Laborator 8 Circuite Timer. Întreruperi

În laboratorul curent se va studia circuitul periferic timer al familiei de microprocesoare PIC32. Va fi prezentat și sistemul de întreruperi al microprocesorului folosit de circuitul timer.

8.1 Circuite Timer PIC32

Circuitele timer sunt utile într-un sistem deoarece pot genera evenimente periodice precise pe bază de întreruperi pentru aplicațiile software. Alte utilizări includ numărarea impulsurilor externe sau măsurarea duratei evenimentelor externe prin utilizarea funcției de oprire a circuitului atunci când valoarea semnalului măsurat este diferită de un nivel logic ce poate fi configurat (*timer gate*).

Microcontrolere familiei PIC32 au în general două tipuri de circuite timer:

- Tip A (cronometru/contor sincron/asincron pe 16 biți cu funcție de închidere)
- Tip B (cronometru/contor sincron pe 16 sau 32 biți cu funcție de închidere și eveniment special de declanșare

Ambele tipuri de circuit au următoarele caracteristici comune:

- Registru contor de 16 biți
- Sursă de ceas internă sau externă selectabilă prin configurare software
- Generare programabilă de întreruperi cu prioritate
- Contor de impulsuri externe cu funcție de închidere

În afară de caracteristicile comune de mai sus, fiecare tip de circuit oferă caracteristici suplimentare prezentate în continuare.

8.2 Circuite timer PIC32 de Tip A

Circuitele timer de tip A au la dispoziție, pe lângă semnalul de ceas T_{PBCLK}, un al doilea oscilator încorporat. Acest lucru le permite să fie operaționale chiar și atunci când microprocesorul este în modul sleep.
Circuitele oferă un divizor preliminar de frecvenţă (selectabil din software) cu un raport de 1:1, 1:8, 1:64 sau 1:256 din frecvența de referință.

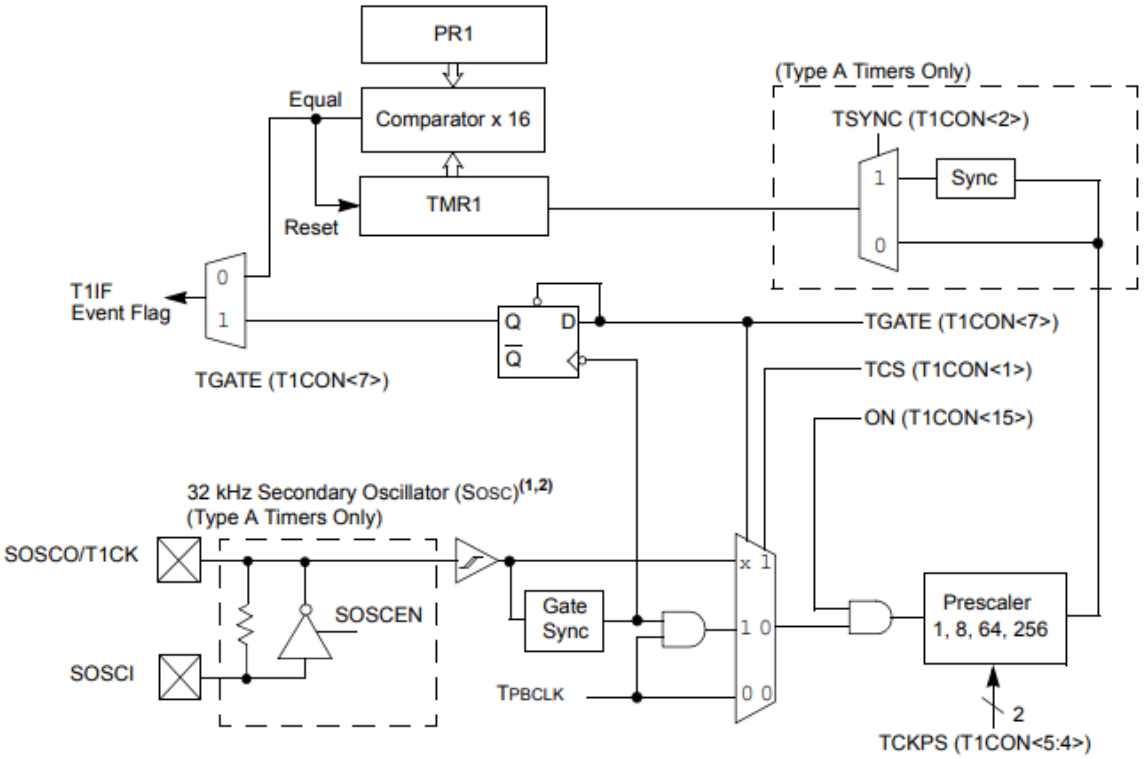


Figura 1 Circuitul timer tip A, schema bloc

8.2.1 Regiștrii de control și de întrerupere

Circuitul TIMER conține regiștrii speciali pentru controlul și stocarea informațiilor legate de circuit. Rezumatul acestor regiștrii poate fi observat în tabelul de mai jos. Pentru funcționalitatea completă a acestor regiștrii se poate studia manualul familiei PIC32, capitolul 14 (link la sfârșitul laboratorului)

Register Name(1)	Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
T1CON	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ON	—	SIDL	TWDIS	TWIP	—	—	—
	7:0	TGATE	—	TCKPS<1:0>		—	TSYNC	TCS	—
TMRx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	TMRx<15:8>							
	7:0	TMRx<7:0>							
PRx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	PRx<15:8>							
	7:0	PRx<7:0>							

T1CON

Acest registru controlează activarea timerului, sursa de ceas, modul de operare sincron/asincron și divizorul inițial de frecvență.

bit 15 ON: Timer On bit

1 = timerul este pornit

0 = timerul este oprit

TCKPS<1:0>: divizor de frecvență

11 = 1:256

10 = 1:64

01 = 1:8

00 = 1:1

TMR1

Acest registru conține valoarea curentă a timerului, este actualizată cu fiecare schimbare de valoare.

PR1

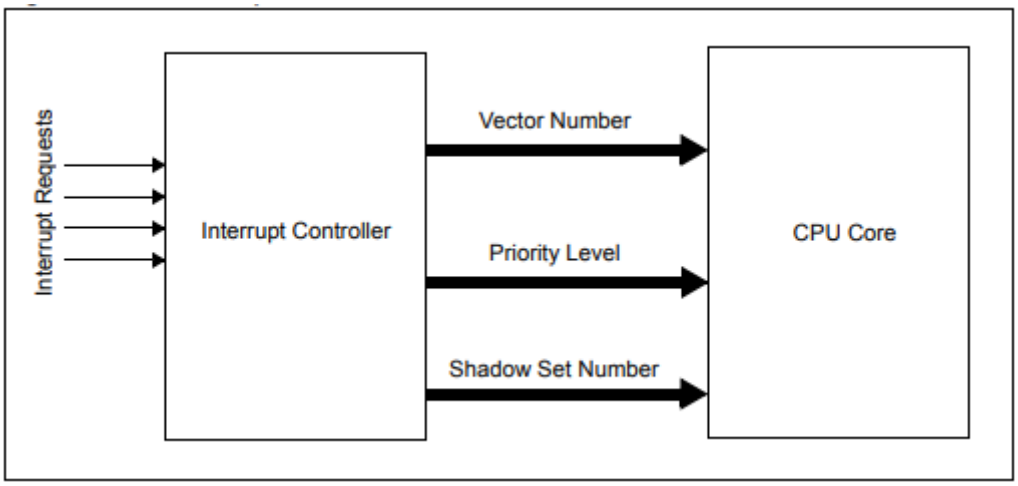
Acest registru conține valoarea maximă până la care circuitul timer va număra. În momentul în care valoarea din TMR1 este egală cu valoarea din PR1, un eveniment este generat.

8.3 Sistemul de întreruperile PIC32

Evenimentele apărute în circuitele periferice duc la apariția întreruperilor la nivelul microprocesorului. Blocul de întreruperi care strânge aceste întreruperi este situat în afara microprocesorului și se ocupă în primul rând de prioritizarea acestora înainte de a le prezenta microprocesorului.

Caracteristicile blocului de întreruperi:

- 96 surse de întrerupere
- 64 vectori de întrerupere
- moduri de lucru single și multi-vector
- 7 nivele de prioritate și 4 de sub prioritate pentru fiecare vector
- posibilitatea de a configura mărimea spațiului de memorie unde sunt stocați vectorii de întrerupere



8.3.1 Regiștrii de control al întreruperilor PIC32

Blocul de întreruperi utilizează următorii regiștrii speciali (SFRs) pentru a trata întreruperile venite de la periferice.

INTCON: Interrupt Control Register

bit 12 MVEC: Multi Vector Configuration bit

1 = blocul de întreruperi este configurat în modul multi-vector

0 = blocul de întreruperi este configurat în modul single-vector

INTSTAT: Interrupt Status Register

bit 5-0 VEC<5:0>: Interrupt Vector bits

00000-11111 = vectorul de întreruperi transmis microprocesorului

Pentru 96 de surse de întreruperi avem nevoie de 3 instanțe din fiecare dintre regiștrii următori:

IFSx: Interrupt Flag Status Register

IECx: Interrupt Enable Control Register

IPCx: Interrupt Priority Control Register

Name	Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
INTCON(1,2,3)	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	SS0
	15:8	—	—	—	MVEC	—	TPC<2:0>		
	7:0	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
INTSTAT(1,2,3)	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	—	—	—	—	—	SRIPL<2:0>		
	7:0	—	—	VEC<5:0>					
IPTMR(1,2,3)	31:24	IPTMR<31:24>							
	23:16	IPTMR<23:16>							
	15:8	IPTMR<15:8>							
	7:0	IPTMR<7:0>							
IFSx(1,2,3)	31:24	IFS31	IFS30	IFS29	IFS28	IFS27	IFS26	IFS25	IFS24
	23:16	IFS23	IFS22	IFS21	IFS20	IFS19	IFS18	IFS17	IFS16
	15:8	IFS15	IFS14	IFS13	IFS12	IFS11	IFS10	IFS09	IFS08
	7:0	IFS07	IFS06	IFS05	IFS04	IFS03	IFS02	IFS01	IFS00
IECx(1,2,3)	31:24	IEC31	IEC30	IEC29	IEC28	IEC27	IEC26	IEC25	IEC24
	23:16	IEC23	IEC22	IEC21	IEC20	IEC19	IEC18	IEC17	IEC16
	15:8	IEC15	IEC14	IEC13	IEC12	IEC11	IEC10	IEC09	IEC08
	7:0	IEC07	IEC06	IEC05	IEC04	IEC03	IEC02	IEC01	IEC00
IPCx(1,2,3)	31:24	—	—	—	IP03<2:0>			IS03<1:0>	
	23:16	—	—	—	IP02<2:0>			IS02<1:0>	
	15:8	—	—	—	IP01<2:0>			IS01<1:0>	
	7:0	—	—	—	IP00<2:0>			IS00<1:0>	

8.4 Aplicații

- Analizati programul disponibil pe platforma eLearning si completati codul lipsa:
  - adaugati adresele registrilor perifericelor TIMER și blocul de întreruperi
  - programați perifericul TIMER și blocul de întreruperi
  - rutina de display

```
.data
#vector SSD
SSD:        .byte 0x3f,...
seconds:    .word 0x00
miliseconds:  .word 0x00
max_seconds:  .word 0x0A


#Registrii TIMER1 de control, numarare si perioada
T1CON:     .word 0xBF800600
T1CONCLR: .word 0xBF800604
T1CONSET: .word 0xBF800608
#
TMR1:       ...
PR1:        ...

#Registrii TIMER1 de control al intreruperilor
#
IEC0:       ...
IFS0:       ...
IPC1:       ...
INTCON:     ...


#Registrii PORTE de directie si date
TRISE:      ...

#-----------------------------------------------------------------------------
main:
        jal PORTE_OUT_EN
        jal DISABLE_INTERRUPTS
        jal CONFIGURE_TIMER1
```

```
        jal ENABLE_INTERRUPTS

loop:
        jal DISPLAY_SECONDS
        j loop


#-----------------------------------------------------------------------------
PORTE_OUT_EN:
        ...

#-----------------------------------------------------------------------------
DISABLE_INTERRUPTS:
        di $v0
        ehb
        nop
        jr $ra

#-----------------------------------------------------------------------------
ENABLE_INTERRUPTS:
        ei $v0
        jr $ra

#-----------------------------------------------------------------------------
# This code example enables the Timer1 interrupts, loads the Timer1 period
# register, and starts the timer.
# When a Timer1 period match interrupt occurs, the interrupt service routine must clear
# the Timer1 interrupt status flag in software.*/
#-----------------------------------------------------------------------------
CONFIGURE_TIMER1:
        #Oprire timer
        lw $t0, T1CON
        sw $0, 0($t0)
        #Stergere registru de timer
        lw $t0, TMR1
        sw $0, 0($t0)
        #Se incarca perioada de numarare a timerului
        lw $t0, PR1
        li $t1, 62499 #aproximativ 1/5 secunde
        sw $t1, 0($t0)

        #Se seteaza prioritatea intreruperilor circuitului TIMER1 la 3
        lw $t0, IPC1SET
        li $t1, 0x000c
        sw $t1, 0($t0)
        #Se sterge intreruperea de TIMER1 din
        lw $t0, IFS0CLR
        li $t1, 0x0010
        sw $t1, 0($t0)
        #Se activeaza interuperile TIMER1
        lw $t0, IEC0SET
        li $t1, 0x0010
        sw $t1, 0($t0)
        #Se porneste timerul
        lw $t0, T1CONSET
        li $t1, 0x8000
        sw $t1, 0($t0)
        #Controller-ul de intrerupe se configureaza in mod multi-vector
        lw $t0, INTCONSET
        li $t1, 0x1000
        sw $t1, 0($t0)

        jr $ra

#-----------------------------------------------------------------------------
DISPLAY_SECONDS:
        # s0: Adresa de baza a vectorului de valori SSD
        la $s0, SSD
        # s3: Adresa unde este stocata valoarea secundelor
        la $s3, seconds
        # s1: current value for seconds
        lw $s1, seconds
        # s2: numarul maxim de secunde pana la care se numara
        lw $s2, max_seconds
```

```
DISPLAY_LOOP:
        ...
DISPLAY:
        ...
        jal DELAY
        j DISPLAY_LOOP

        jr $ra

#------------------------------------------------------------------------
DELAY:
        ...

#------------------------------------------------------------------------
.section .vector_4,code
        #salt catre subrutina de tratare a intreruperilor TIMER1
        j Timer1Handler
        nop
        #salt catre programul principal
        j DISPLAY_LOOP
#------------------------------------------------------------------------
.text
.ent Timer1Handler
Timer1Handler:
        #shadow reg part
        rdpgpr      $sp,$sp
        addiu       $sp,$sp,-32
        #Se salveaza in stiva registrii ce vor fi folositi in subrutina
        sw          $v1,8($sp)
        sw          $v0,4($sp)
        lw          $v1,20($sp)
        andi        $v1,$v1,0xf
        sw          $s8,12($sp)
        nop
        addu        $s8,$sp,$0

        #Se incrementeaza numarul de secunde
        ...

        #Se sterge intreruperea generata de TIMER1
        lw $v0, IFS0CLR
        li $v1, 0x0010
        sw $v1, 0($v0)


        #Se restaureaza valorile setului de registrii
        addiu       $sp,$sp,32
        wrpgpr      $sp,$sp
        eret
.end Timer1Handler
```

8.5 Bibliografie

https://reference.digilentinc.com/reference/pmod/pmodssd/reference-manual
https://ro.wikipedia.org/wiki/Display_de_%C8%99apte_segmente
https://www.digikey.com/eewiki/pages/viewpage.action?pageId=78087460

PIC32 Family Reference Manual Section 14. Timers:
http://ww1.microchip.com/downloads/en/devicedoc/61105f.pdf

PIC32 FRM - Section 8. Interrupts:
http://ww1.microchip.com/downloads/en/devicedoc/61108f.pdf