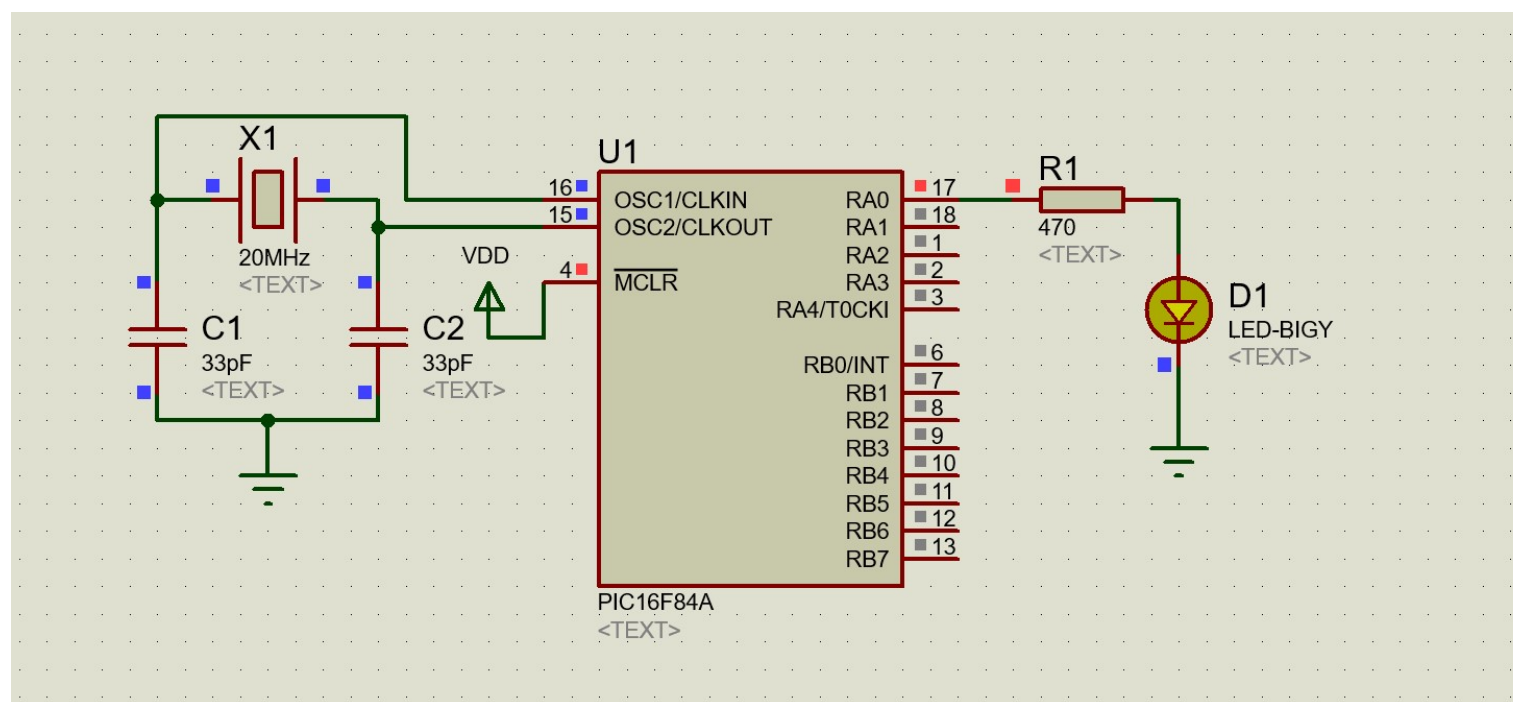




# MIKRORAČUNARSKI SISTEMI

## II deo računskih vežbi

### PIC16 mikrokontroleri



# Uvod

- PIC (***P**eripheral **I**nterface **C**ontroller*) je familija mikrokontrolera razvijena od strane kompanije Microchip Technology
- Naziv su dobili po modelu PIC1650, razvijenog sredinom sedamdesetih godina proslog veka
- Kasnije, promenjeno značenje u ***P**rogrammable **I**ntelligent **C**omputer*
- Karakteristike:
  - niska cena
  - dobra podrška
  - velika zajednica korisnika

# Istorija

- Nastao iz potrebe da se odvoje I/O operacije od CPU (1975. godine)
- Originalno je bio 8-bit
- Kreiran da radi zajedno sa 16-bit CPU – CP1600
- Koristio ROM memoriju za čuvanje koda
- Jedan od prvih RISC-baziranih uređaja
- 1993. se javlja PIC16C84, prvi sa EEPROM
- 2001. se javljaja PIC sa flash memorijom
- Do 2013. godine je u upotrebi preko 12 milijardi primeraka mikrokontrolera iz ove familije

# Familije PIC

- Različite širine instrukcija i memorije podataka
- Memorija podataka
  - 8, 16, 32-bit
- Instrukcije
  - 12, 14, 16 i 24-bit
- Različite dubine poziva steka
- Različit broj U/I portova
- Različit broj registara i memorijskih reči
- Naprednije verzije poseduju dodatne module
  - Serijska komunikacija: USART, I2C, SPI, USB
  - ADC/DAC
  - Ethernet

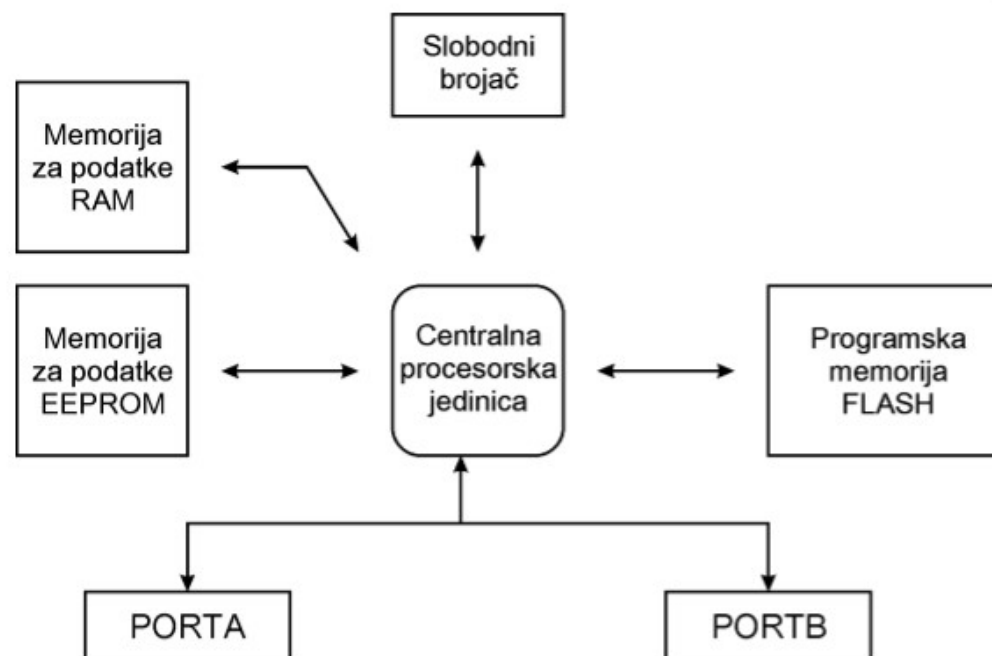
# Pregled PIC familija

- PIC10 i PIC12
  - 12bit instrukcije, 2 nivoa steka, nema HW mul/div
- PIC16
  - 14bit instrukcije, 8 nivoa steka
- PIC17
  - 16bit instrukcije, 16 nivoa steka, HW množenje (8b x 8b)
- PIC18
  - 16bit instrukcije, 31 nivo steka, rad sa USB
- PIC24
  - 24bit
  - Hardversko množenje u 1 ciklusu (16b x 16b)
  - deljenje u 19 ciklusa (32b / 16b)
  - DSP operacije
  - DMA pristup periferijama
- PIC32
  - MIPS-baziran, USB OTG, Ethernet

# Razvojna okruženja za PIC

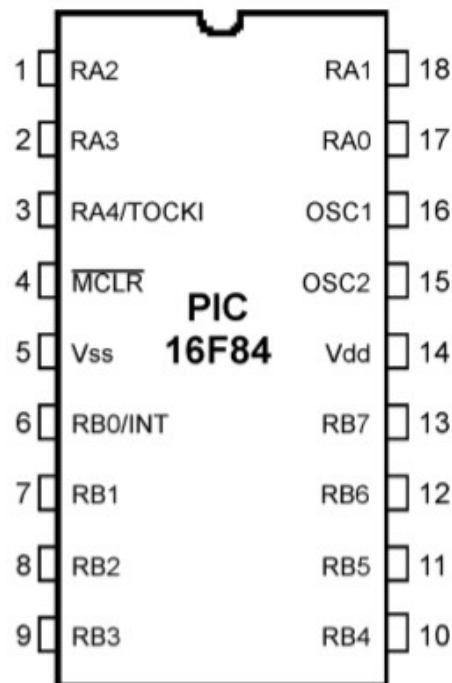
- Freeware MPLAB X razvojno okruženje
  - Razvijeno od strane proizvođača
  - Integriše assembler, linker, softverski simulator i debugger
- Različite mogućnosti pisanja koda
  - assembler
  - C (XC8, MikroC)
  - C++
  - Basic
  - Pascal
  - PICBASIC
- Proteus 8.4 SP0
  - Podrška za simulaciju
  - XC8 i ASM

# Blok šema PIC16F84



Sl. 2.1 Blok šema mikrokontrolera PIC16F84

# Pinovi PIC16F84

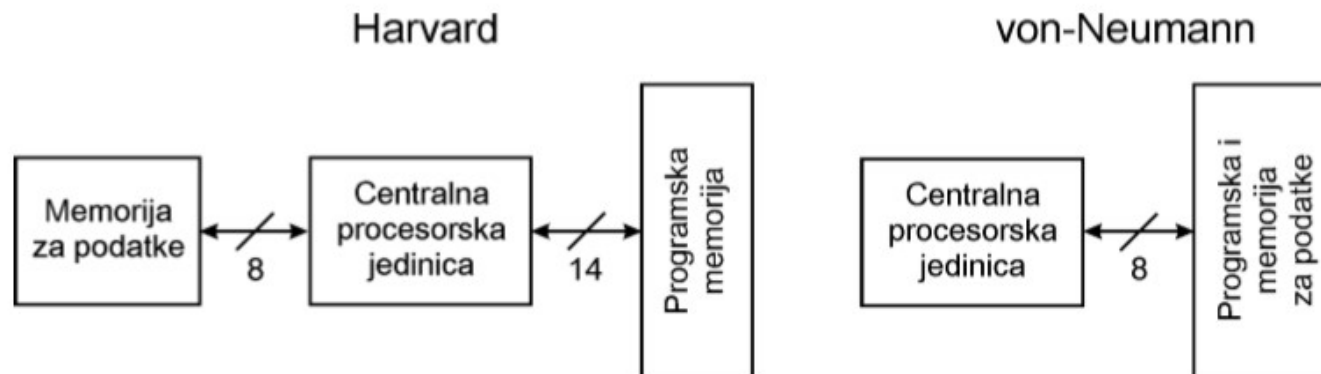


Pin br. 1	<b>RA2</b>	Drugi pin porta A. Nema nikakvu dodatnu ulogu.
Pin br. 2	<b>RA3</b>	Treći pin porta A. Nema nikakvu dodatnu ulogu.
Pin br. 3	<b>RA4</b>	Četvrti pin porta A. Na njemu se takođe nalazi i TOCK1 koji ima tajmersku f-j.
Pin br. 4	<b>MCLR</b>	Reset ulaz i Vpp napon programiranja mikrokontrolera
Pin br. 5	<b>Vss</b>	Napajanje, masa.
Pin br. 6	<b>RB0</b>	Nulti pin porta B. Dodatna funkcija je interaptni ulaz.
Pin br. 7	<b>RB1</b>	Prvi pin porta B. Nema nikakvu dodatnu ulogu.
Pin br. 8	<b>RB2</b>	Drugi pin porta B. Nema nikakvu dodatnu ulogu.
Pin br. 9	<b>RB3</b>	Treći pin porta B. Nema nikakvu dodatnu ulogu.
Pin br. 10	<b>RB4</b>	Četvrti pin porta B. Nema nikakvu dodatnu ulogu.
Pin br. 11	<b>RB5</b>	Peti pin porta B. Nema nikakvu dodatnu ulogu.
Pin br. 12	<b>RB6</b>	Šesti pin porta B. 'Clock' linija u programskom modu.
Pin br. 13	<b>RB7</b>	Sedmi pin porta B. 'Data' linija u programskom modu.
Pin br. 14	<b>Vdd</b>	Pozitivan pol napajanja.
Pin br. 15	<b>OSC2</b>	Pin namenjen spajanju sa oscilatorom.
Pin br. 16	<b>OSC1</b>	Pin namenjen spajanju sa oscilatorom.
Pin br. 17	<b>RA0</b>	Drugi pin porta A. Nema nikakvu dodatnu ulogu.
Pin br. 18	<b>RA1</b>	Prvi pin porta A. Nema nikakvu dodatnu ulogu.



# Harvard RISC arhitektura

- Harvard arhitektura omogućava istovremeno čitanje instrukcije i čitanje/upis podataka

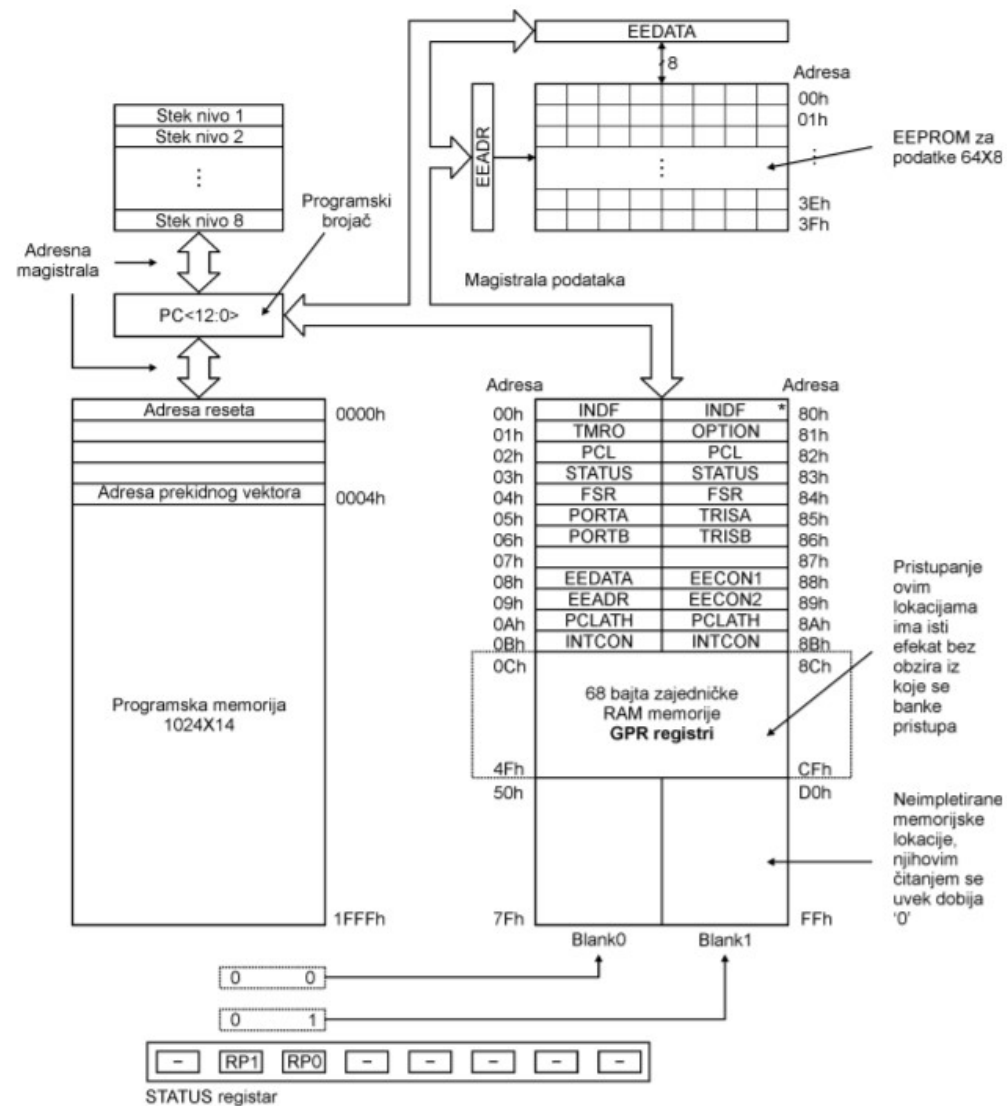


# Primene PIC16F84

- Elektronske brave
- Sigurnosni uređaji
- Automobilaska industrija
- Kućni aparati
- Svetleće reklame

# Organizacija memorije

- Programski blok
  - FLASH :1Kx14b
- Blok za podatke
  - EEPROM: 64x8b
  - RAM
    - SFR (registri specijalne namene):
      - Dve memorijske banke
      - prvih 12 lokacija u banci 0 i 1
    - GPR (registri opšte namene): 68x8b lokacija, pristup nezavisno od banke



Sl. 2.15. Memorijska organizacija mikrokontrolera 16F84

# Statusni registar

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

bit 7	Ne koristi se
bit 6-5	bitovi za selekciju banke kod direktnog adresiranja RP0 = 0 BANKA0 (00h - 7Fh) RP0 = 1 BANKA1 (80h - FFh) bit 6 (RP1) se ne koristi kod 16F8X
bit 4	TO: (time out) 1=nakon dovođenja napajanja, CLRWDWT ili SLEEP naredbe 0=dogodio se time out reset od WDT (watch dog timer)
bit 3	PD: (power down) 1=nakon dovođenja napajanja ili CLRWDWT naredbe 0=nakon izvršenja SLEEP naredbe
bit 2	Z: (zero) indikator (flag) za '0' 1=rezultat aritmetičke ili logičke operacije je 0 0= rezultat aritmetičke ili logičke operacije nije 0
bit 1	DC: Digital carry (decimalni prenos ili pozajmica kod BCD operacija) 1=biloje prenosa iz niže u višu tetradu 0= nije biloje prenosa iz niže u višu tetradu
bit 0	C: (carry/borrow) prenos ili pozajmica iz bit naviše pozicije 1=dogodio se prenos iz bita najviše pozicije 0=nije bilo prenosa iz bita najviše pozicije <b>Napomena:</b> kod oduzimanja je vrednost pozajmice komplemnirana, tj bit je 1 kad nije bilo pozajmice

# Vrste prekida kod PIC16F84

PORTB

- Spoljašnji prekid na RB0/INT pinu mikrokontrolera
- Prekid prilikom promene na pinovima 4, 5, 6 i 7 porta B
- Prekid prilikom prekoračenja TMR0 brojača
- Prekid prilikom završetka upisivanja u EEPROM

# INTCON registar

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

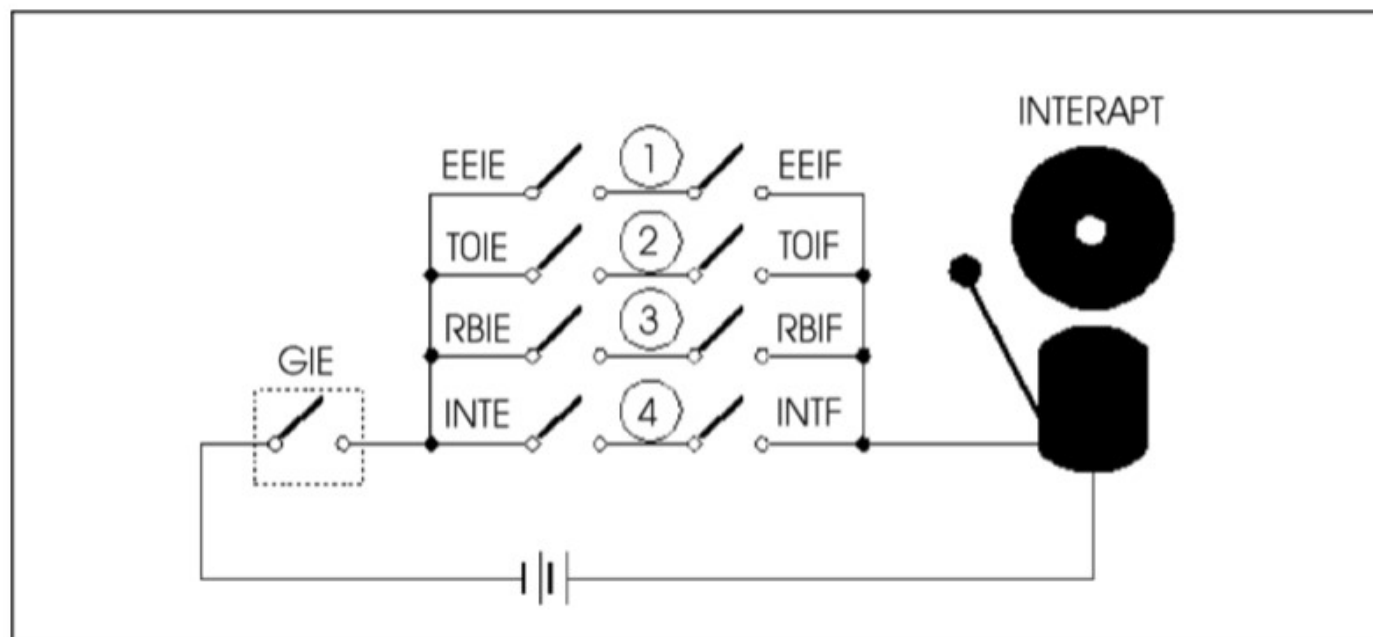
## LEGENDA:

R = bit koji se može čitati  
U = neiskorišćen bit, čita se kao "0"

W = bit koji se može upisati  
n = vrednost posle reseta

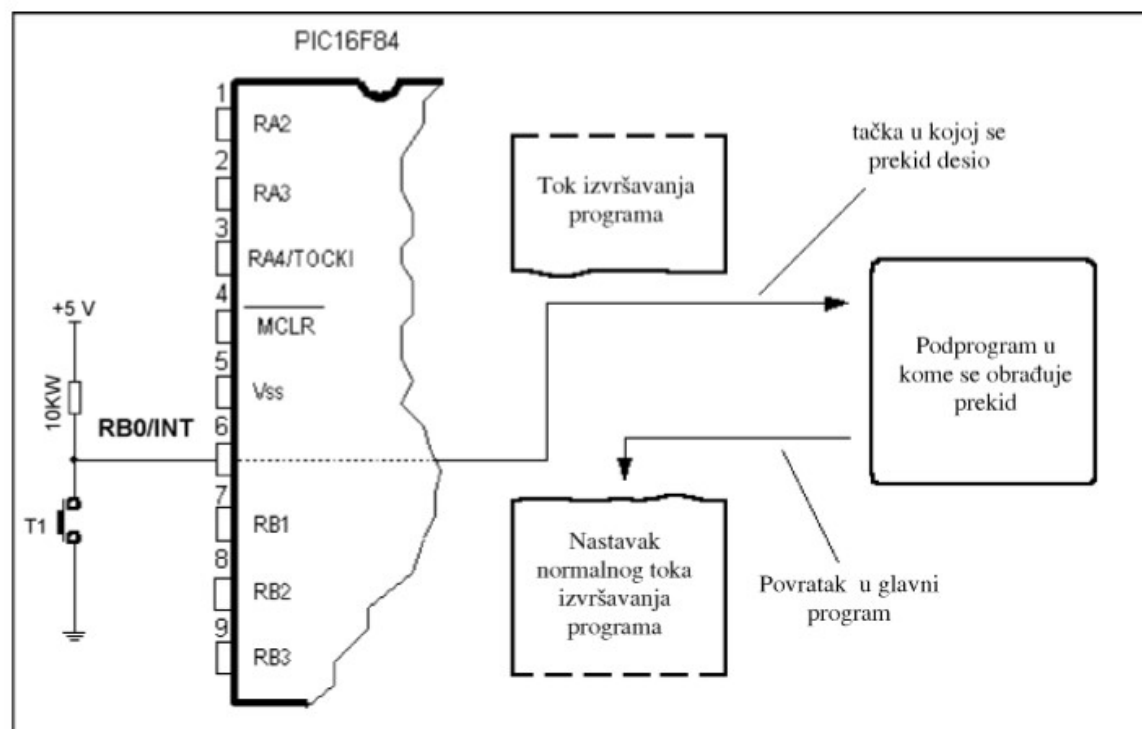
bit 7	<b>GIE:</b> bit globalne dozvole prekida 1=dozvoljeni su svi prekidi 0=zabranjeni su sviprekidi
bit 6	<b>EEIE</b> dozvola prekida na završetku upisa u data EEPROM 1=dozvoljen prekid na završetku upisa u data EEPROM 0=zabranjen prekid na završetku upisa u data EEPROM
bit 5	<b>TOIE:</b> dozvola prekida od tajmera0 (TMR0) 1=dozvoljen prekid od TMR0 0=zabranjen prekid od TMR0
bit 4	<b>INTE:</b> dozola prekida na RB0/INT priključku 1=dozoljen prekid na RB0/INT priključku 0=zabranjen prekid na RB0/INT priključku
bit 3	<b>RBIE:</b> dozvola prekida na promenu na priključcima RB4-RB7 1=dozvoljen prekid na promenu na priključcima RB4-RB7 0=zabranjen prekid na promenu na priključcima RB4-RB7
bit 2	<b>TOIF:</b> indikator prekida od TMR0 1=dogodio se prelaz TMR0 a 255 na 0 0=nije se dogodio prelaz TMR0 a 255 na 0
bit 1	<b>INTF:</b> indikator prekida na RB0/INT priključku 1=dogodio se prekid na RB0/INT priključku 0=nije se dogodio prekid na RB0/INT priključku
bit 0	<b>RBIF:</b> indikator prekida na promenu na priključcima RB4-RB7 1=dogodio se prekid na na promenu na priključcima RB4-RB7 0=nije se dogodio prekid na na promenu na priključcima RB4-RB7

# Šema prekida





# Prekidi



# Struktura prekidne rutine

- 1. Čuva se W registar bez obzira na tekuću banku
- 2. Čuva se STATUS registar u bank0.
- 3. Čuvaju se ostali registri
- [Test interrupt flegova]
- [Izvršava se prekidna rutina za obradu detektovanog prekida (ISR)]
- [Čišćenje flega obrađenog prekida]
- 4. Restaurišu se ostali registri
- 5. Restauriše se STATUS registar
- 6. Restauriše se W registar

# Option 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
RBPŪ	INTEDG	T0CS	TOSE	PSA	PS2	PS1	PS0
bit 7	<b>Not RBPŪ:</b> Otpornici vezani na Vcc uključeni/isključeni 0=otpornici na PORTB uključeni 1= otpornici na PORTB isključeni						
bit 6	<b>INTEDG:</b> ivica za spoljašnji prekid 1= prekid na rastućoj ivici signala na RB0/INT priključku 0= prekid na opadajućoj ivici signala na RB0/INT priključku						
bit 5	<b>T0CS:</b> izbor takta za TMR0 1= sa RA4/T0CKI priključka 0= interni takt kojim se izvršavaju naredbe (CLKOUT)						
bit 4	<b>TOSE:</b> izbor ivice takta za TMR0 1 = Uvećava se na prelazu sa visoko na nisko na RA4/T0CKI priključku 0 = Uvećava se na prelazu sa nisko na visoko na RA4/T0CKI priključku						
bit 3	<b>PSA:</b> dodela preskalera 1= preskaler je dodeljen WDT tajmeru 0= preskaler je dodeljen TMR0						
bit 2-0	PS2:PS0: bitovi za izbor odnosa deljenja preskalera						
	PS2:PS0		TMR0 učestanost		WDT učestanost		
	000		1 : 2		1 : 1		
	001		1 : 4		1 : 2		
	010		1 : 8		1 : 4		
	011		1 : 16		1 : 8		
	100		1 : 32		1 : 16		
	101		1 : 64		1 : 32		
	110		1 : 128		1 : 64		
	111		1 : 256		1 : 128		

bit 0

# Preskaler

- Preskaler je naziv za deo mikrokontrolera koji deli instrukcijski ciklus pre nego što on dođe do logike koja povećava stanje brojača. Ovim se dobija mogućnost merenja dužih vremenskih perioda
- Preskaler se može pridružiti jednom od dva brojača pomoću PSA bita
  - TMR0
  - Watchdog timer
- PS2,PS1,PS0 definišu vrednost preskalera

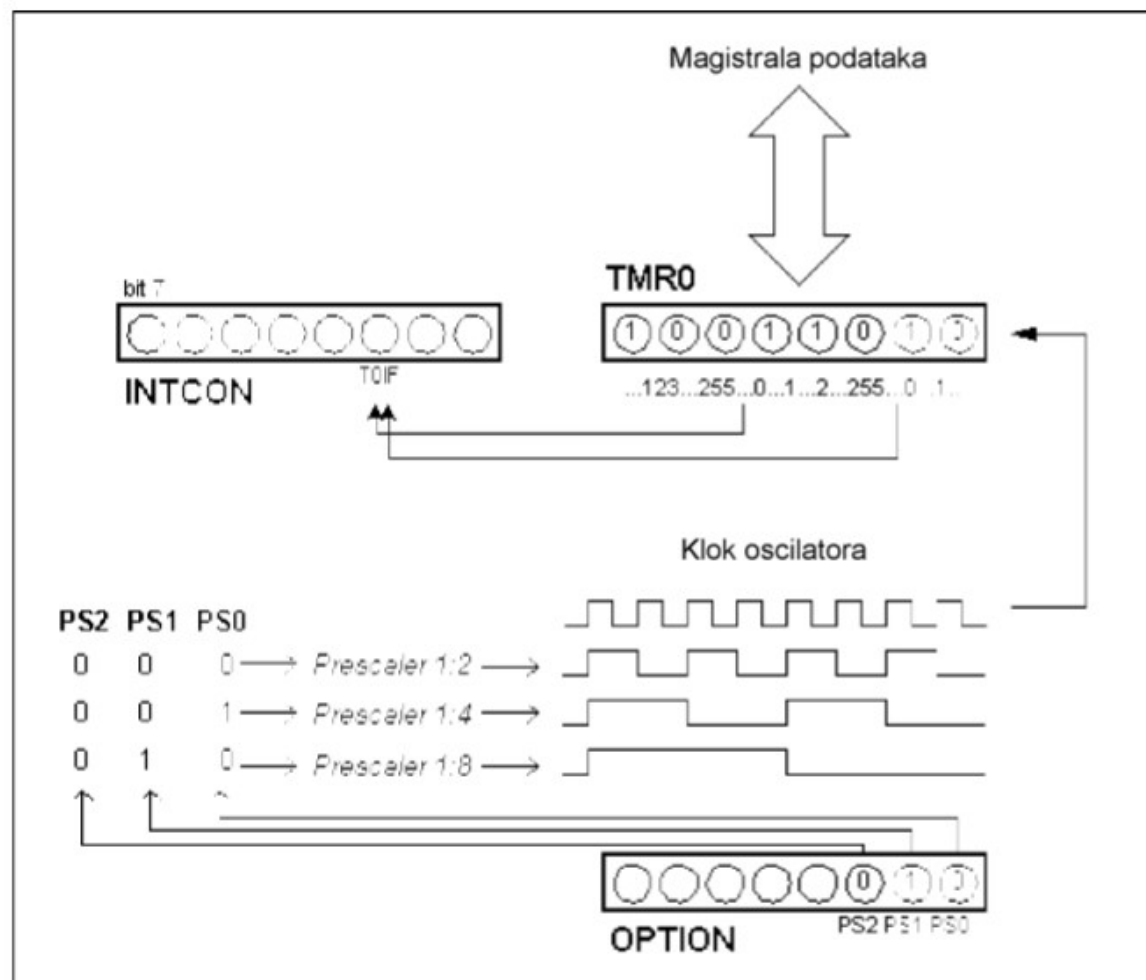
Bitovi	TMR0	WDT
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:18
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

# TMRO (Slobodni brojač )

- Pomoću njih se stvara relacije između realne veličine kao što je vreme, sa promenljivom koja predstavlja stanje brojača unutar mikrokontrolera
- PIC16F84 ima 8-bitni brojač
- Nakon svakog odbrojavnja do 255 brojač resetuje svoju vrednost na nulu i kreće sa novim ciklusom brojanja do 255. Prilikom svakog prelaska sa 255 na nulu, setuje se bit T0IF u INTCON registru. Ukoliko je dozvoljena pojava prekida, ovo se može iskoristiti za generisanje prekida i obradu prekidne rutine.
- Na programeru je da resetuje bit T0IF u prekidnoj rutini, kako bi se mogao detektovati novi prekid, tj. novo prekoračenje

# Watchdog timer (Sigurnosni brojač)

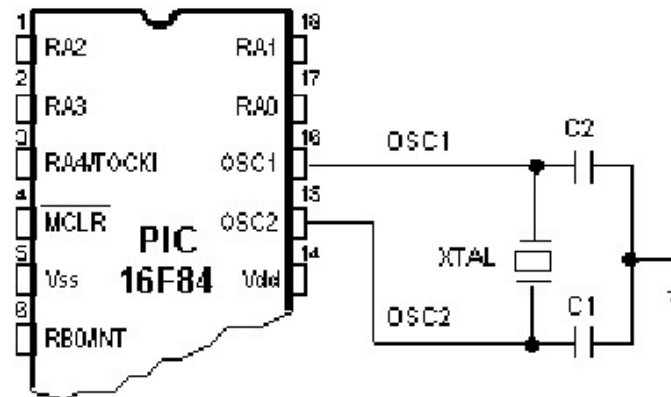
- Mehanizam mikrokontrolera kojim se mikrokontroler brani od zaglavljivanja programa. Kao i kod svakog elektronskog kola, i kod mikrokontrolera može doći do kvara ili nepravilnog rada.
- Kada se to desi mikrokontroler će stati sa radom i ostati u tom stanju sve dok ga neko ne resetuje.
- Zbog toga je uveden mehanizam watchdog koji, nakon određenog vremena, resetuje mikrokontroler (mikrokontroler sam sebe resetuje).
- Watchdog radi na jednostavnom principu: ako dođe do prekoračenja brojača mikrokontroler se resetuje i kreće sa izvršavanjem programa iz početka.
- Sledeći korak je sprečavanje resetu u slučaju pravilnog rada, što se postiže upisom nule u WDT registar (instrukcijom CLRWDT) svaki put kada se približi svom prekoračenju, čime će program sprečiti reset sve dok se pravilno izvršava.
- Jednom, kada dođe do zaglavljivanja, nula neće biti upisana, doći će do prekoračenja WDT brojača i desiće se reset koji će mikrokontroler ponovo vratiti u pravilan rad.



Sl. 2.21. Odnos brojača TMR0 i preskalera

# Oscilatori

- Procesorski takt predstavlja  $\frac{1}{4}$  špoljašnjeg takta oscilatora
- Najčešće se koriste dve vrste oscilatora
  - Kristalni oscilator (XT, HS, LP)
  - Sa otpornikom i kondenzatorom (RC)





# Primer podešavanja preskalera

- Neka je radni takt mikrokontrolera 3.2768Mhz. Potrebno je podesiti vrednost OPTION registra tako da se svaki od displej osvežava frekvencijom od po 200Hz, pri čemu imamo 2 displeja.

$$\frac{2^{15} \times 100 \text{ Hz}}{2^2 \times 2^8 \times 2^{n+1}} = 400 \text{ Hz}, \Rightarrow n = 2$$

OPTION\_REG = 1XX0 0010 = 1000 0010; (nisu uljučeni otpornici na PORTB)

```
MOVLW 0x82
```

```
MOVWF OPTION_REG ; kraj primera
```

# EEPROM memorija

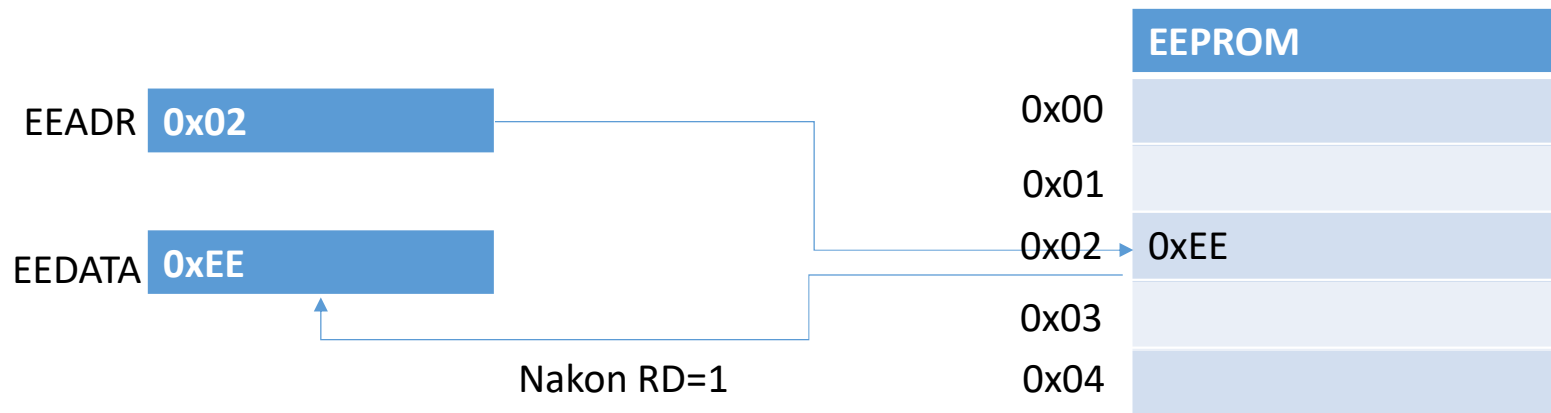
- EEPROM memorija se nalazi u posebnom memorijskom prostoru i pristupa joj se preko specijalnih registara.
- Registri za rad sa EEPROM memorijom su:
  - EEDATA
    - sadrži podatak koji je pročitao ili koga treba upisati.
  - EEADR
    - sadrži adresu EEPROM lokacije kojoj se pristupa.
  - EECON1
    - sadrži kontrolne bitove, statute i interupt flag
  - EECON2
    - ovaj registar ne postoji fizički i služi da zaštiti EEPROM od slučajnog upisa.

# EECON1 registar

- EEIF
  - Interrupt flag za detekciju prekid nakon upisa u EEPROM
- WRERR
  - Greška pri upisu
- WREN
  - Omogućiti upis
- WR
  - Izvršiti upis
- RD
  - Izvršiti čitanje

U-0	U-0	U-0	R/W-1	R/W-1	R/W-x	R/S-0	R/S-x
-	-	-	EEIF	WRERR	WREN	WR	RD
bit 7							bit 0

# Čitanje EEPROM memorije



# Upis u EEPROM memoriju



# Prekid prilikom završetka upisa u EEPROM

- Prekid može biti uključen/isključen setovanjem/resetovanjem EEIE bita u INTCON registru.
- Ovaj prekid je čisto praktične prirode. Kako upis u jednu lokaciju EEPROM-a traje oko 10ms (što je za pojmove mikrokontrolera veoma dugo), to se mikrokontroleru ne isplati da čeka da se taj upis završi, već je dodat mehanizam prekida po kome on može da nastavi sa izvršenjem glavnog programa, dok se u pozadini vrši upis u EEPROM.
- Kada se upis završi, prekid obaveštava mikrokontroler da je upis gotov. Bit EEIF, kojim se ovo obaveštenje vrši, nalazi se registru EECON1. Pojava prekida može biti onemogućena resetovanjem EEIE bita u INTCON registru.

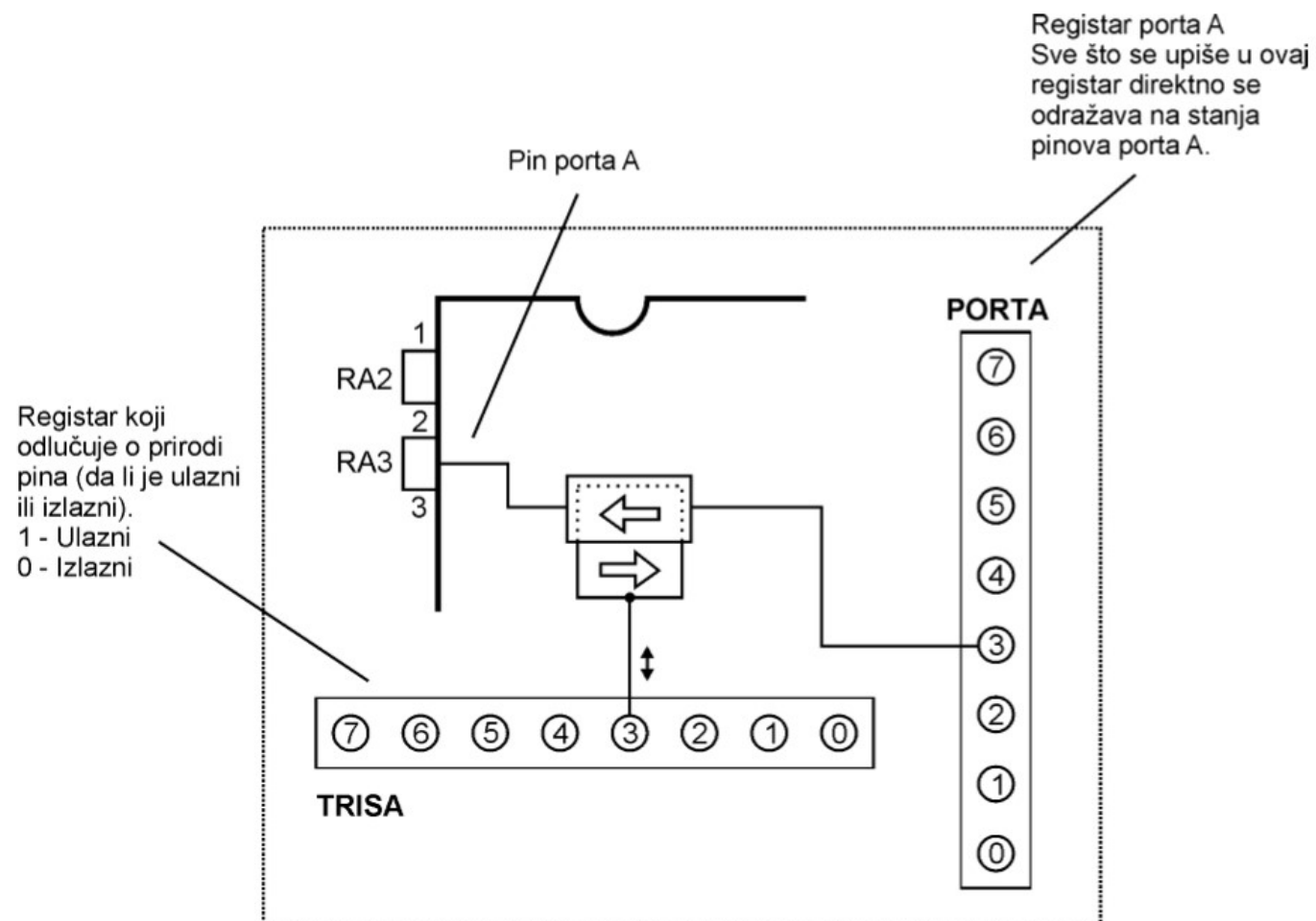
# Portovi

- Fizička veza centralne procesorske jedinice sa spoljnim svetom.
- Svi pinovi portova mogu se definisati kao ulazni ili kao izlazni, već prema potrebama uređaja koji se razvija.
- PIC16F84 ima dva porta: PORTA i PORTB.
- Da bi definisali pin kao ulazni ili kao izlazni mora se u registar TRIS upisati odgovarajuća kombinacija nula i jedinica. Ako je na odgovarajućem mestu u TRIS registru upisana logička jedinica "1" onda je taj pin ulazni, u obratnom slučaju pin je izlazni.
- Svaki port ima svoj odgovarajući TRIS registar. Tako port A ima TRISA a port B TRISB. Promena smera pinova se može vršiti u toku rada što je posebno pogodno za slučaj komunikacije preko jedne linije gde se menja smer prenosa podataka.
- Registri stanja portova PORTA i PORTB se nalaze u banci 0 dok se registri za smer pinova TRISA i TRISB nalaze u banci 1.

# PORTA

- Ima 5 pinova koji su mu pridruženi
- Odgovarajući registar za smer podataka je TRISA
- RA4 pin
  - Na tom pinu se još nalazi i spoljni ulaz za brojač TMR0.
  - Da li se ovaj pin bira kao standardni ulazni pin ili kao ulaz brojača koji se vodi na TMR0 zavisi od bita T0CS (TMR0 Clock Source Select bit).
  - Ovaj bit omogućava da brojač TMR0 uvećava svoje stanje ili iz internog oscilatora ili preko spoljnih impulsa na pinu RA4/T0CKI.





Sl. 2.14. Odnos TRISA i PORTA registra

# PORTB

- PORTB ima 8 pinova koji su mu pridruženi.
- Odgovarajući registar za smer podataka je TRISB
- Svaki pin na PORTB ima slabi interni pull-up otpornik (otpornik koji definiše liniju na logičku jedinicu) koji se može aktivirati resetovanjem sedmog bita RBPU u registru OPTION.
- Ovi "pull-up" otpornoci se automatski isključe kada je pin porta konfigurisan kao izlaz. Pri uključenju mikrokontrolera pull-up -ovi su onemogućeni.
- RB7:RB4 mogu izazvati prekid (interapt) koji se dešava kada se stanje na njima promeni iz logičke jedinice u logičku nulu i obratno.
- RB0 se koristi takođe kao izvor prekida (rastuća ili opadajuća ivica)

# Mašinski jezik mikrokontrolera PIC16F84

- RISC, redukovan skup instrukcija
- 14-bit instrukcije
- U zavisnosti od tipa, mogu sadržati sledeće elemente
  - f-Memorijski (fajl) registar (7-bit)
  - d-Destinacioni bit (1-bit)
    - d=0 :destinacija akumulator (w)
    - d=1 :destinacija memorijski registar
  - b-kod odgovarajućeg bita u registru (3-bit)
  - k-konstanta (literal)
    - 8-bit za operacije za rad sa memorijskim registrima
    - 11-bit za adrese bezulsovnih skokova

# Tipovi mašinskih instrukcija PIC16F84

- Tri grupe:
  - Bajt-orijentisane za rad sa memorijskim registrima
  - Bit-orijentisane za rad sa memorijskim registrima
  - Kontrolne instrukcije i rad sa literalima
- Sve instrukcije se izvršavaju u jednom taktu procesora, izuzev
  - Uslovnih i bezuslovnih skokova
  - Poziva i povratka iz potprograma
  - Ove instrukcije se izvršavaju u dva takta

# Bajt-orijentisane naredbe za rad sa memorijskim registrima

13	8	7	6	0
Operacioni kod		Destinacija (d)		Fajl registar (f)

- MOVF f,d
  - d=0->w:=f
  - d=1->f:=f (korisno za testiranje ZERO FLAG)
- SWAPF f,d
  - Niža i viša četvorka osmobitnog REGISTRA menjaju mesta
- ADDWF, ANDWF, SUBWF, IORWF, XORWF
  - Za SUBWF f,d recimo, kada je d=0 efekat je w:=f-w, a za d=1 f:=f-w
- COMF REGISTRAR,d
  - d=0->w:=komplement(REGISTAR)
  - d=1->REGISTAR:=komplement(REGISTAR)
- INCF f,d/DECF f,d
- RRF f,d
  - Sadržaj registra se rotira za jedan bit udesno kroz CARRY FLAG.
- RLF f,d
  - Sadržaj registra se rotira za jedan bit ulevo kroz CARRY FLAG.

# Bit-orijentisane naredbe za rad sa memorijskim registrima

13	10	9	7	6	0
Operacioni kod		Bit kod (b)			Fajl registar (f)

- BTFSS f,b
  - Testira bit b u registru f. Preskače narednu instrukciju ako je  $f[b] == 1$ .
- BTFSC f,b
  - Testira bit b u registru f. Preskače narednu instrukciju ako je  $f[b] == 0$ .
- BSF f,b
  - Postavlja b-ti bit registra na vrednost 1.
- BCF f,b
  - Postavlja b-ti bit registra na vrednost 0.

# Rad sa literalima

13	8	7	0
Operacioni kod		Literal (k)	

- k je u ovom slučaju 8-bit konstanta
- MOVLW k
  - $w := k$
- ADDLW k, ANDLW k, SUBLW k, IORLW k, XORLW k
  - $w := w + k$
  - $w := w \text{ and } k$
  - $w := k - w$
  - ...
- RETLW k
  - Povratak iz interapta uz  $w := k$

# Rad sa literalima

13	11	10	0
Operacioni kod		Apsolutna adresa skoka (k)	

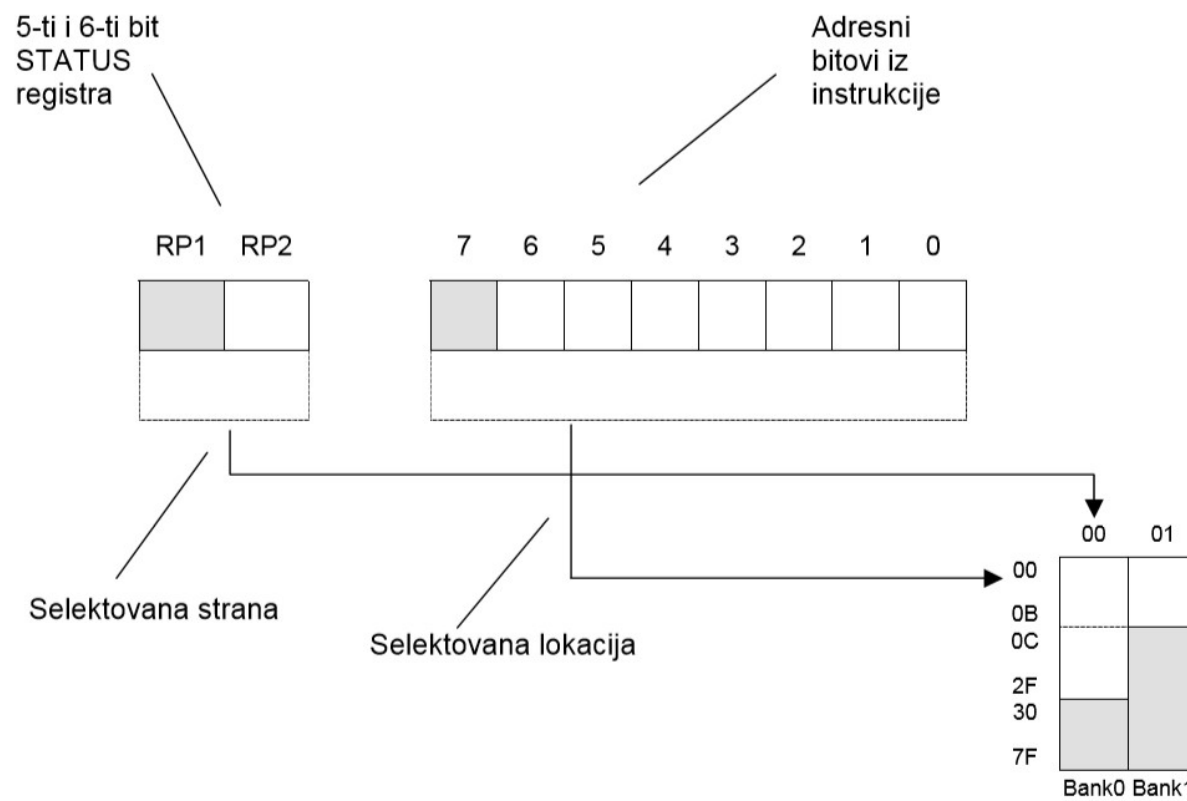
- k je u ovom slučaju 11-bit konstanta
- GOTO k
  - $w := k$
  - Bezuslovni skok
- CALL k
  - Poziv potprograma



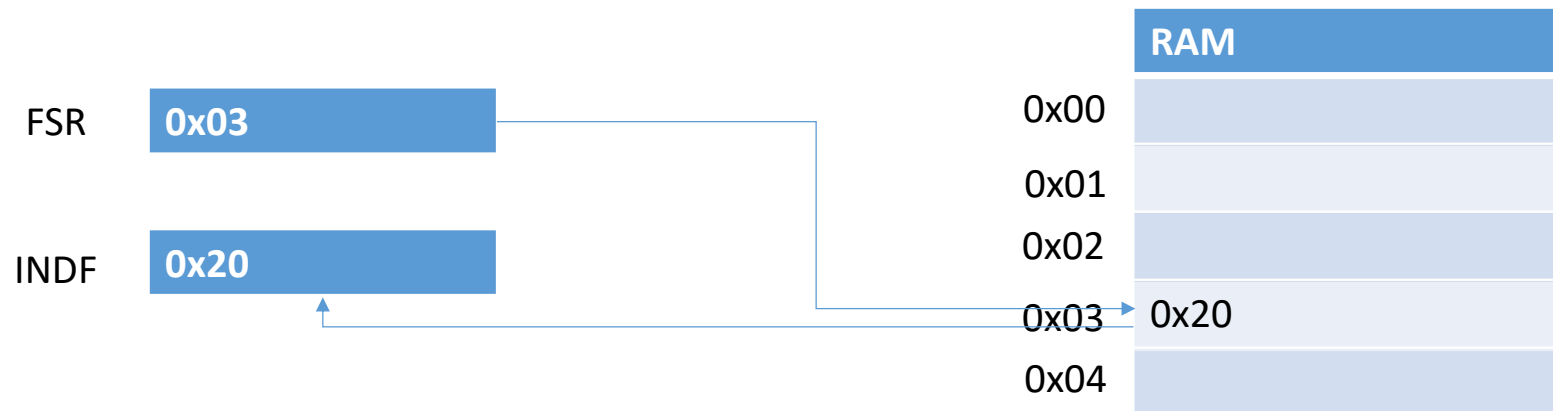
# Selekcija memorijske banke

- Za pristup specijalnim registrima potrebno je pored navođenja imena registra u okviru instrukcije prethodno selektovati odgovarajuću banku.
- Selekcija banke se može vršiti i pomoću direktive banksel posle koje se navodi ime registra kome se pristupa. Na ovaj način nema potrebe da se pamti koji je registar u kojoj banci.
  - banksel TRISB ; Pistupi banci u kojoj je TRISB clrf TRISB ; Izvrsi operaciju nad registrom TRISB
  - banksel POTRB ; Pristupi banci u kojoj je PORTB clrf PORTB ; Izvrsi operaciju nad registrom PORTB
- Primer: bcf STATUS,RPO
  - Efekat: Instrukcija BCF resetuje bit RPO (RPO=0) u STATUS registru i time selektuje banku 0, koja ostaje selektovana sve dok se RP= ne postavi na jedan.
- Primer: bsf STATUS,RPO Instrukcija BSF setuje bit RPO
  - Efekat: Instrukcija BSF setuje bit RPO (RPO=1) u STATUS registru i time selektuje adresiranje registara iz banke1.

# Direktno adresiranje



# Indirektno adresiranje



# FSR i INDF registri

- Indirektno adresiranje se ostvaruje pomoću INDF i FSR registra
- FSR čuva adresu
- Korišćenjem INDF dobijamo vrednost na adresi iz FSR
- Primer: Ako na adresi 0Fh imamo vrednost 20, upisom vrednosti 0Fh u registar FSR dobićemo pokazivač na registar na adresi 0Fh, a čitanjem iz registra INDF dobijamo vrednost 20, što znači da smo iz prvog registra pročitali njegovu vrednost a da mu nismo direktno pristupili (već preko FSR i INDF).
- Indirektno adresiranje je veoma pogodno kada se vrše operacijama sa nizovima podataka koji su smešteni u okviru GPR registra. U tom slučaju je na početku potrebno inicijalizovati registar FSR na vrednost adrese prvog člana niza, a zatim se narednim članovima niza pristupa uvećanjem registra FSR.

# Stek

- 13-bitni stek (Stack) sa 8 nivoa ili drugim rečima, grupisanih 8 memorijskih lokacija širine 13 bita sa posebnom namenom.
- Njegova osnovna uloga je da sačuva vrednost programskog brojača nakon što se iz glavnog programa skoči na adresu podprograma koji se izvršava.
- Da bi program znao da se vrati na mesto odakle je pošao, mora sa steka da vrati vrednost programskog brojača. Pri prelasku iz programa u podprogram, programski brojač se potiskuje na stek (Primer je instrukcija CALL), a pri izvršenju instrukcija kao što su RETURN, RETLW ili RETFIE koje se izvršavaju na kraju podprograma, vraća sa steka da bi program mogao da nastavi tamo gde je stao pre nego što je bio prekinut.
- Ove operacije stavljanja i vraćanja sa steka programskog brojača u žargonu se nazivaju PUSH i POP po instrukcijama koje pod istim imenom postoje na nekim većim mikrokontrolerima.

# Programski brojač

- Programski brojač (PC) je 13-to bitni registar koji sadrži adresu instrukcije koja se izvršava.
- Fizički se realizuje pomoću petobitnog registra PCLATH koji predstavlja pet viših bitova adrese i osmobitnog registra PCL koji predstavlja nižih osam bita adrese.
- Njegovim uvaćanjem ili promenom (npr. u slučaju skoka) mikrokontroler izvršava jednu po jednu instrukciju programa.

# Incijalizacija prekida

clrf INTCON ; svi prekidi onemogućeni

movlw B'00010000' ; omogućen samo spoljni prekid

movwf INTCON

bsf INTCON, GIE ; dozvoljena pojava prekida

# Čuvanje sadržaja važnih registara

- Za vreme prekida, samo se povratna vrednost programskog brojača čuva na steku (pod povratnom vrednošću programskog brojača podrazumeva se adresa instrukcije koja je trebala da se izvrši, ali nije se izvršila jer se prekid desio)
- Procedura snimanja važnih registara pre odlaska u prekidnu rutinu u žargonu se naziva "puš" (PUSH), dok se procedura vraćanja snimljenih vrednosti naziva "pop" (POP). PUSH i POP su instrukcije kod nekih drugih mikorokontrolera (Intel), ali su toliko prihvaćene da se po njima naziva čitava operacija. PIC16F84 nema instrukcija kao što su PUSH i POP i one se moraju programski napraviti
- Za razmenu podataka između registara koristi se instrukcija SWAPF umesto MOVF jer ona ne utiče na stanje bitova STATUS registra
- SWAP f,d : zamena više i niže tetrade u registru f



---

Prekidni\_program:

```
    movwf    WREG_TEMP    ;save WREG
    movwf    WREG_TEMP    ;save WREG
    swapf    STATUS,W     ;store STATUS in WREG
    clrf     STATUS       ;select file register bank0
    movwf    STATUS_TEMP  ;save STATUS value
```

;test interrupt flags here

;-----

;resave routine

EndInt:

```
    swapf    STATUS_TEMP,W ;get saved STATUS value
    movwf    STATUS        ;restore STATUS
    swapf    WREG_TEMP,F   ;prepare WREG to be
restored
    swapf    WREG_TEMP,W   ;restore WREG without
affecting STATUS
    retfie    ;return from interrupt
    end
;kraj primera
```

# Čitanje EEPROM memorije

- Setovanje bita RD inicira prenos podataka sa adrese koja se nalazi u registru EEADR u EEDATA registar.
- Kako za čitanje podataka nije potrebno vreme kao za upis, preuzeti podatak iz EEDATA registra može se već u narednoj instrukciji koristiti dalje.
- Primer čitanja sadržaja EEPROM memorije
  - `bcf STATUS, RPO ;bank0, jer je EEDAR na 09h`
  - `movlw 0x00 ;adresa lokacije koja se čita`
  - `movwf EEADR ;adresa se prebacuje u EEADR`
  - `bsf STATUS, RPO ;bank1 jer je EECON1 na 88h`
  - `bsf EECON1, RD ;čitanje uz EEPROM-a`
  - `bcf STATUS, RPO ;Bank0 jer je EEDATA na 08h`
  - `movf EEDATA, W ;W <-- EEDATA`
- Nakon poslednje programske instrukcije, sadržaj sa adrese nula EEPROM se nalazi u radnom registru w.

# CONFIGURATION WORD

**REGISTER 6-1: PIC16F84A CONFIGURATION WORD**

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	PWRTÉ	WDTE	F0SC1	F0SC0
bit13											bit0		

- bit 13-4      **CP:** Code Protection bit  
                  1 = Code protection disabled  
                  0 = All program memory is code protected
- bit 3          **PWRTÉ:** Power-up Timer Enable bit  
                  1 = Power-up Timer is disabled  
                  0 = Power-up Timer is enabled
- bit 2          **WDTE:** Watchdog Timer Enable bit  
                  1 = WDT enabled  
                  0 = WDT disabled
- bit 1-0       **F0SC1:F0SC0:** Oscillator Selection bits  
                  11 = RC oscillator  
                  10 = HS oscillator  
                  01 = XT oscillator  
                  00 = LP oscillator