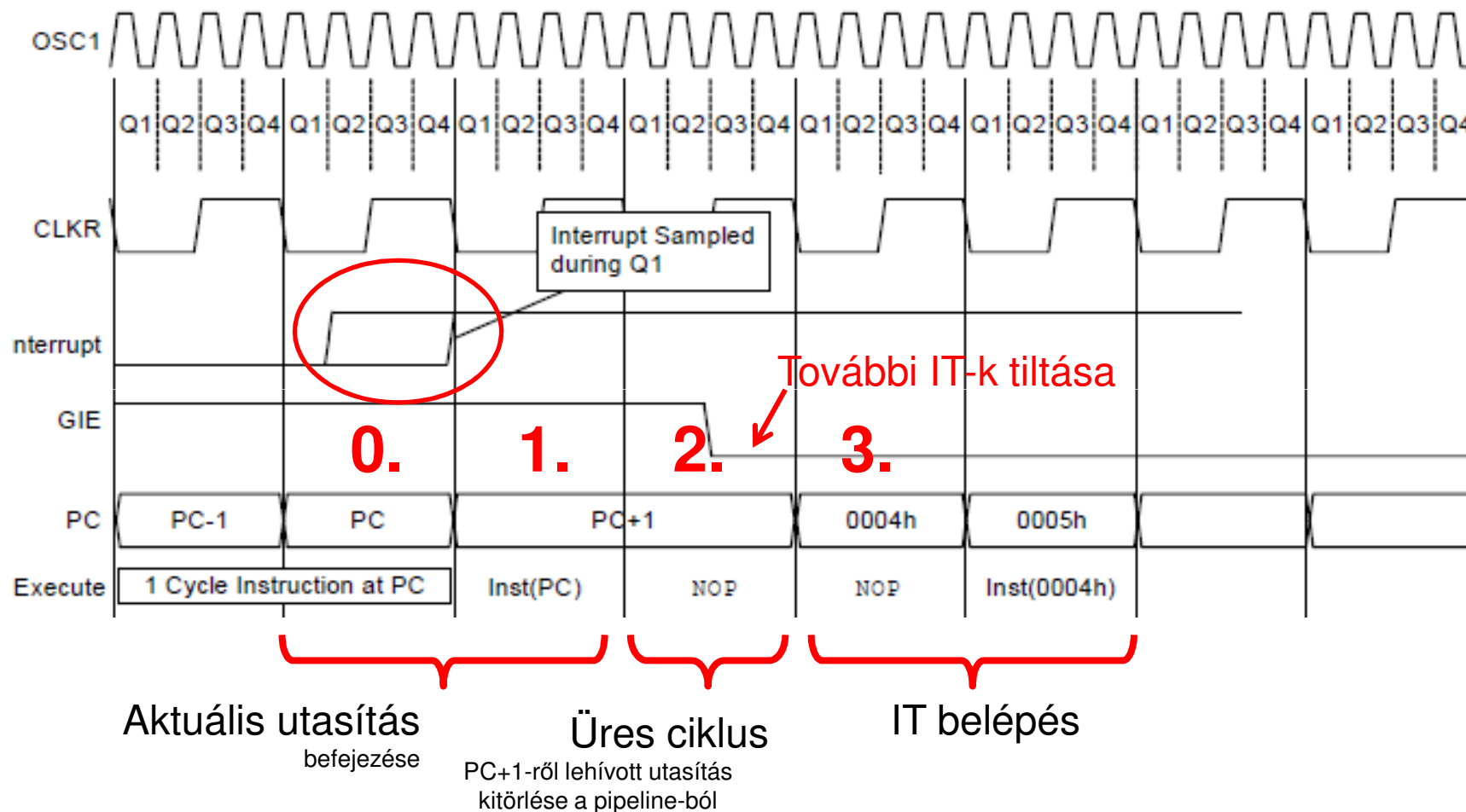


Perifériák külön
is tilthatók

Megszakítások
engedélyezése

Megszakítás időzítése

1 ciklusú utasítás esetén



2 és 3 ciklusú utasítások esetén a lappangási idő 3..5 utasításciklus (CLKR) lehet

Megszakításba lépéskor automatikusan **elmentődnek**:

- W regiszter
- STATUS regiszter
- FSR0 és FSR1 regiszterek
- PCLATH regiszter

Visszatéréskor az elmentett értékek helyreállításra kerülnek (RETFIE utasításra)

Minden más „rombolt” változó mentését a programozónak kell megoldania!

BANK 31	
F80h	Core Registers (Table 3-2)
F8Bh F8Ch	
	Unimplemented Read as '0'
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH
FF0h	Common RAM
	Accesses 70h – 7Fh
FFFh	

- Számláló áramkörök különböző kiegészítőkkel
- 8 vagy 16 bites
- Ciklus rövidítés, jelzőbit az időzítés végén → megszakítást is tud okozni
- Adott frekvencia előállítása
- Változtatható kitöltési tényező előállítása (PWM)
- Bemeneti váltás időpontjának megmérése
- Frekvencia mérés
- Periódus idő mérés



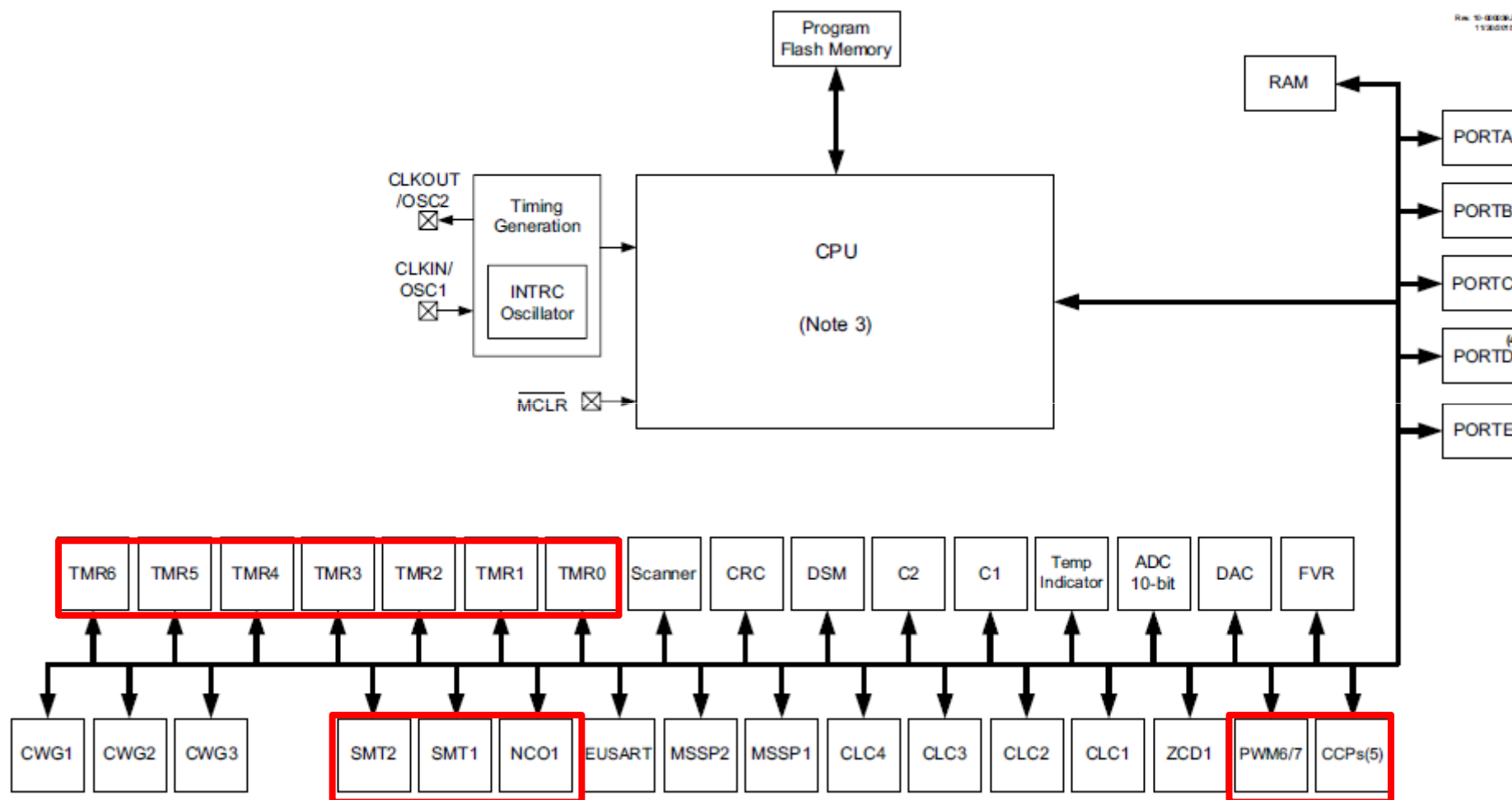
Hardveres időzítők a PIC16F18875-ben

- Watchdog
- 3db 8 bites timer (TMR2, TMR4, TMR6)
- 4db 16 bites timer (TMR0, TMR1, TMR3, TMR5)
- 2db SMT (Signal Measurement Timer, 24 bites)

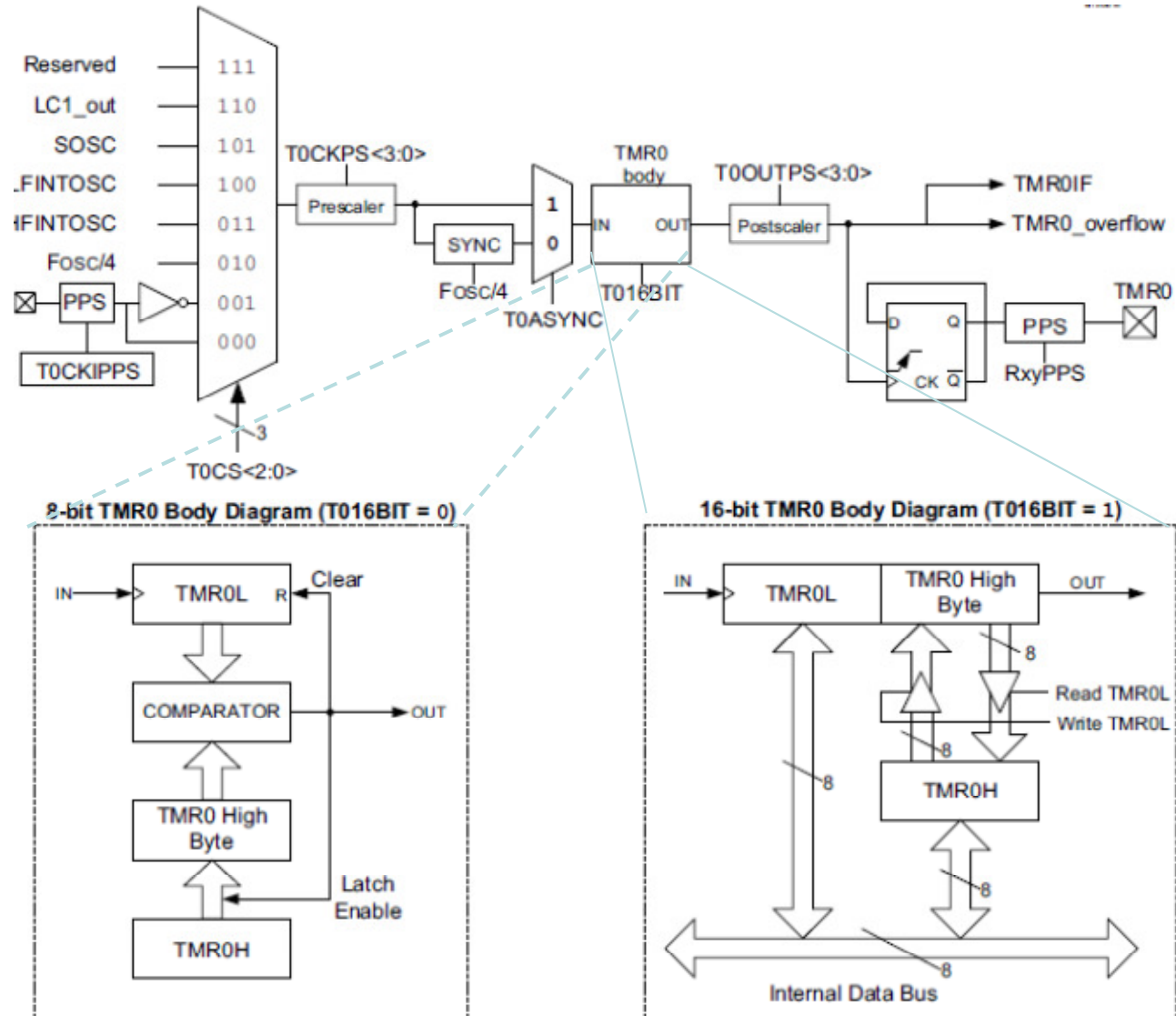
Időzítőt használó perifériák:

- 5db CCP modul (16 bites mérés, 10 bites PWM)
- 2db 10 bites PWM (Pulse Width Modulator) modul
- NCO (Numerically Controlled Oscillator)

Hardveres időzítők a PIC16F18875-ben



TMR0



REGISTER 27-1: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
bit 7				bit 0			

- Választható 8 vagy 16 bites működés
- Programozható utóosztó 1:1...1:16 között

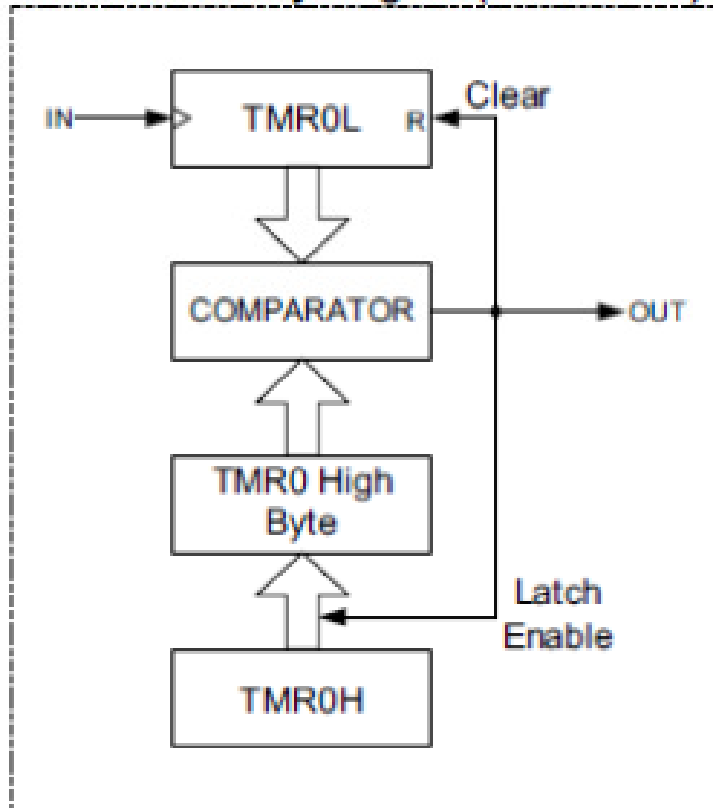
REGISTER 27-2: T0CON1: TIMER0 CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0CS<2:0>			T0ASYNC	T0CKPS<3:0>			
bit 7				bit 0			

- Választható órajel forrás
- Működése szinkronizálható az utasítások végrehajtásához
- Programozható előosztó 1:1...1:32768
- Túlcsordulás kimenet portlábba is kivezethető

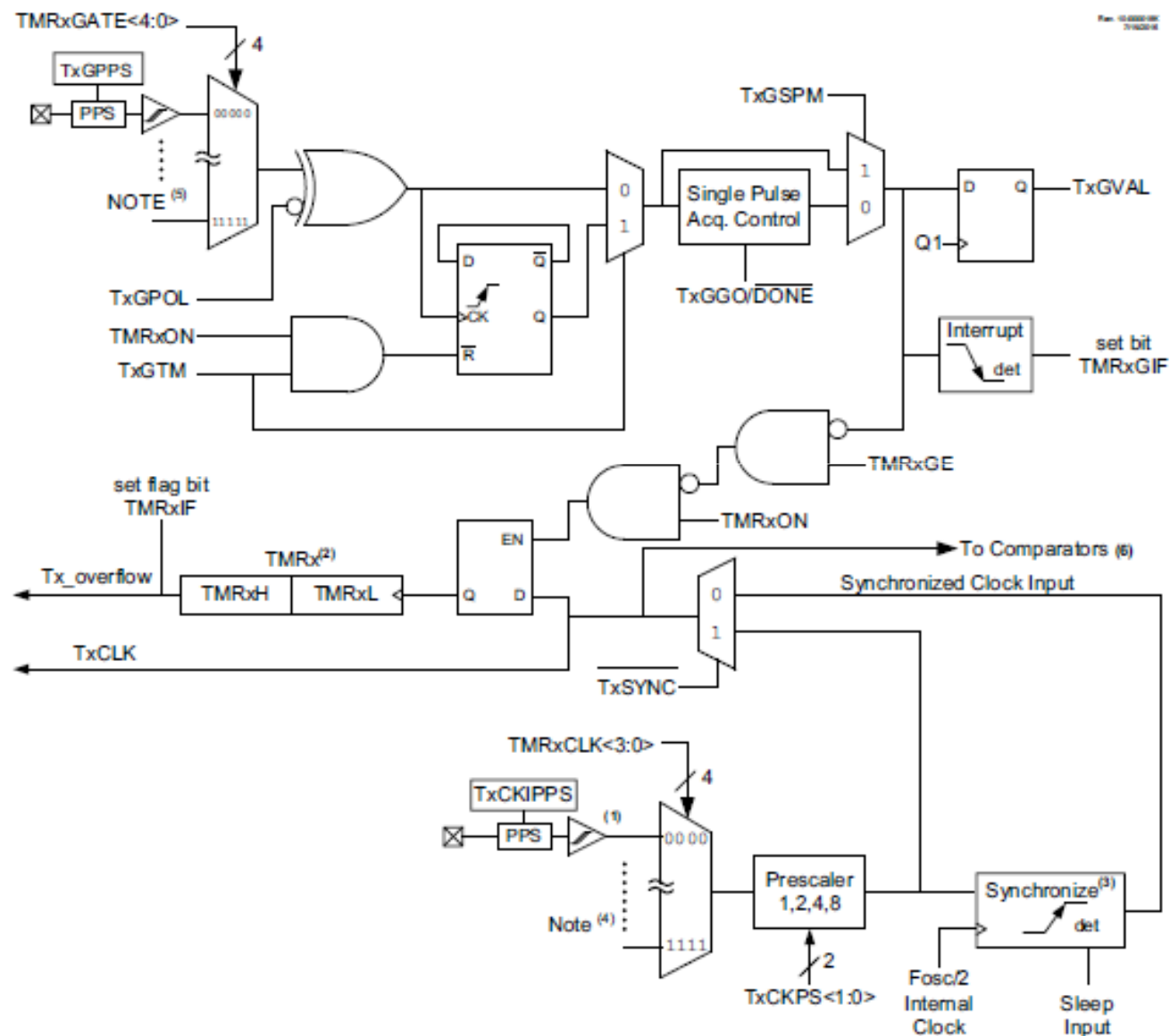
TMR0-automatikus újratöltés mód

8-bit TMR0 Body Diagram (T016BIT = 0)



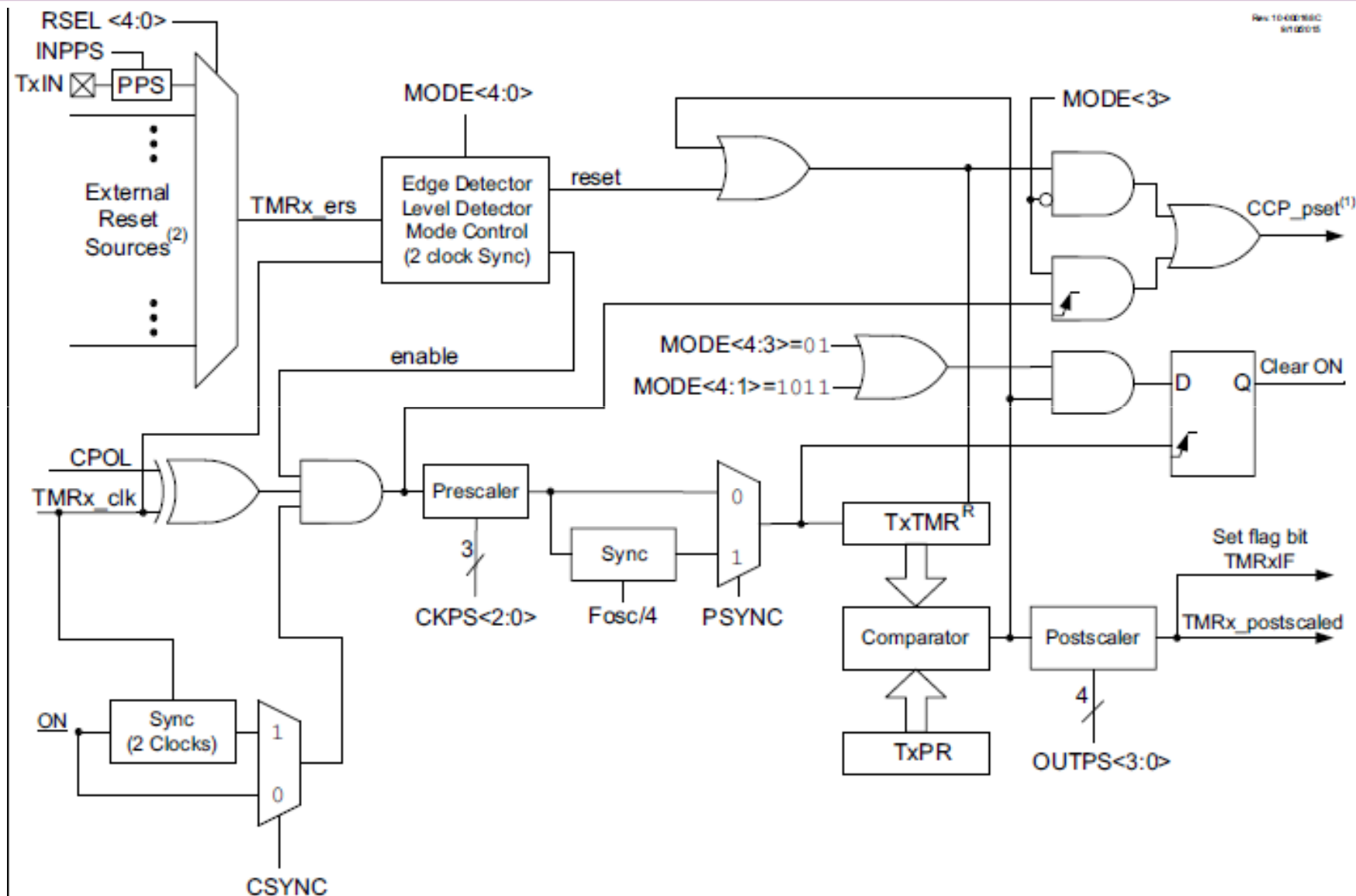
- Felfelé számlálás
- Beállított érték elérésekor törlés+kimenet állítása
- Alkalmazási példák:
 - Órajel előállítás külső eszköznek
 - Program ütemezés
 - Jelzőhang generálás

TMR1, 3, 5



- Felfelé számlálás
- 16 bites időzítő/számláló
- Választható belső vagy külső órajel forrás
- 2 bites előosztó
- Túlcsoordulásakor megszakítás kérés
- Választható számlálás engedélyezés (Gate)

TMR2, 4, 6



- 8 bites számláló
- 8 bites periódus regiszter
- Választható külső törlési lehetőségek
- Programozható előosztó (1:1...1:128)
- Programozható utóosztó (1:1...1:16)
- Választható órajel források
- Megszakítás kérés a periódus végén
- Működési módok:
 - Szabadon futó mód
 - Egyszer lefutó mód

- Program ütemezés
- Óra
- Órajel előállítás külső egységnek
- Külső jel frekvencia mérése
- Külső jel periódusának mérése
- Külső jel kitöltési tényezőjének mérése
- Külső pulzusok számolása



Program ütemezése TMR0-al

Példa: késleltessünk 10ms-időt a TMR0-al

$$100Hz = 8000000Hz \cdot \frac{1}{32} \cdot \frac{1}{250} \cdot \frac{1}{10}$$

Regiszter név	Érték (binárisan)	Megjegyzés
T0CON0	B'1000 1001'	Bekapcsolás + 1:10 utóosztás
T0CON1	B'0100 0101'	Fosc/4 + 1:32 előosztás
TMR0H	B'1111 1001'	0...249 között számol

REGISTER 27-1: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
bit 7				bit 0			

REGISTER 27-2: T0CON1: TIMER0 CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0CS<2:0>			T0ASYNC	T0CKPS<3:0>			
bit 7				bit 0			



Várakozás timerrel

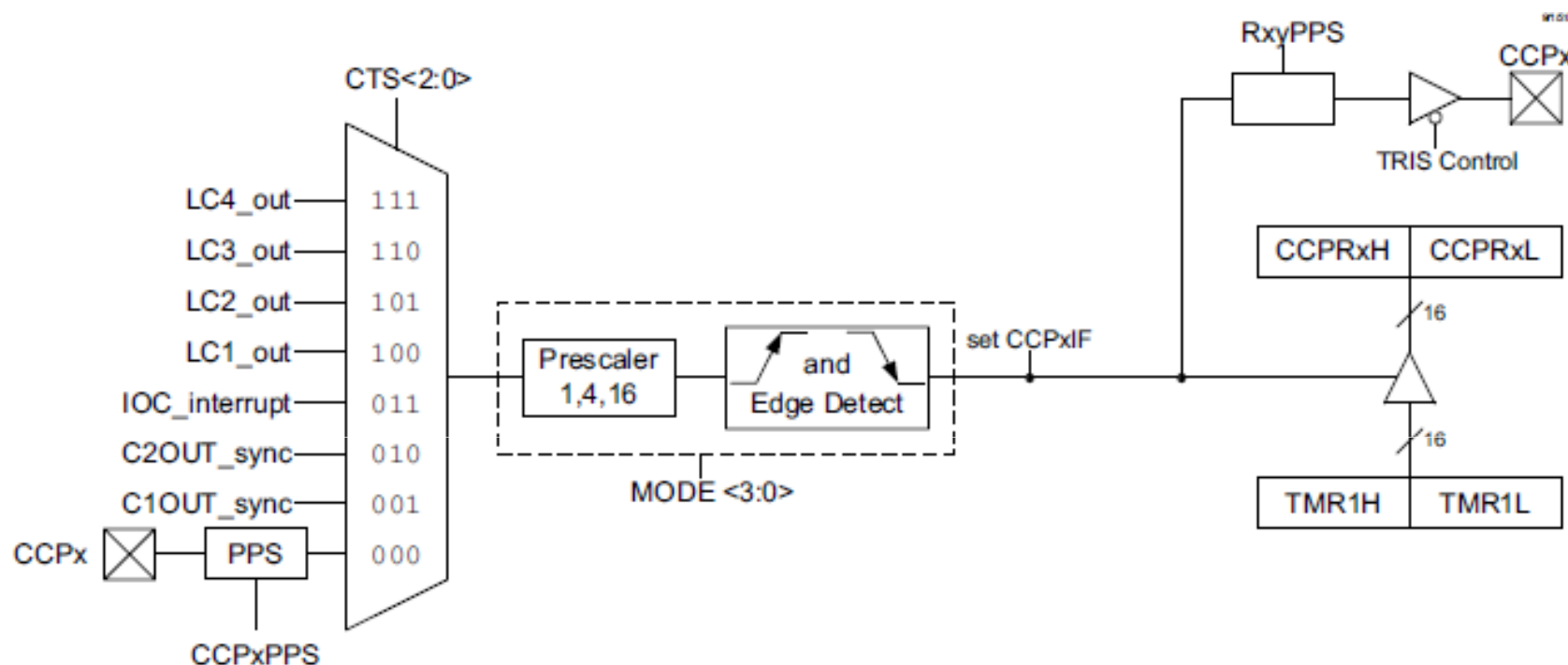
TMR0INIT:

```
MOVLW    B'10001001'  
MOVWF    T0CON0  
MOVLW    B'01000101'  
MOVWF    T0CON1  
MOVLW    D'249'  
MOVWF    TMR0H  
RETURN
```

Delay10ms:

```
CLRF     TMR0L  
BANKSEL  PIR0           ; BANK7  
BCF      PIR0,TMR0IF  
BTFSS    PIR0,TMR0IF ; sw várakozás TMR flagbitre  
GOTO     $-1  
BANKSEL  TMR0L ; BANK0  
RETURN
```

Input capture



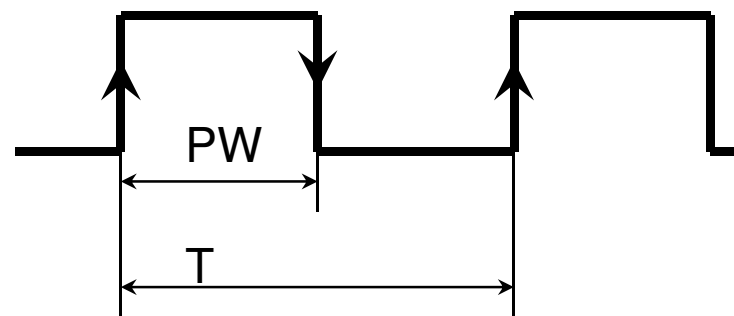
Adott (beállítható) él hatására elteszi a számláló pillanatnyi értékét annak leállítása nélkül:
 $CCPRxH:CCPRxL = TMR1H:TMR1L$

- Impulzus hosszának megmérése

$$PW = CAP2\uparrow - CAP1\downarrow$$

- Jel periódus idejének mérése

$$T = CAP2\uparrow - CAP1\uparrow$$

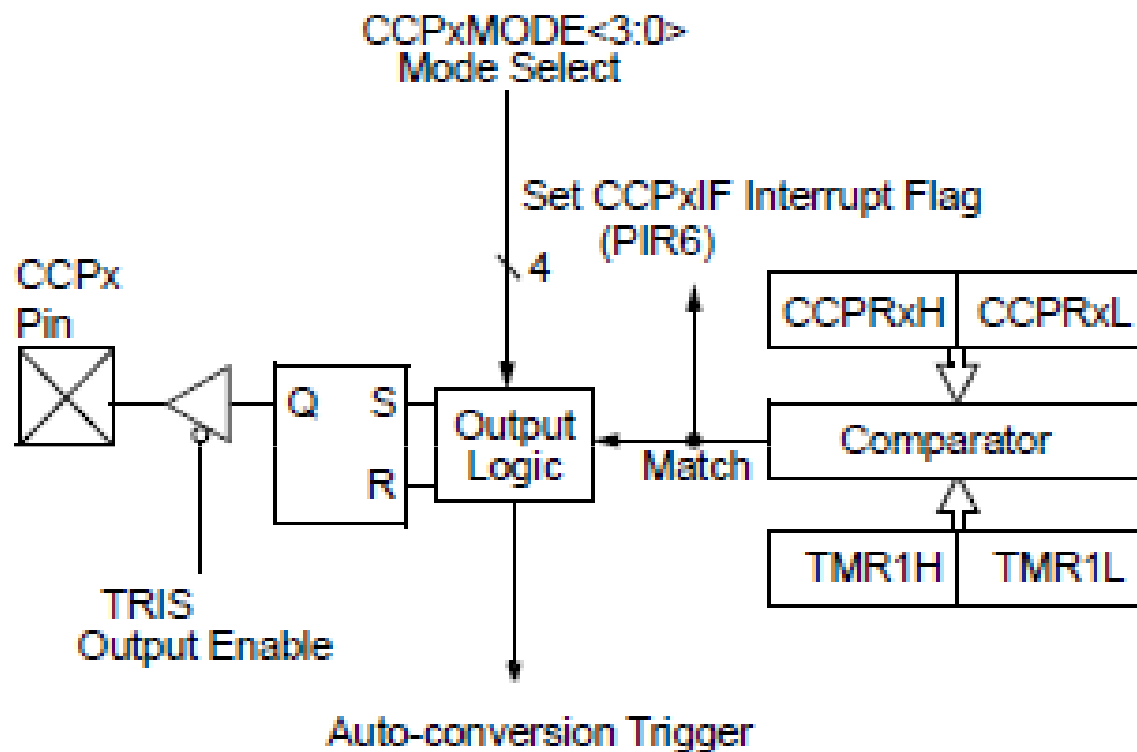


- Frekvencia mérés N darab minta után:

$$F = \frac{N}{Tösszes} = \frac{darabszám}{össz.idő}$$

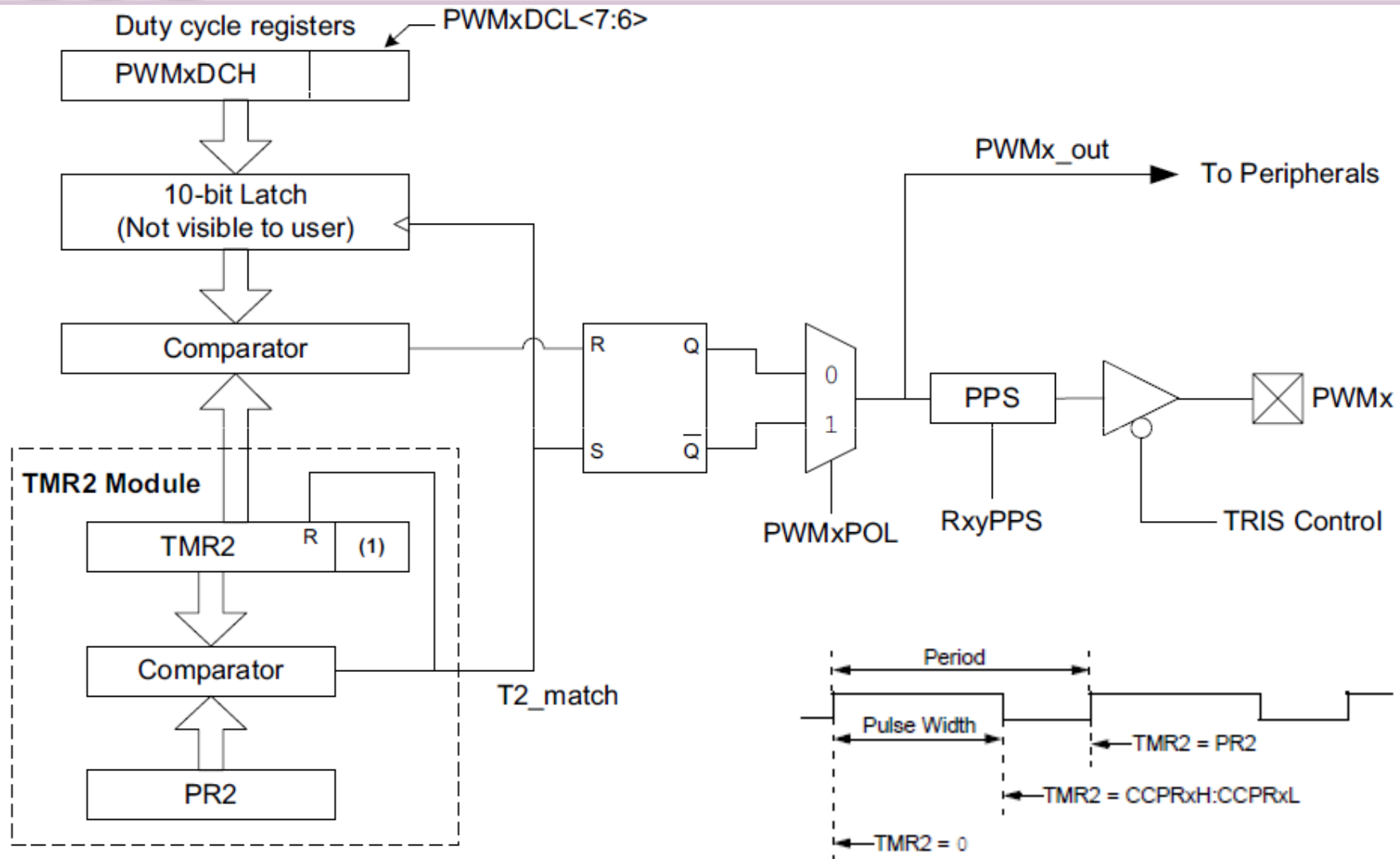
- Időbélyeg előállítása kommunikációhoz

Output compare



- Kimenet beállítása a komparátor jelzésére
- Adott hosszúságú impulzus hardveres előállítása

PWM jel előállítás





PWM előállítás

- Külön állítható a periódus (PR2)
és külön a kitöltés (CCPRxH:CCPRxL)
- Ciklus elején kimenet 1-be áll
- Kitöltés elérésekor kimenet 0-ba áll
- Alkalmazási példák:
 - D/A átalakítás (külső szűrés!)
 - Kijelző fényerő állítása
 - PWM jel továbbítás (pl.: rádió csatornán)
 - DC motor sebességének változtatás
(pl.. PC ventilátor)



PWM bekapcsolása

PWMINIT:

```
BANKSEL    T2CON
MOVLW      H'01'
MOVWF      T2CLKCON ;timer2 órajele fosc/4 -> 8MHz
MOVLW      B'10100000' ;on, elosztó(3 bit), utóosztó (4 bit)
MOVWF      T2CON    ;T2 elosztó 1:4-es -> 2MHz
MOVLW      D'255'
MOVWF      T2PR      ;a timer2 255-ig számol, majd újraindul (8MHz/4/255=7,8kHz)
MOVLW      D'0'
BANKSEL    PWM7CON
MOVWF      PWM7DCH ;kezdeti kitöltés 0%
BSF        PWM7CON,PWM7EN ;PWM modul indul
BANKSEL    RC2PPS
MOVLW      H'0F'      ; PWM7output
MOVWF      RC2PPS     ; PORT RC2-re

BANKSEL    LATA
BCF        TRISC,2      ; RC2 portbit kimenetre
RETURN
```



PWM használata

PWMSET : ; kitöltés beállítása W szerint 0..255

BANKSEL PWM7DCH

MOVWF PWM7DCH

BANKSEL LATA

RETURN

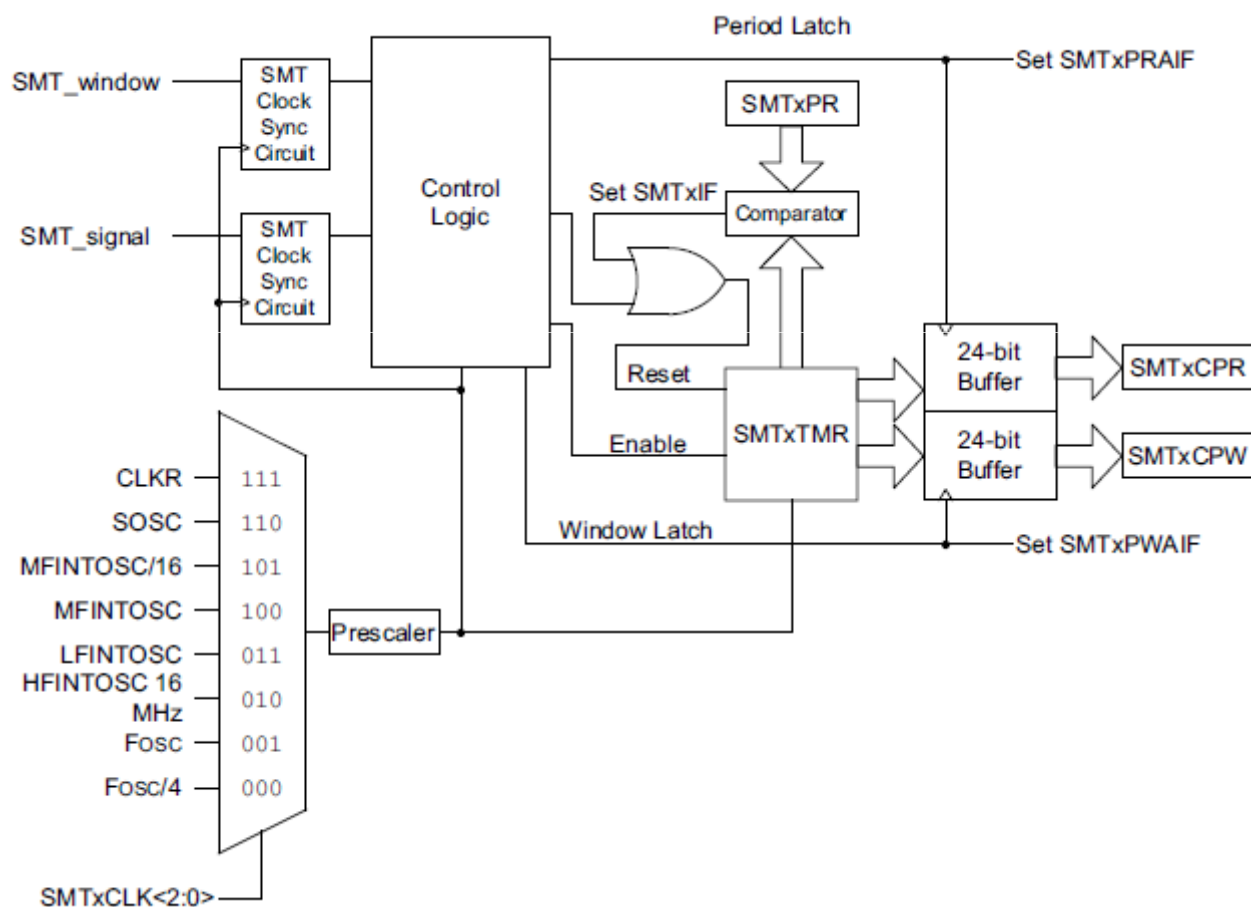
Rev. 10-01103
2/0/06

- A külső órajel ütemében (NCOx_clk)
- egy regiszter (NCOxACCU) értékét növeli
- megadható lépésközzel (NCOxINC)
- 20 bites
- Működhet fix kitöltési tényezővel vagy
- Megadott impulzusszélességgel
- A kimeneti frekvencia nagy felbontással állítható

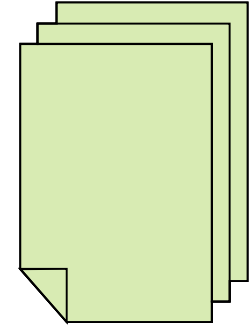
$$\Delta = \frac{\text{NCOx_clk}}{2^{20}}$$

SMT modul

Signal measurement timer



- 24 bites időzítő
- Precíziós mérési feladatok időzítéséhez
- Képes megszakítást okozni
- Más perifériákat is tud működtetni (indítani)



- Microchip weboldala
<http://www.microchip.com>
- MPLABX fejlesztői környezet és dokumentáció letöltése
<https://www.microchip.com/mplab/mplab-x-ide>
- Online help a mikrokontrollerhez és a fejlesztői környezethez
<http://microchipdeveloper.com/mcu1102:start>
- Tárgy honlapján
<https://www.iit.bme.hu>