

TAJMERI I BROJAČI: **PIC16F887**

- PIC 16F887 mikrokontroler ima tri tajmera/brojača:
 - 8 - bitni tajmer/brojač (registar **TMR0**)
 - 16 - bitni tajmer/brojač (registar **TMR1H | TMR1L**)
 - 8 - bitni tajmer/brojač (registar **TMR2**)

TAJMER/BROJAČ: **TMR0**

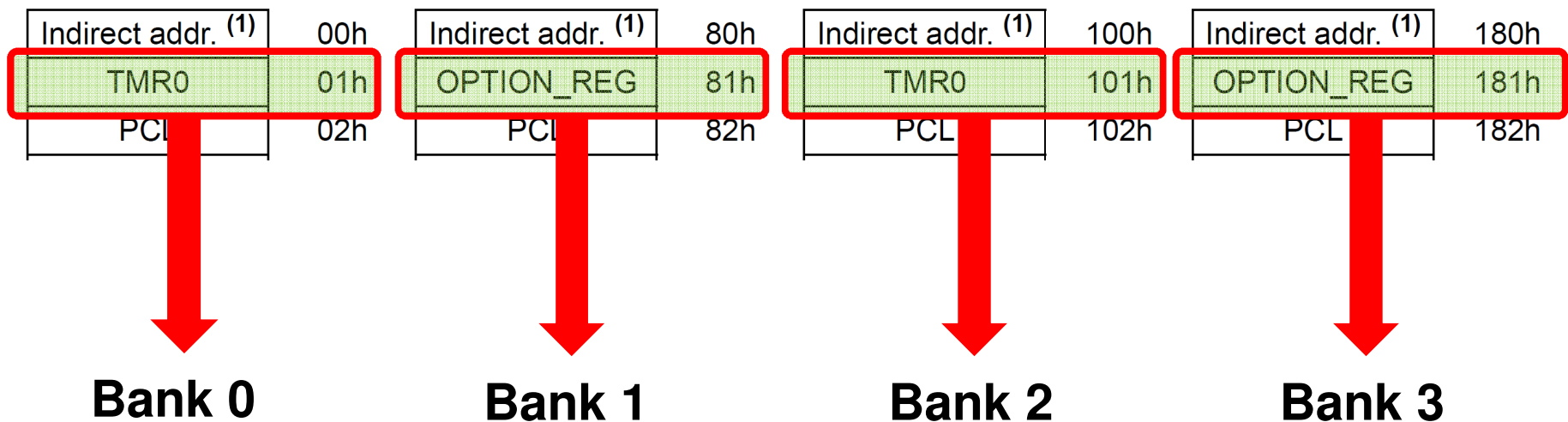
- Tajmer mod:
 - sadržaj **registra TMR0** se inkrementira na svaku ivicu takta **Fosc/4** (bez djelitelja)
- Brojački mod:
 - sadržaj **registra TMR0** se inkrementira na svaku ivicu **vanjskog impulsa** koji se pojavi na I/O pinu **RA4/T0CKI** (bez djelitelja)

RE3/MCLR/VPP	☐	1	40	☐	RB7/ICSPDAT
RA0/AN0/ULPWU/C12IN0-	☐	2	39	☐	RB6/ICSPCLK
RA1/AN1/C12IN1-	☐	3	38	☐	RB5/AN13/T1G
RA2/AN2/VREF-/CVREF/C2IN+	☐	4	37	☐	RB4/AN11
RA3/AN3/VREF+/C1IN+	☐	5	36	☐	RB3/AN9/PGM/C12IN2-
RA4/T0CKI/C1OUT	☐	6	35	☐	RB2/AN8
RA5/AN4/SS/C2OUT	☐	7	34	☐	RB1/AN10/C12IN3-

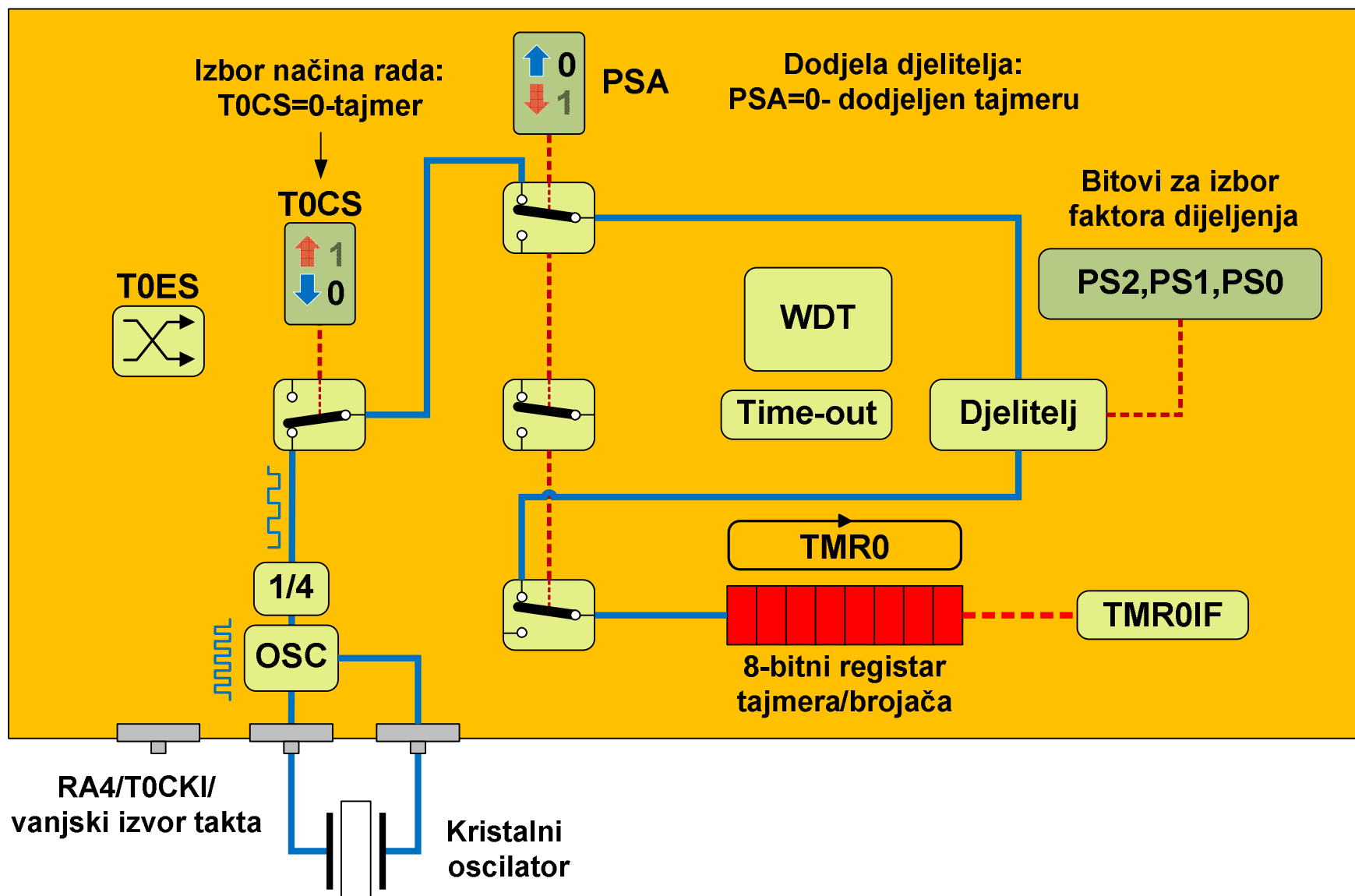
- Sadržaj **registra TMR0** dostupan je korisniku i bilo kom trenutku u programu

TAJMER/BROJAČ: TMR0- REGISTRI U SFR MEMORIJI

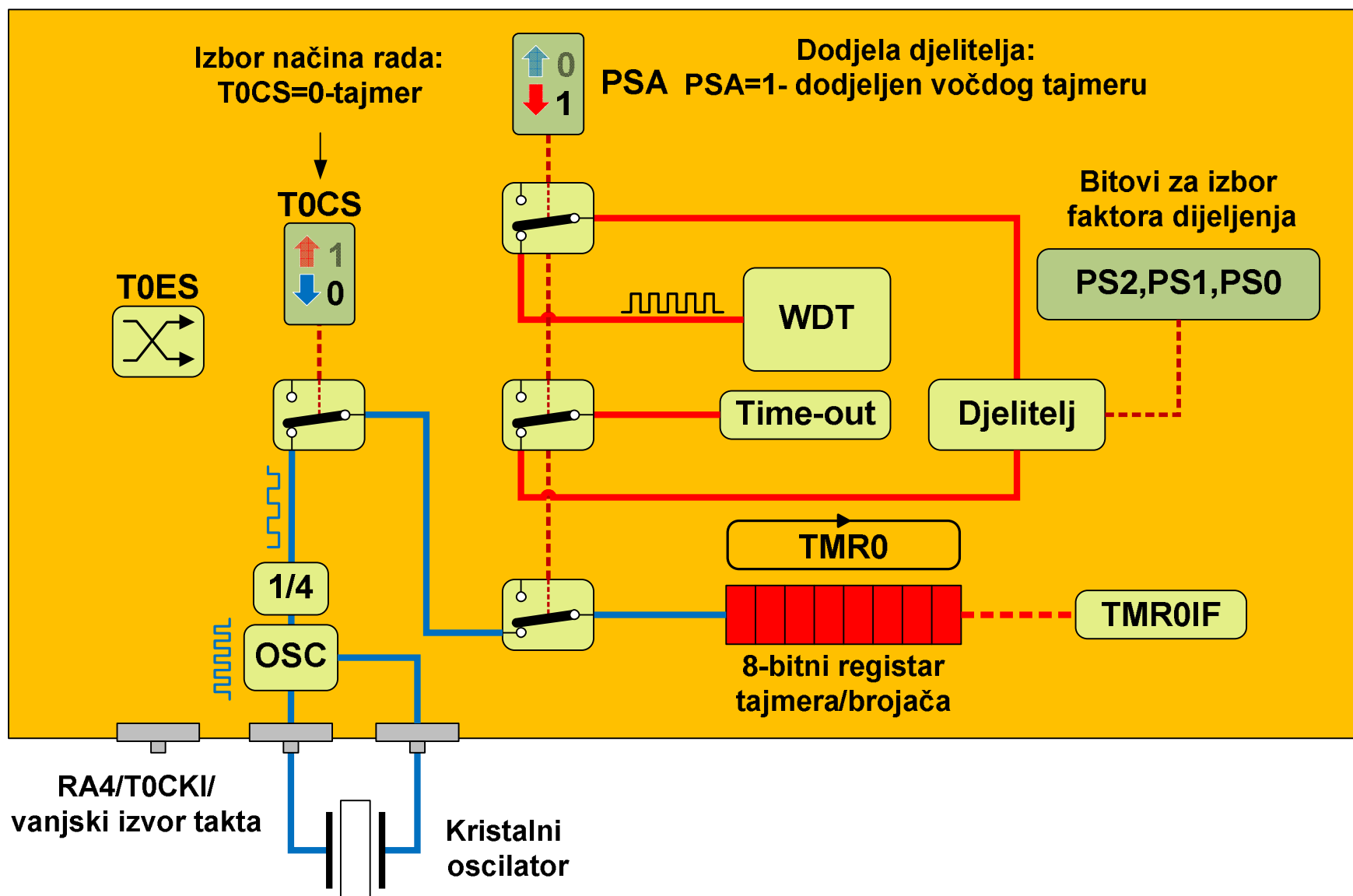
- **OPTION_REG** konfiguracioni registar dostupan je u **Bank 1** i **Bank 3** SFR memorije
- Trenutni sadržaj tajmera/brojača - **TMR0 registar** dostupan je u **Bank 0** i **Bank 2** SFR memorije



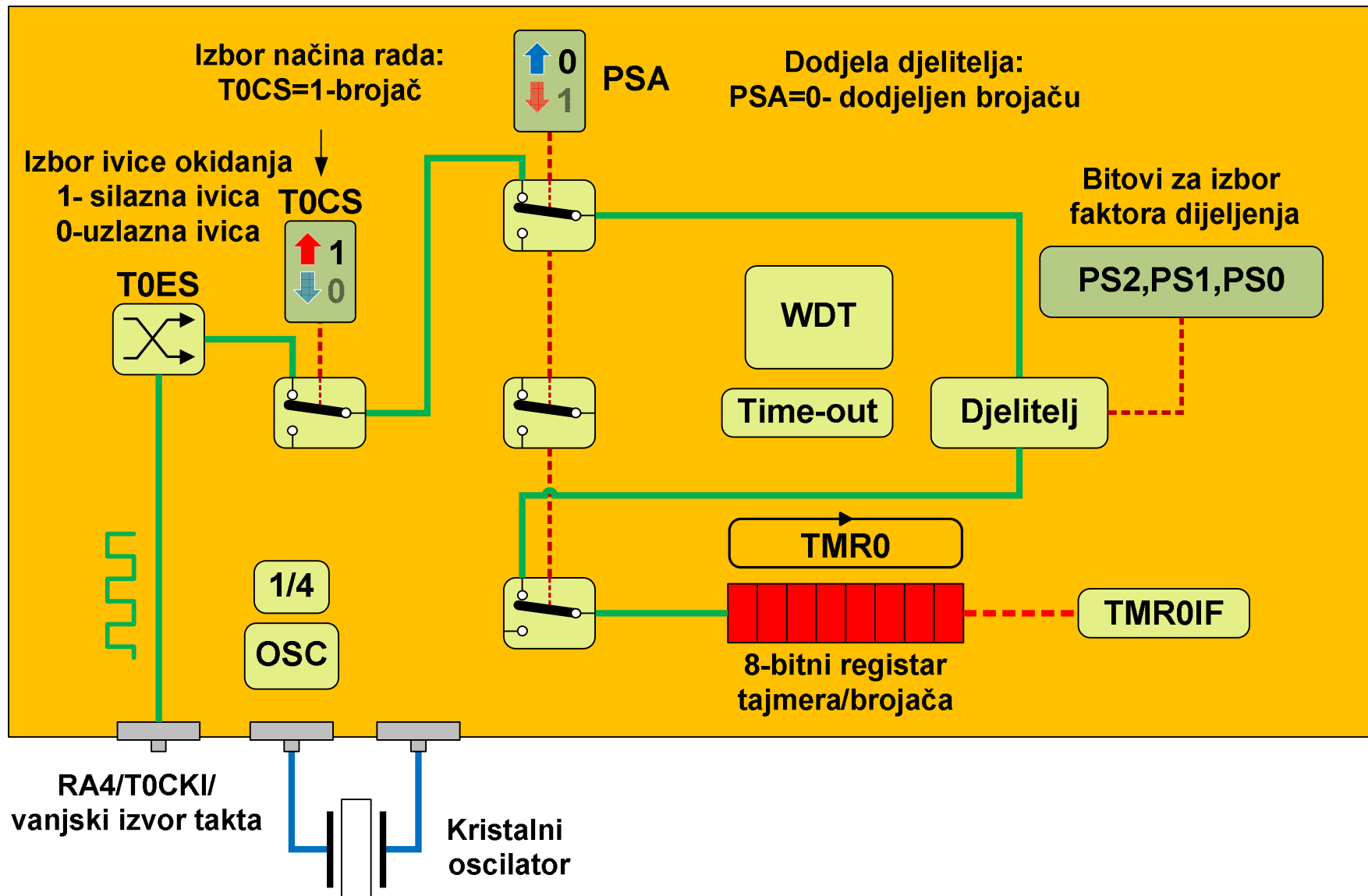
TAJMER/BROJAČ: **TMR0- TAJMERSKI MOD SA** DODJELJENIM DJELITELJEM



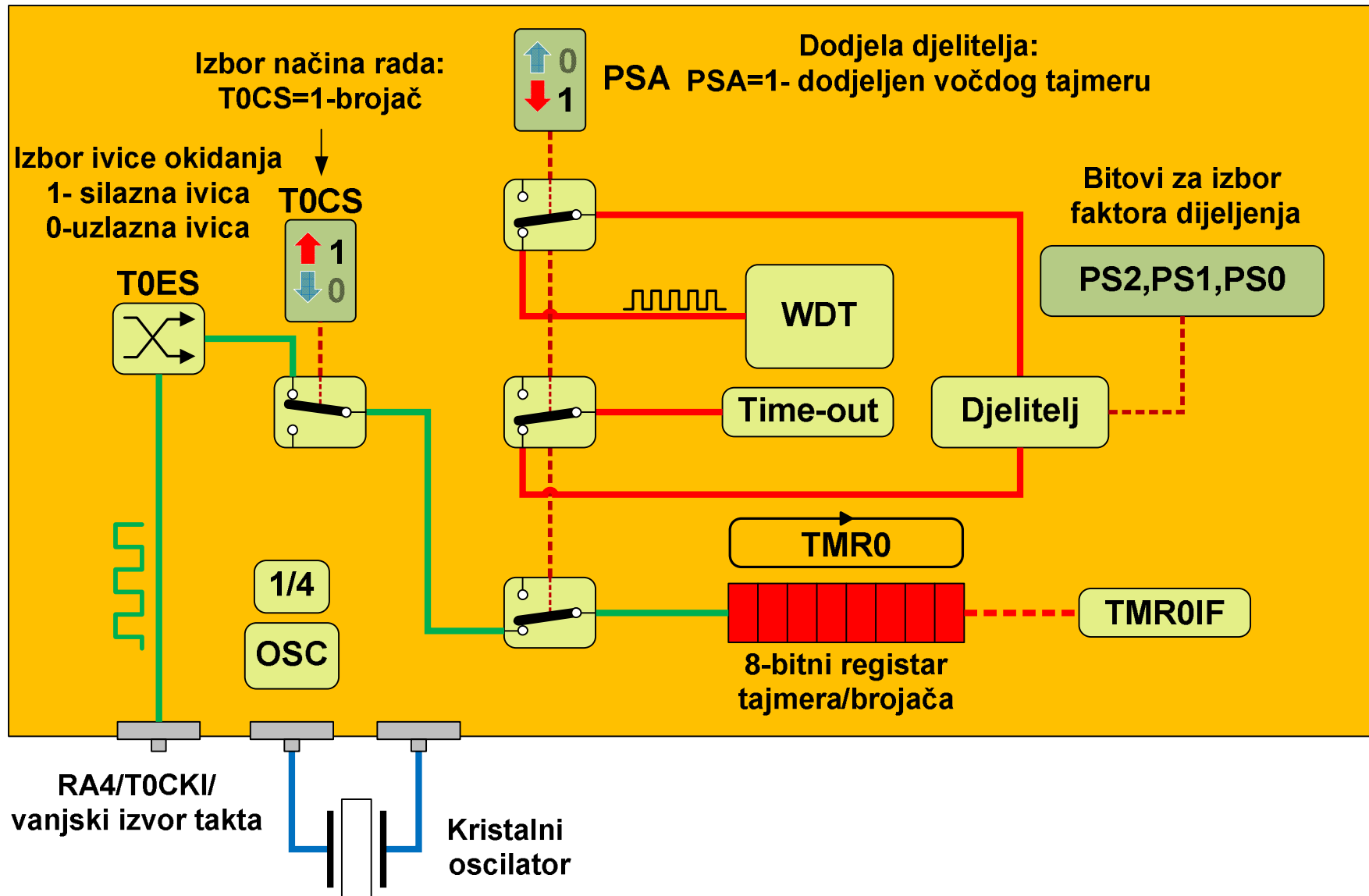
TAJMER/BROJAČ: **TMR0- TAJMERSKI MOD BEZ DJELITELJA**



TAJMER/BROJAČ: TMR0- BROJAČKI MOD SA DODJELJENIM DJELITELJEM



TAJMER/BROJAČ: TMR0- BROJAČKI MOD BEZ DJELITELJA



TAJMER/BROJAČ: **TMR0- OPTION_REG** REGISTAR

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)
OPTION_REG	RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **T0CS – Clock Select bit**

Izbor načina rada TMR0:

1 – **TMR0** radi kao **brojač** impulsa dovedenih na I/O pin RA4

0 – **TMR0** radi kao **tajmer** i odbrojava taktne impulse $F_{osc}/4$

- **T0SE – TMR0 Source Edge Select bit**

TMR0 radi kao brojač

1 – Sadržaj registra TMR0 se uvećava nailaskom **silazne ivice** impulsa na I/O pinu RA4

0 – Sadržaj registra TMR0 se uvećava nailaskom **uzlazne ivice** impulsa na I/O pinu RA4

TAJMER/BROJAČ: **TMR0- OPTION_REG** REGISTRAR

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)
OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

- **PSA – Prescaler Assignment bit**

Dodjela djelitelja:

- 1 – djelitelj dodjeljen **vočdog tajmeru** (WDT)
- 0 – djelitelj dodjeljen **tajmeru/brojaču** TMR0

- **PS2, PS1, PS0 – Prescaler Rate Select bits**

Kombinacija ovih bitova određuje faktor dijeljenja djelitelja

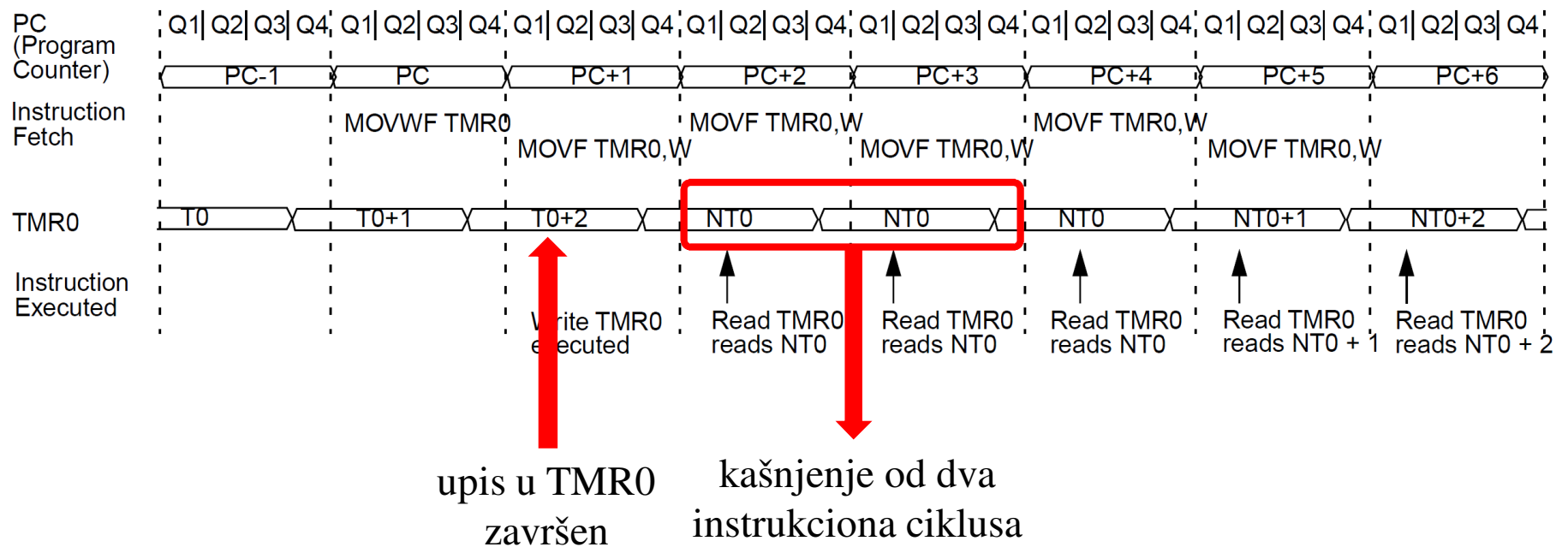
TAJMER/BROJAČ: **TMR0- OPTION_REG** REGISTAR

Faktor dijeljenja u zavisnosti od kombinacije bitova **PS2, PS1, PS0**

PS2	PS1	PS0	TMR0	WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

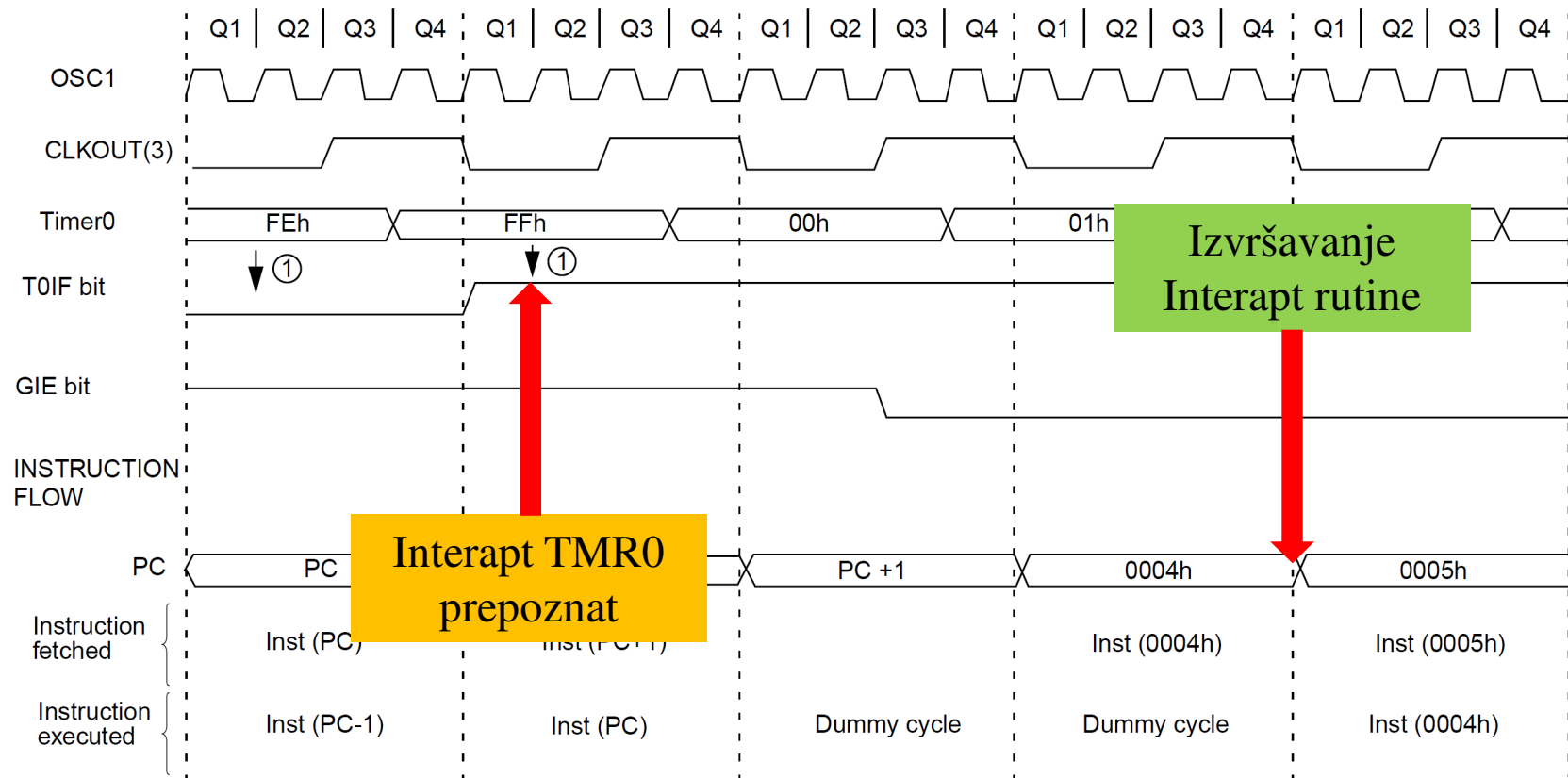
TAJMER/BROJAČ: TMR0- VREMENSKI DIJAGRAM

- Tajmer TMR0 započinje rad upisom 8-bitnog podatka u registar TMR0
- Upis u registar TMR0 uzrokuje kašnjenje od dva instrukciona ciklusa ($2 \cdot T_{CY}$)
- Do ikrementiranja tajmera dolazi tek nakon trećeg instrukcionog ciklusa



TAJMER/BROJAČ: TMR0- INTERAPT VREMENSKI DIJAGRAM

- Interapt tajmera/brojača dešava se pri prekoračenju, tj. promjeni stanja 8-bitnog registra TMR0 sa FFh na 00h (sa 255 na 0)
- Pri tome dolazi do setovanja interapt flega (bita) T0IF u INTCON registru



TAJMER/BROJAČ: **TMR0- DODJELA DJELITELJA TMR0→WDT**

*Dodjela djelitelja TMR0→WDT

BANKSEL TMR0; izabrati banku koja sadrži TMR0

CLRWDWT; obrisati sadržaj vočdog tajmer (WDT=0)

CLRFTMR0; obrisati sadržaj TMR0 i djelitelja

BANKSEL OPTION_REG; izabrati banku koja sadrži
OPTION_REG

BSF OPTION_REG, PSA ; PSA=1 djelitelj dodjeljen WDT

CLRWDWT ; obrisati sadržaj vočdog tajmer (WDT=0)

MOVLW b'11111**000**'; maskirati bitove za faktor dijeljenja

ANDWF OPTION_REG, W ; I operaciom obrisati maskirane
bitove

IORLW b'00000**101**'; OR operaciom postaviti faktor dijeljenja

MOVWF OPTION_REG; na 1:32 (101=1:32)

TAJMER/BROJAČ: **TMR0- DODJELA DJELITELJA WDT→TMR0**

*Dodjela djelitelja WDT→TMR0

CLRWDW; obrisati sadržaj vočdog tajmera (WDT=0) i djelitelja

BANKSEL OPTION_REG; izabrati banku koja sadrži

OPTION_REG

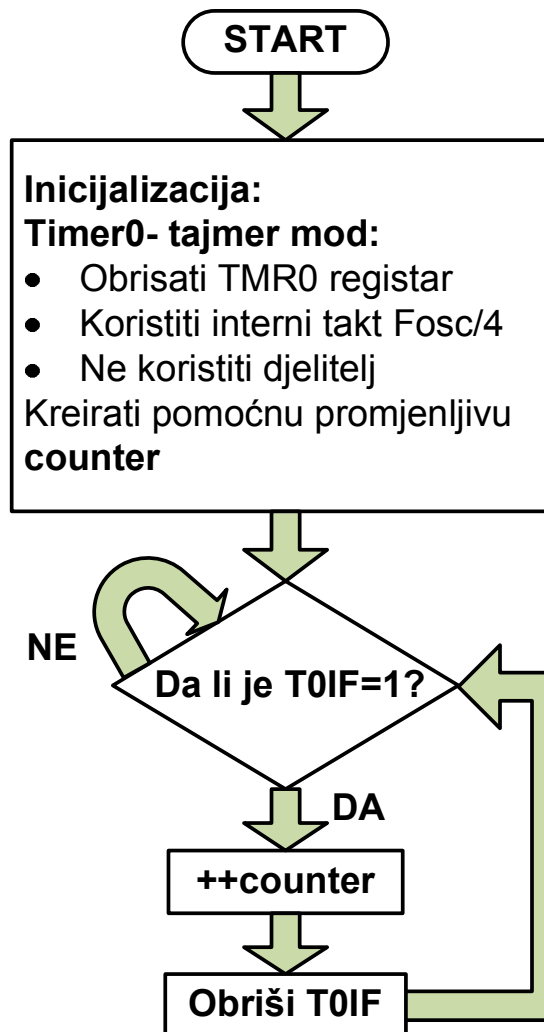
MOVLW b'1111**0000**'; maskirati PSA + bitove za faktor
dijeljenja

ANDWF OPTION_REG, W; I operaciom obrisati maskirane
bitove

IORLW b'0000**0011**'; OR operaciom postaviti PSA=0 i faktor MOVWF
OPTION_REG; dijeljenja na 1:16 (011=1:16)

TAJMER/BROJAČ: **TMR0- POLLING ALGORITAM**

- Kod polling algoritma periodično se provjerava status interapt flega T0IF. Svaki put kada je T0IF=1 inkrementira se pomoćna promjenljiva counter



TAJMER/BROJAČ: **TMR0- POLLING ALGORITAM**

*TMR0 – polling algorithm

unsigned char counter; //promjenljiva za čuvanje broja
prekoračenja tajmera TMR0

Init(void)

{

TMR0 = 0; //Obrisati TMR0 registar

OPTION = 0B00001000; //TMR0 konfigurisan u
tajmerski mod, djelitelj
dodjeljen vočdog tajmeru

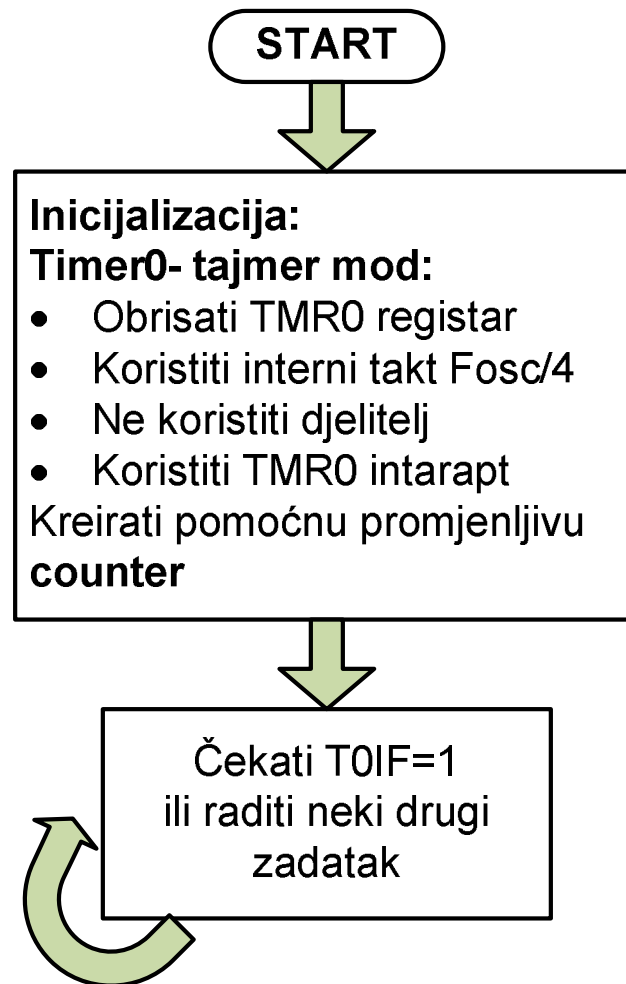
}

TAJMER/BROJAČ: **TMR0- POLLING ALGORITAM**

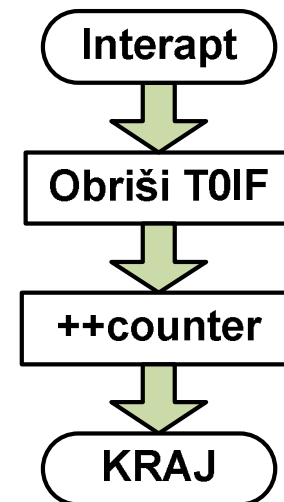
```
main(void)
{
    Init(); //Poziv procedure za inicijalizaciju
    while(1) //beskonačna petlja za provjeru da li je T0IF=1?
    {
        if (T0IF)
        {
            ++counter; //ako je T0IF = 1 uvećati brojač za 1
            T0IF = 0; //Obrisati T0IF fleg da bi se
                       //detektovao sljedeće prekoračenje
        }
    }
}
```

TAJMER/BROJAČ: **TMR0- INTERAPT ALGORITAM**

- Svaki put kad je interapt fleg T0IF setovan (T0IF=1) prekida se izvršenje tekućeg programa i odlazi se na izvršenje interapt rutine (ISR – interrupt service routine)



T0IF=1



TAJMER/BROJAČ: **TMR0- INTERRUPT ALGORITAM**

*TMR0 – interrupt algorithm

unsigned char counter; //promjenljiva za čuvanje broja
prekoračenja tajmera TMR0

void interrupt Timer0_ISR(void) //ISR rutina

{

if (T0IE && T0IF) //ako je TMR0 interapt omogućen i ako
je T0IF=1 izvršiti

{

T0IF=0; //obrisati TMR0 interapt fleg

++counter; //uvećati sadržaj brojača za 1

}

}

TAJMER/BROJAČ: **TMR0- INTERRUPT ALGORITAM**

*TMR0 – interrupt algorithm

Init(void)

{

TMR0 = 0; //Obrisati TMR0 registar

OPTION = 0B00001000; //TMR0 konfigurisan u
tajmerski mod, djelitelj
dodjeljen vočdog tajmeru

TOIE = 1; //dozvoliti interapt od strane TMR0

GIE = 1; //dozvoliti Global interapte

}

TAJMER/BROJAČ: **TMR0- INTERRUPT ALGORITAM**

*TMR0 – interrupt algorithm

```
main(void)
{
    Init(); //Poziv procedure za inicijalizaciju

    while(1) //beskonačna petlja
    {
    }
}
```

TAJMER/BROJAČ: TMR0- PRIMJER

- Napraviti zadržku od $T_{\text{delay}} = 0.5$ s pomoću tajmera TMR0 ako je $F_{\text{osc}} = 4$ MHz
- Vrijeme zadržke 1 ciklusa tajmera TMR0 dato je izrazom:

$$T_{\text{ciklusa}} = \text{Prescaler} * \left[256 - (TMR0 - 2) \right] * 4 * T_{\text{osc}}$$

- Ukupno vrijeme zadržke T_{delay} dobije se zbrajanjem dojedinačnih ciklusa zadržke T_{ciklusa} ako je $T_{\text{delay}} > T_{\text{ciklusa}}$

$$T_{\text{delay}} = \text{Count} * T_{\text{ciklusa}}$$

- Odredimo prvo najduže vrijeme 1 ciklusa tajmera TMR0 koje se dobije se za:
Prescaler=256 i TMR0=0

TAJMER/BROJAČ: **TMRO- PRIMJER**

- Uvrštavanjem brojnih vrijednosti dobija se:

$$T_{ciklusa\ max} = 256 * [256 - (0 - 2)] * 4 * 500\ ns$$

$$T_{ciklusa\ max} = 256 * 258 * 1\ \mu s$$

$$T_{ciklusa\ max} = 66048\ \mu s$$

- Sada se može odrediti promjenljiva Count kao:

$$Count = T_{delay} / T_{ciklusa\ max}$$

$$Count = 0.5\ s / 66048\ \mu s$$

$$Count = 7.57$$

- Ona bi trebala biti cijeli broj pa imamo grešku u T_{delay} :

$$T_{delay}^1 = 7 * 66048\ \mu s = 0.46233\ s$$

$$T_{delay}^2 = 8 * 66048\ \mu s = 0.52838\ s$$

TAJMER/BROJAČ: TMR0- PRIMJER

- Očigledno vrijednost $T_{ciklusa}$ treba korigovati upisom u TMR0 odgovarajuće vrijednosti
- Zaokružimo Count na sljedeću veću cjelobrojnu vrijednost: Count=8
- Odredimo sada vrijeme 1 ciklusa tajmera TMR0

$$T_{ciklusa} = T_{delay} / Count$$

- Sada možemo odrediti vrijednost koju treba upisati u registar TMR0:
 $T_{ciklusa} = 0.5 \text{ s} / 8 = 0.0625 \text{ s}$

$$TMR0 = 2 + 256 - \frac{T_{ciklusa}}{4 * T_{osc} * Prescaler}$$

TAJMER/BROJAČ: **TMR0- PRIMJER**

- Uvrštavanjem brojnih vrijednosti dobija se:

$$TMR0 = 2 + 256 - \frac{0.0625}{4 * 500 ns * 256}$$

$$TMR0 = 2 + 256 - \frac{62500}{256} = 13.85$$

- Zaokružimo vrijednost koju treba upisati u registar TMR0 na veću cjelobrojnu vrijednost: **TMR0=14**
- Provjerimo sada dobijeno vrijeme zadržske T_{delay}

$$T_{delay} = Count * Prescaler * [256 - (TMR0 - 2)] * 4 * T_{osc}$$

$$T_{delay} = 8 * 256 * [256 - (14 - 2)] * 4 * 500 ns$$

$$T_{delay} = 2048 * 244 * 1 \mu s = 0.4997 s$$

TAJMER/BROJAČ: **TMR0- PRIMJER**

- Napisati kod!!!!