

Lucrarea . Lucrul cu canale de timp

Obiectiv

Deprinderea modului de programare a canalelor de timp.

1. Introducere

Canalele de timp sunt periferice de bază ale oricărui microcontroller. În familia dsPIC33F/PIC24H sunt incluse mai multe canale de timp ce operează pe 16 biți. Microcontrollerul dsPIC33FJ32MC302 conține 5 astfel de module, notate de la Timer 1 la Timer 5.

Aceste canale sunt clasificate în 3 categorii :

- Canalul de timp de tip A (Timer 1);
- Canalul de timp de tip B (Timer 2, Timer 4);
- Canalul de timp de tip C (Timer 3, Timer 5).

Un canal de timp de tip B poate fi concatenat cu unul de tip C pentru a forma un timer pe 32 biți iar canalele de tip C pot starta conversii A/D.

Fiecare canal de timp funcționează pe 16 biți, poate fi folosit ca numărător de evenimente sau ca temporizator și conține următoarele registre regăsite și în Fig.1:

- Registrul de numărare (TMRx);
- Registrul de perioadă (PRx);
- Registrul de control pentru configurarea timer-ului (TxCON).

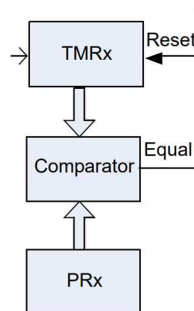


Fig.1 : Elementele principale ale ceasului „Timer 1”

Valoarea din registrul numărătorului, TMRx, se incrementează la fiecare perioadă a semnalului de ceas, această valoare fiind comparată în permanență cu valoarea din registrul de perioadă. În momentul în care cele două valori devin egale, se setează flag-ul întreruperii timer-ului x. Această funcționare este schițată în Fig.2.

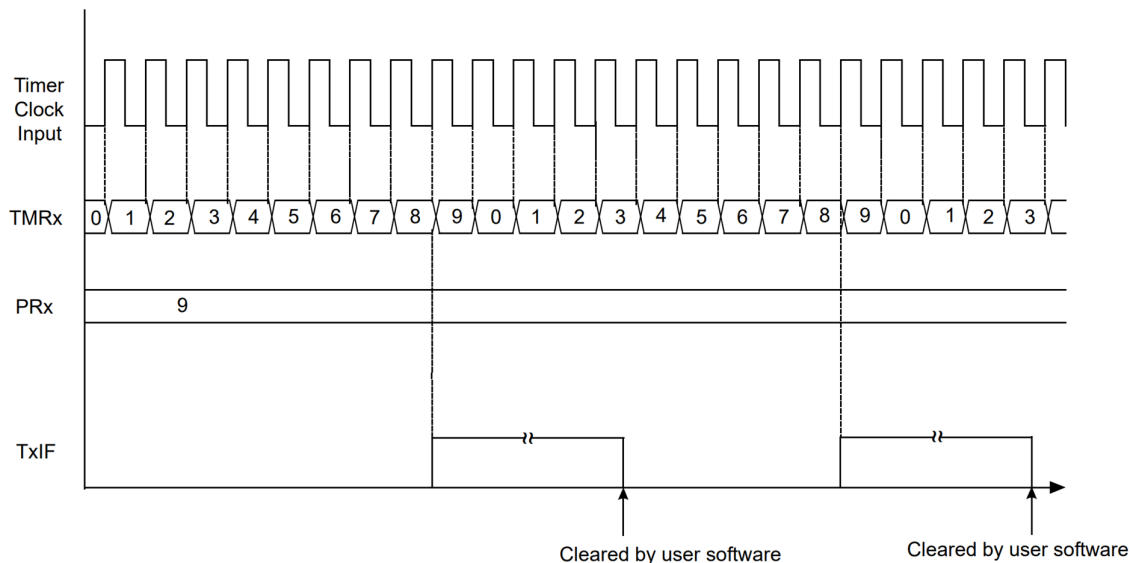


Fig.2 Timpul de întrerupere pentru potrivirea perioadei de ceas

Fiecărui timer îi sunt asociați și următorii biți de control ai întreruperilor :

- Bitul de validare a întreruperii (TxIE din registrul IEC0 si IEC1);
- Indicatorul de întrerupere (TxIF din registrul IFS0 si IFS1);
- Biții de control a priorității întreruperii (TxIP<2:0> din registrele IPCy).

Canalele de timp funcționează ca temporizatoare având frecvența derivată de la ceasul intern (FCY), iar ca numărătoare având frecvența derivată de la o sursă de ceas extern primită la pinul TxCK.

În mod uzual, pentru a comanda numărătorul se folosesc mai multe surse de ceas.

La laborator se folosește oscilatorul de mare viteză, de 7,37 MHz, dar se poate folosi în combinație cu modulul PLL, studiat la o temă anterioară. Astfel se poate obține o frecvență maximă de 80 MHz, frecvența de implementare a instrucțiunilor și a ceasului de magistrală ajungând la valoarea de 40MHz. Perioada ceasului de magistrală, atunci când se folosește modulul PLL, ajunge la 25 ns, așa cum se arată în exemplele următoare:

Fără modul PLL:

$$F_{FRC} = 7,37 \text{ MHz} \Rightarrow F_{OSC} = 7,37 \text{ MHz}$$

$$F_{CY} = \frac{F_{OSC}}{2} = 3,685 \text{ MHz}$$

$$T_{CY} = \frac{1}{F_{CY}} \cdot 10^{-6} = \frac{1}{3,685} \cdot 10^{-6} = 271 \text{ ns}$$

Cu modul PLL:

$$F_{FRC} = 7,37 \text{ MHz} \Rightarrow F_{OSC} = 80 \text{ MHz}$$

$$T_{CY} = \frac{1}{F_{CY}} \cdot 10^{-6} = \frac{1}{40} \cdot 10^{-6} = 25 \text{ ns}$$

2. Canalul de timp de tip A

Diagrama bloc a Timer-ului 1 este prezentată în Fig.3, în care sunt marcate cu roșu componentele principale care stau la baza funcționării canalului de timp, iar cu verde este marcat prescaler-ul care are drept scop măsurarea de perioade mai mari.

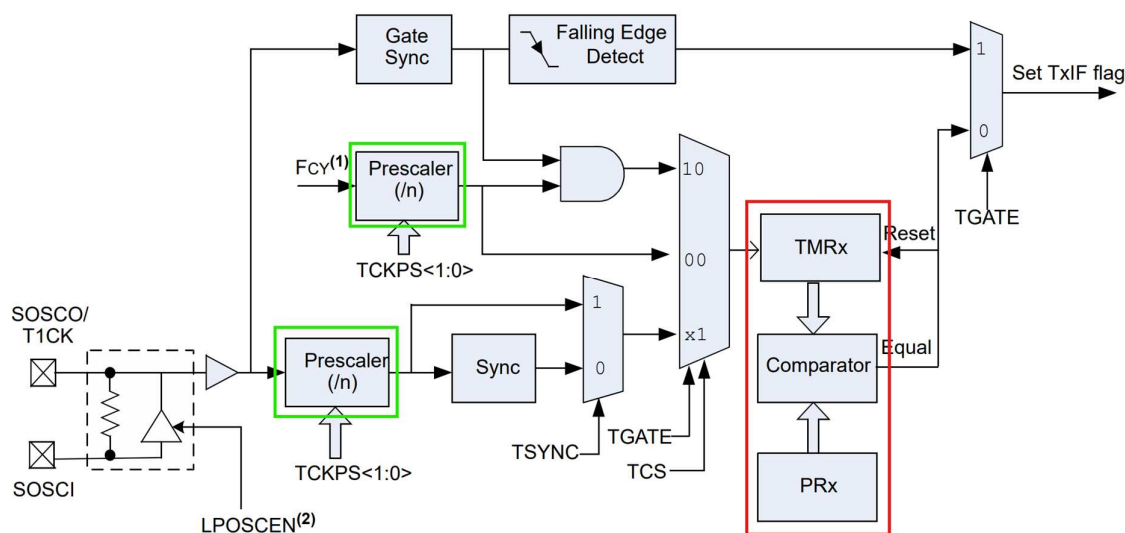


Fig.3 : Structura circuitelor de configurare pentru ceasul „Timer 1”

3

funcționa în starea de SLEEP sau IDLE a microcontrollerului și generează o cerere de întrerupere când valorile din registrii TMR1 și PR1 sunt egale.

Modurile de funcționare ale Timer-ului 1 pot fi selectate prin intermediul a 3 biți din registrul de control TICON :

- Bitul de control al sursei de ceas (TCS);
- -Bitul de control al sincronizării (TSYNC);
- Bitul de control al modului „gated” (TGATE).

Moduri de configurare a ceasului Timer 1 :

Mod	TCS	TGATE	TSYNC
Temporizator	0	0	x
Gated Timer	0	1	x
Numărător sincron	1	x	1
Numărător asincron	1	x	0

3. Canalele de timp de tip B și de tip C

Timer 2 și Timer 4 sunt canale de timp de tip B și pot fi concatenate cu un canal de tip C pentru a forma un timer pe 32 biți.

Pentru aceste timere semnalul de ceas extern de la bitul TxCK este întotdeauna sincronizat cu semnalul de ceas intern, sincronizare ce se realizează după divizarea cu prescaler-ul.

Diagrama bloc pentru canalele de tip B este prezentată în Fig.4.

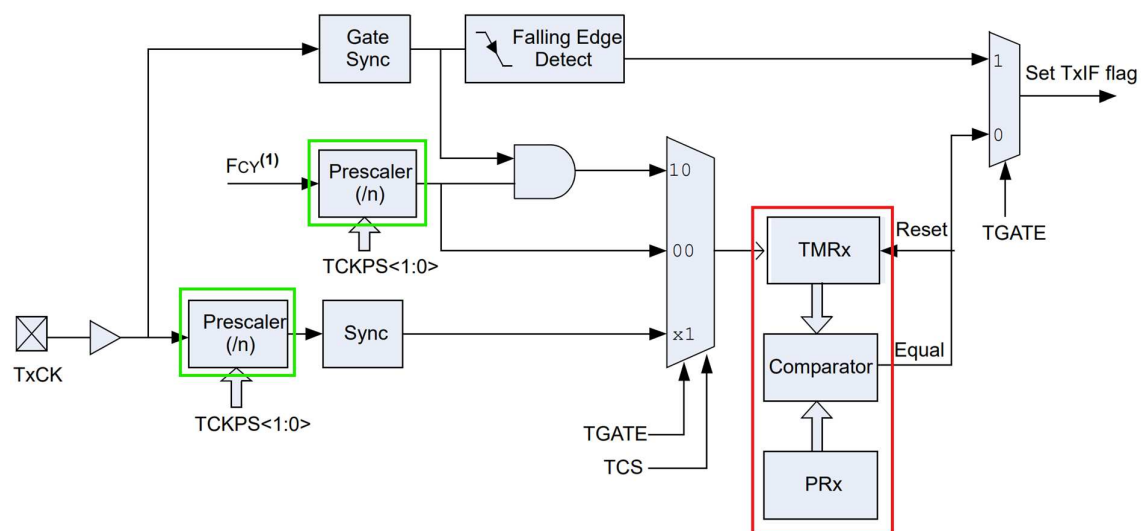


Fig.4 : Structura circuitelor de configurare pentru ceasul de tip B ($x = 2$ sau 4)

Timer 3 și Timer 5 sunt canale de timp de tip C și pot fi concatenate cu un canal de tip B pentru a forma un timer pe 32 biți.

Pentru timerele de tip C semnalul de ceas extern de la bitul TxCK este întotdeauna sincronizat cu semnalul de ceas intern, sincronizare ce se realizează înainte de divizarea cu prescaler-ul. Totodata cel puțin un canal de tip C poate starta conversia A/D, așa cum se poate observa și în marcajul albastru din Fig.5.

Diagrama bloc pentru canalele de tip C este prezentată în Fig.5.

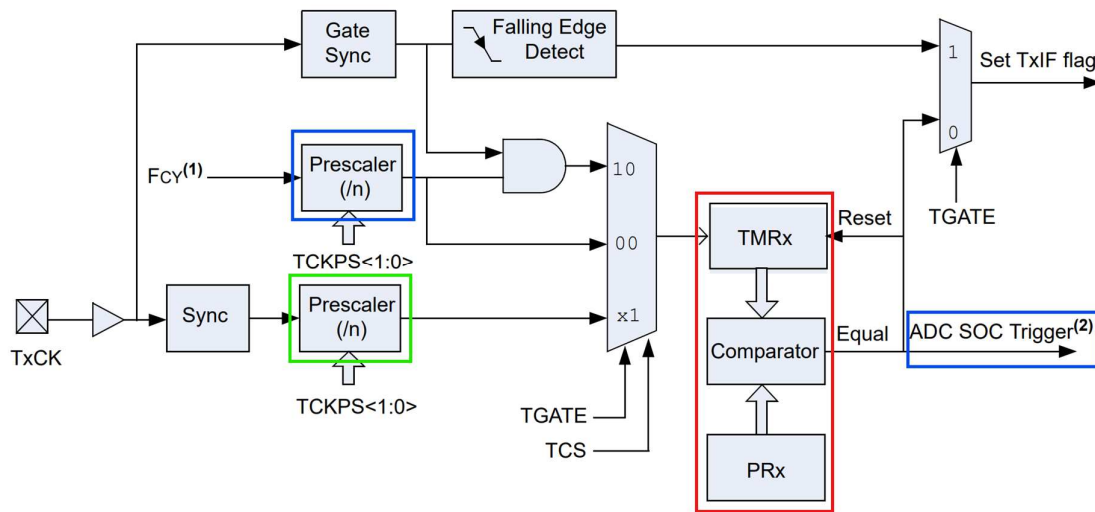


Fig.5 : Structura circuitelor de configurare pentru ceasul de tip C (x = 3 sau 5)

Atât timerele de tip B cât și cele de tip C pot fi configurate să funcționeze ca temporizator sau ca numărător sincron după cum este prezentat în tabelul de mai jos.

Moduri de configurare a ceasurilor Timer 2/3 și Timer 4/5 :

Mod	TCS	TGATE
Temporizator	0	0
Gated timer	0	1
Numărător sincron	1	x

Exemple de calcul pentru implementarea diferitor valori ale perioadei, utilizând modulul PLL ($F_{OSC} = 80\text{MHz}$, $T_{CY} = 25\text{ns}$):

Ex.1 Pentru o perioadă de $100\text{ }\mu\text{s}$:

$$n = \frac{T}{T_{CY}} = \frac{100\mu\text{s}}{25\text{ns}} = \frac{100 \cdot 10^3}{25} = 4 \cdot 10^3 = 400 \leq 65535$$

Deoarece valoarea obținută este mai mică decât 65535, nu există necesitatea folosirii unui prescaler.

Ex.2. Pentru o perioadă de 100 ms:

$$n = \frac{T}{T_{CY}} = \frac{100ms}{25ns} = 4 \cdot 10^6 = 4000000 \geq 65535$$

Valoarea obținută este cu mult mai mare decât 65535, iar în acest caz va trebui să fie folosit un prescaler. Pentru simplitate atribuim prescaler-ului valoarea maximă, 1:256, astfel:

$$4000000:256 = 15625 \leq 65535$$

După aceste calcule, constanta care trebuie să fie încărcată în registrul de perioadă PRx este 15625, cu condiția ca în grupul de biți de configurare TCKPS să se scrie faptul că se folosește un prescaler de 1:256.

Ex: T2CONbits.TCKPS = 0b11

4. Canalul de timp pe 32 biți

Canalul de timp pe 32 biți este rezultatul concatenării unui timer de tip B cu unul de tip C. Modurile în care se poate realiza această concatenare sunt alăturarea Timer-elor 2 și 3, respectiv a Timer-elor 4 și 5. De obicei se utilizează prima variantă pentru obținerea unui ceas pe 32 biți.

- Acest canal poate funcționa ca temporizator sau ca numărător sincron și poate realiza o conversie A/D, sau utiliza un prescaler pentru divizarea tactului inițial, poate funcționa în starea de IDLE a microcontrollerului, sau poate genera o cerere de întrerupere atunci când registrul concatenat TMR3/TMR2 are aceeași valoare cu cea a registrului setat inițial PR3/PR2.
- Pentru operațiile pe 32 biți, bitul de control T32 (TxCON<3>) al timer-ului de tip B trebuie setat. Canalul de tip B conține cei mai puțin semnificativi biți pentru operațiile pe 32 biți, în timp ce canalul de tip C îi conține pe cei mai semnificativi.
- Pentru configurarea acestui canal sunt necesari doar biții din registrul de control al timer-ului de tip B, cei ai timer-ului de tip C sunt ignorați cu excepția bitului pentru selectarea stării de IDLE, TSIDL (TyCON<13>).
- Pentru controlul întreruperilor sunt utilizați doar biții de control ai timer-ului de tip C. Biții de control ai întreruperilor și cei de stare ai ceasului de tip B sunt ignorați în timpul operațiilor pe 32 biți.

Diagrama bloc a canalului pe 32 biți este prezentată în Fig.6.


```

#include "p33fxxx.h"

void Init_Timer1( void )
{
    T1CON = 0;

    T1CONbits.TCKPS = 0b11;    // Selectare prescaler 1:256

    TMR1= 0;
    PR1 = 14740;

    _T1IF = 0;
    _T1IP = 1;
    _T1IE = 1;

    T1CONbits.TON = 1;
}

void __attribute__ ((interrupt, no_auto_psv)) _T1Interrupt( void )
{
    _RB15 = ~_RB15;
    _T1IF = 0;                // Resetare flag de intrerupere
}

int main( void )
{
    TRISB = 0x0000;          // Setare PORTB ca iesire
    PORTB = 0xF000;          // Setare valori pini

    Init_Timer1();           // Configurare Timer 1

    while(1)
    {
    }
}

```

Probleme propuse :

1. Să se modifice programul de la Exemplul 1 astfel încât comutarea pinului RB15 să fie făcută la interval de 350us.
2. Să se realizeze un program care să comute starea pinului RB14 la interval de 10 secunde. Se va folosi un temporizator pe 32 biți.