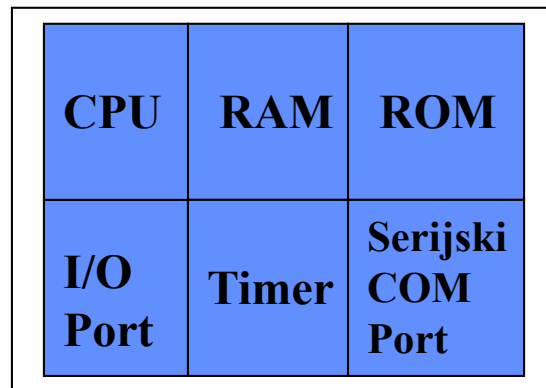


MIKROKONTROLER 8051



Mikrokontroler :

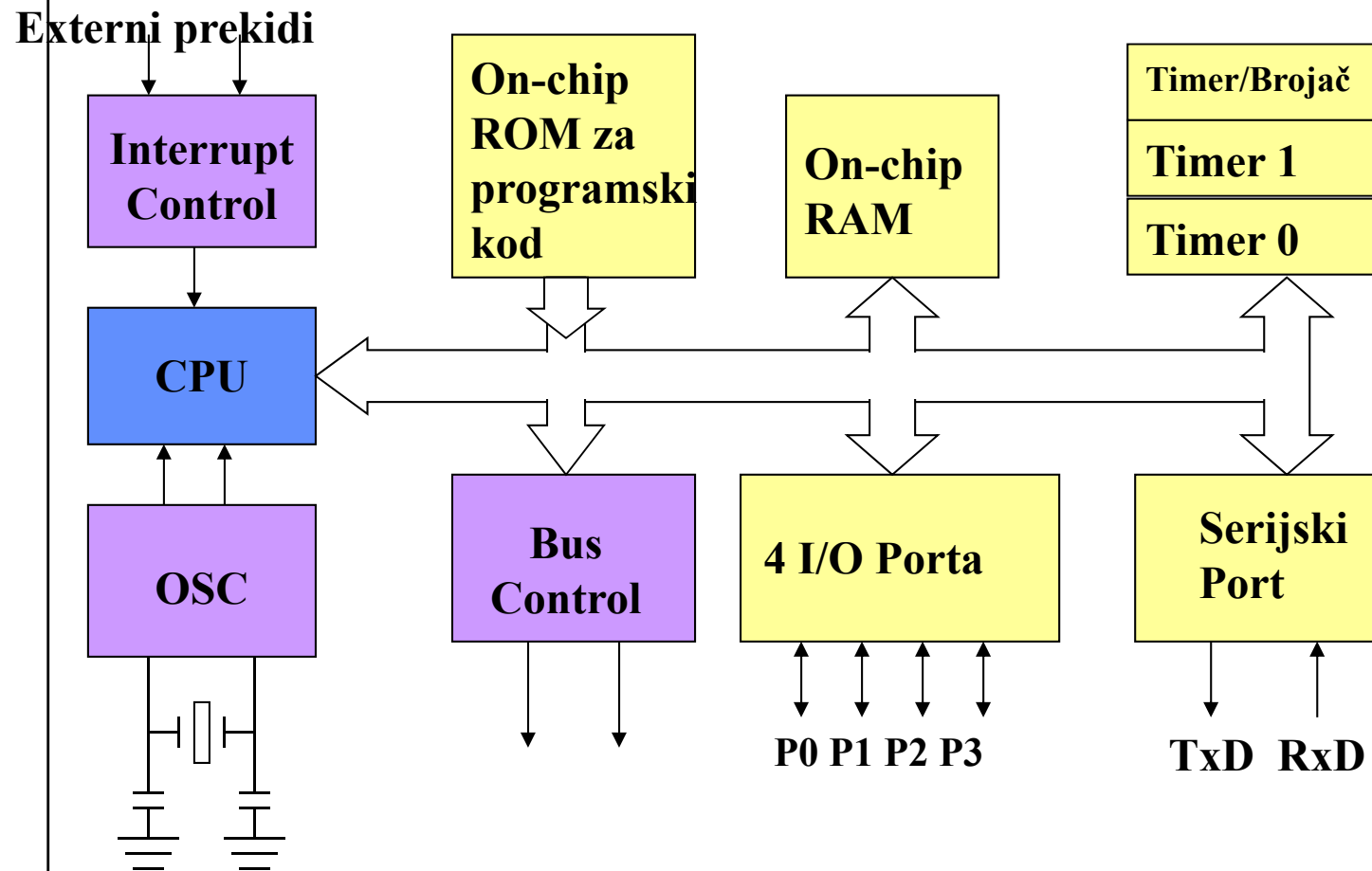
- ❑ Mali računar
- ❑ On-chip RAM, ROM, I/O portovi...
- ❑ Primjer: Motorola 6811, Intel 8051 i PIC 16X



← Jedan čip

Mikrokontroler

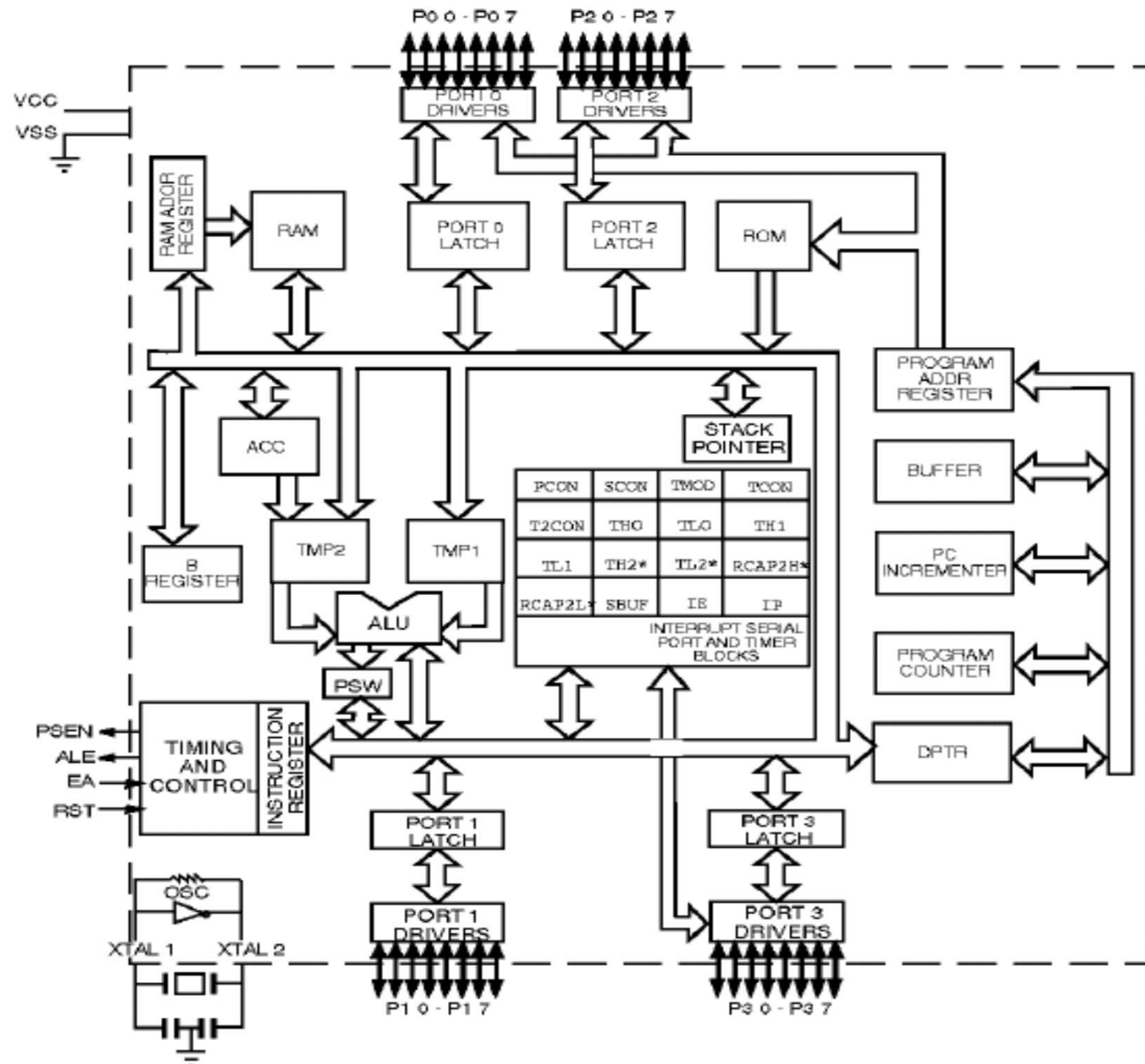
Osnovna arhitektura 8051



Neki članovi familije 8051

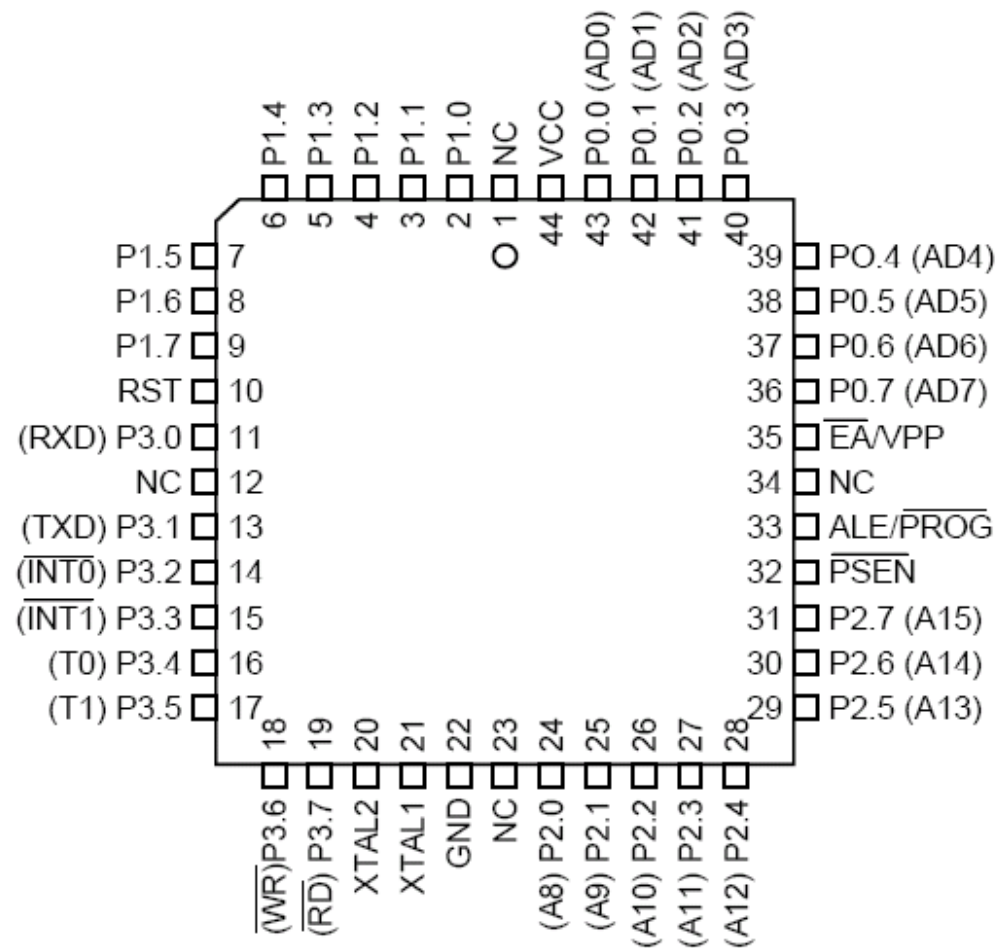
	8051	AT89LS8252	AT89LS53	AT89C51RB2/C2
ROM (u bajtovima)	4K	8K	12K	16K/32K
RAM (u bajtovima)	128	256	256	256+1024
Tajmeri	2	3	3	3
I/O pinova	32	32	32	32
Serijski port	1	1	1	1
Izvori prekida	6	6	9	9
SPI	NE	DA	DA	DA

Blok diagram arhitekture 8051



Pin Description za 8051

PLCC



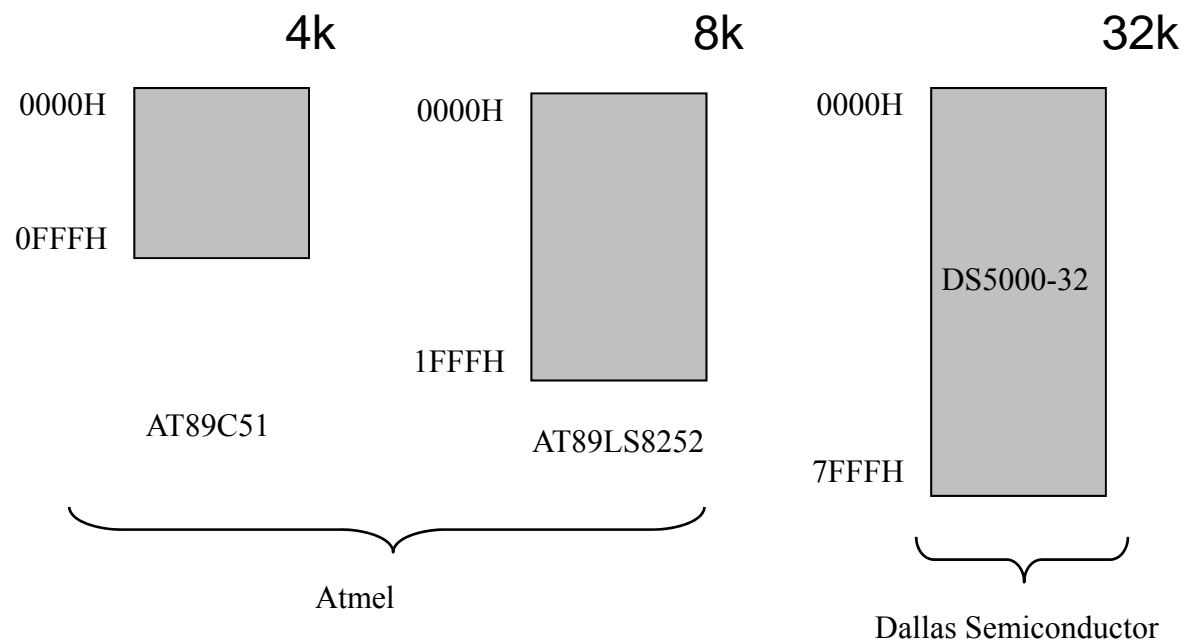
Pinovi 8051 1/2

- ❑ **Vcc** – pin 44:
 - Vcc obezbjeđuje napajanje za čip.
 - Izvor napajanja je +5V.
- ❑ **GND** – pin 22:
 - masa
- ❑ **XTAL1** i **XTAL2** – pinovi 21,20:
 - Ova 2 pina se koriste za externi oscilator
- ❑ **RST** – pin 10:
 - Reset – ulazni pin koji je aktivan high, inače low.
- ❑ **EA** i **PSEN** – pinovi 35 i 32:
 - Koriste se kada je programski kod nalazi u externj ROM memoriji

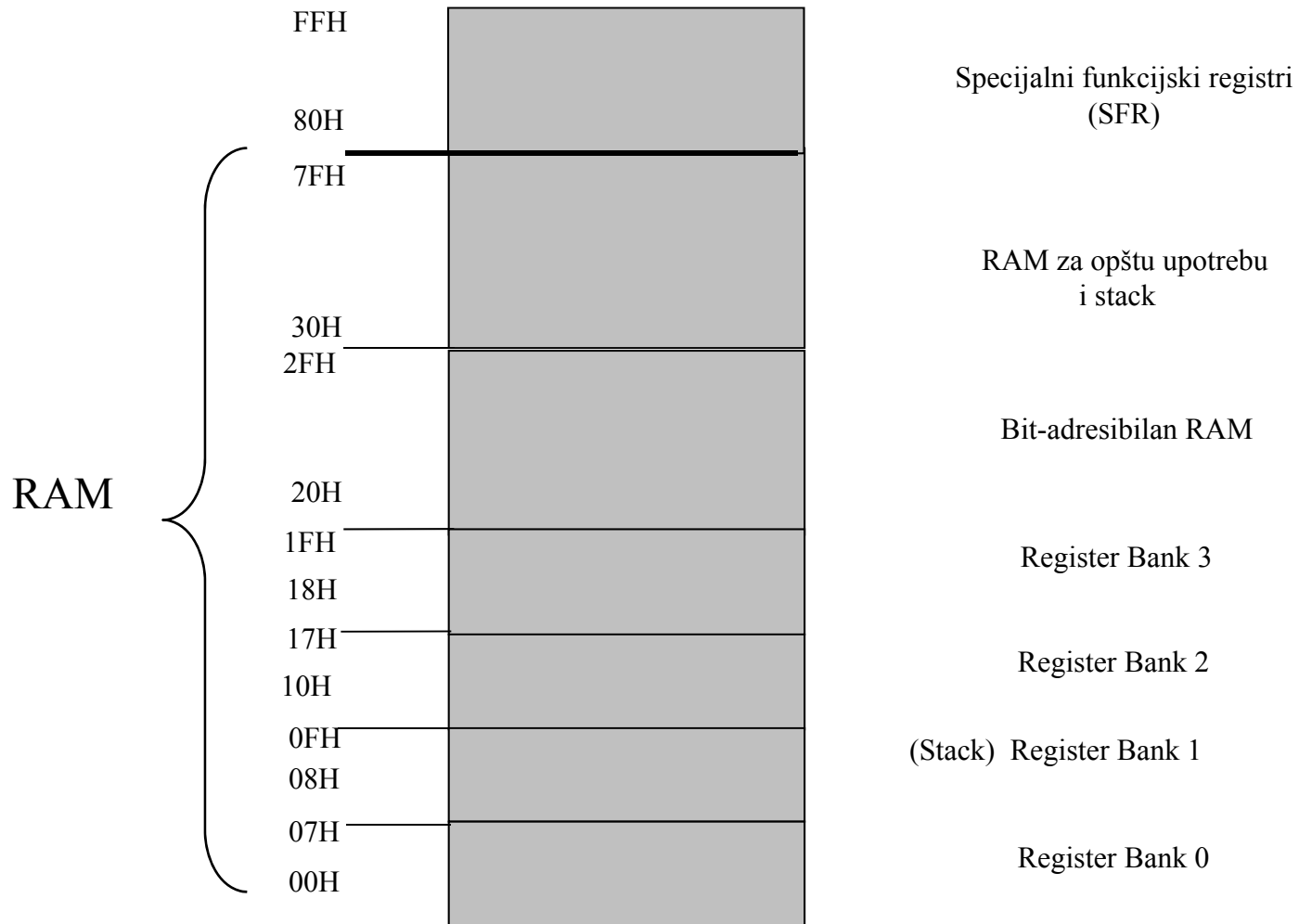
Pinovi 8051 2/2

- ❑ **ALE** - pin 33:
 - address latch enable – koristi se za demultiplexiranje adrese i podataka pri pristupu ekstornoj memoriji
- ❑ **I/O portovi:**
 - Četiri porta - P0 , P1, P2 i P3.
 - Svaki port koristi 8 pinova.
 - Svi I/O pinovi su dvosmerni.

Mapiranje memorije kod 8051



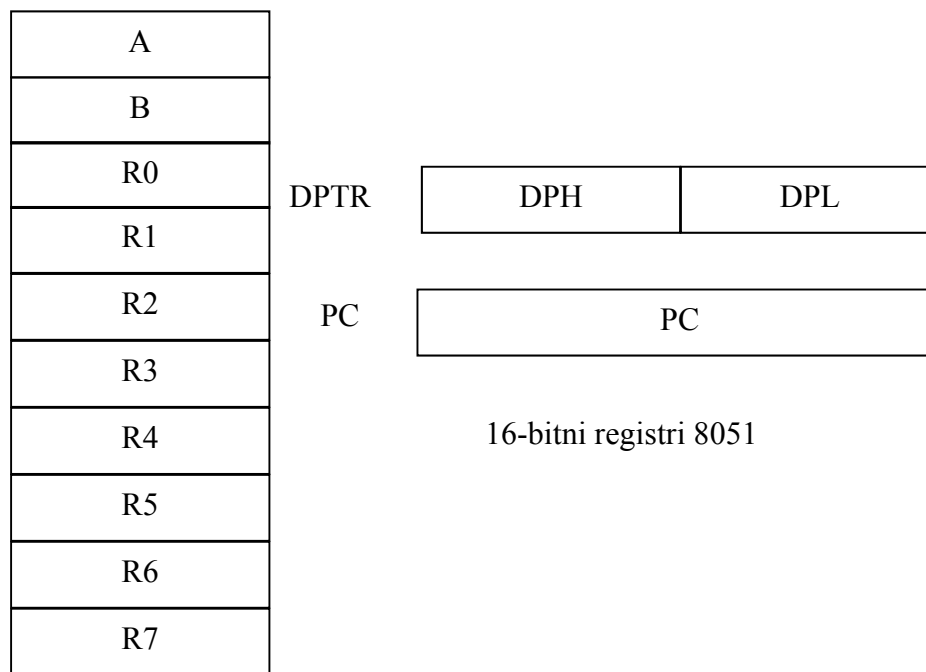
On-chip memorija kod 8051



Specijalni funkcijski registri

	Bit Addressable	8 Bytes Non-bit Addressable							
F8h									FFh
F0h	B								F7h
E8h									EFh
E0h	ACC								E7h
D8h									DFh
D0h	PSW								D7h
C8h	(T2CON)		(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CFh
C0h									C7h
B8h	IP								BFh
B0h	P3								B7h
A8h	IE								AFh
A0h	P2								A7h
98h	SCON	SBUF							9Fh
90h	P1								97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	CKCON	8Fh
80h	P0	SP	DPL	DPH				PCON	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Osnovni registri kod 8051



16-bitni registri 8051

Neki 8-bitni registri 8051

Osnovni registri – 8 bitni

A - Akumulator

Akumulator se koristi kao opšti registar u koji se smješta rezultat velikog broja instrukcija. To je 8-bitni registar i on je registar koji se najčešće koristi pri izvršavanju programa pošto više od polovine instrukcija koje podržava 8051 koristi akumulator na neki način.

Npr. Ako hoće da se broju 10 doda 20, rezultat 30 će da se upiše u akumulator. Kada je izračunata vrijednost u akumulatoru, može da se nastavi dalje obrađivati ili može da se upiše u neki drugi registar ili memoriju.

"R" registri

"R" registri su grupa od osam registara koji imaju imena R0, R1, ..., R7. Ovi registri se koriste kao pomoćni registri u velikom broju operacija. Ako se uzme prethodni primjer, gdje se broju 10 dodaje broj 20, broj 10 može biti upisan u akumulatoru, a broj 20 u npr. registru R4. U tom slučaju bi se sabiranje izvršilo na sljedeći način:

ADD A,R4

Kada se izvrši instrukcija akumulator će da sadrži vrijednost 30. Ovi registri su veoma važni za privremeno smještanje promjenjivih.

"B" Registar

"B" registar je sličan akumulatoru, takođe je 8-bitni registar, a koristi se samo za dvije instrukcije MUL AB (množenje) i DIV AB (dijeljenje). Ako se žele pomnožiti ili podijeliti dva 8-bitna broja, prvi broj se upiše u akumulator, drugi u B registar i pozove se odgovarajuća od ove dvije instrukcije. Pri pozivu množenja, niži bajt proizvoda smješta se u akumulator, a viši u registar B. Pored toga što se koristi kod množenja i dijeljenja, B registar se često može iskoristiti i kao još jedan registar za privremeno smještanje promjenjivih, slično kao R registri.

Osnovni registri – 16 bitni

DPTR - Data Pointer

Data Pointer (DPTR) je jedini 16-bitni registar u 8051 kome korisnik može da pristupi. DPTR se, kao što mu ime nagovještava, koristi kao pokazivač na podatke. Koristi se kod komandi kojima program u 8051 pristupa eksternoj memoriji.

PC - Program Counter

Program Counter (PC) je 16-bitna adresa koja govori 8051 gdje se u memoriji nalazi sljedeća instrukcija koja treba da se izvrši. Kada se 8051 inicijalizuje, uvijek počinje od adrese 0000h i inkrementira se svaki put kada se instrukcija izvrši. Bitno je napomenuti da se PC ne inkrementira uvijek samo za 1. Pošto neke instrukcije zauzimaju 2 ili 3 bajta u memoriji, PC će u tim slučajevima biti inkrementiran za 2 ili 3.

PC se razlikuje od drugih registara pošto ne postoji mogućnost da mu se direktno promjeni vrijednost, odnosno ne može da se uradi npr. PC=2430h. Postoje, naravno, komande programskog skoka koje to rade na indirektan način. Takođe, ne postoji načina da se direktno pročita vrijednost iz PC, odnosno da se odredi gdje se nalazi adresa komande koja upravo treba da se izvrši.

PSW registar

CY	AC	F0	RS1	RS0	OV	--	P
----	----	----	-----	-----	----	----	---

<i>Carry flag</i>	PSW.7	CY
<i>Auxiliary carry flag</i>	PSW.6	AC
<i>Slobodan</i>	PSW.5	--
<i>Register Bank selector bit 1</i>	PSW.4	RS1
<i>Register Bank selector bit 0</i>	PSW.3	RS0
<i>Overflow flag</i>	PSW.2	OV
<i>Slobodan</i>	PSW.1	--
<i>Parity flag Set/Reset odd/even parity</i>	PSW.0	P

RS1	RS0	Register Bank	Adresa
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

8051 ima 4 različite registar banke za registre R0 do R7 koje se nalaze u prva 32 bajta RAM-a. Kada se 8051 pokrene, registar banka 0 (adreses 00h do 07h) se koristi kao default. Ukoliko se koristi samo jedna registar banka (npr. reg. banka 0) ostatak memorijskog prostora iz prva 32 bajta može se koristiti kao RAM opšte namjene.

RESET vrijednosti nekih registara kod 8051

Registar	Reset Vrijednost
PC	0000
ACC	0000
B	0000
PSW	0000
SP	0007
DPTR	0000

RAM je čitav na nuli.

Neke jednostavne instrukcije

MOV odredište,izvor ; odredište = izvor

MOV A,#72H ;A=72H
MOV R4,#62H ;R4=62H
MOV R4,A ;R4=72H

MOV DPTR,#7634H
MOV DPL,#34H
MOV DPH,#76H

Napomena:

MOV A,#72H ≠ **MOV A,72H**
Instrukcija “MOV A,72H” sadržaj 72 byta RAM-a upisaće se u Akumulator.

ADD A, Izvor ;A=A+Izvor

ADD A,#6 ;A=A+6
ADD A,R6 ;A=A+R6
ADD A,6 ;A=A+[6] odnosno A=A+R6
ADD A,08AH ;A=A+[08AH]

Neke jednostavne instrukcije – bit promjenive

SETB	bit	; bit=1
CLR	bit	; bit=0

SETB	P0.0	;bit 0 iz porta 0 = 1
SETB	P3.7	;bit 7 iz porta 3 = 1
SETB	ACC.2	;bit 2 iz akumulatora = 1
SETB	05	;bit 5 iz bajta na RAM lokaciji 20h = 0

CLR instrukcija je ista kao SETB

npr:

CLR	P0.0	;P0.0=0
CLR	ACC.1	;bit 1 iz akumulatora = 0
CLR	C	;CY=0

ali ako se CLR pozove za čitav akumulator, svi bitovi u akumulatoru postavljaju se na 0:

CLR	A	;A=0
-----	---	------

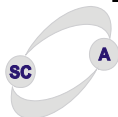
Memorijski prostor u RAM-u od 20h do 2Fh (16 bajtova) je bit adresabilan. Kada se koriste bitske instrukcije SETB i CLR, navodi se adresa bita koji se postavlja na 1 ili se briše. Adrese idu od 0 (0 bit bajta 20h) do 127 (7 bit bajta 2Fh)

Tako npr. **SETB 32** postavlja 0 bit bajta 24 u RAM-u na logičku "1".

Jedan dio SFR je takođe bit adresabilan (portovi, akumulator, ...). Bitovima u tim SFR može da se pristupi tako što se uz ime registra stavi .x gdje je x broj bita kome se pristupa ili preko imena samog bita, ako ga ima. Tako npr. **SETB P2.1** postavlja 1 bit u registru P2 na logičku "1".

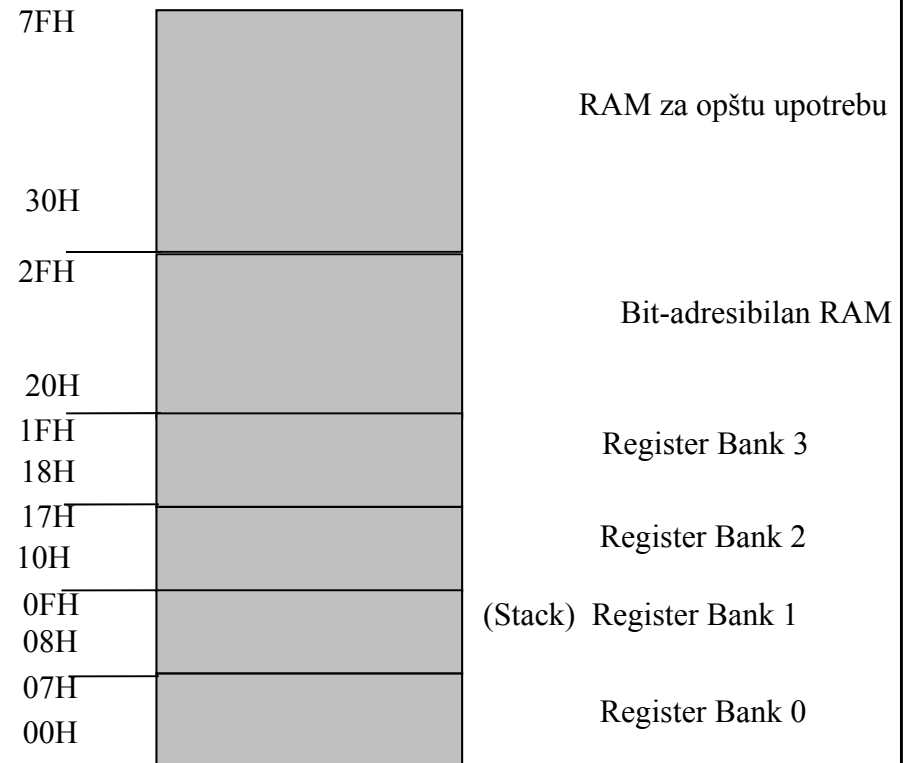
I bitovima u SFR koji su bit adresabilni može da se pristupi preko adresa. Adrese tih bitova su od 128 (bit 0 u P0) do 255 (7 bit SFR registra na adresi F8h).

Tako npr **SETB 129** postavlja 1 bit u registru P0 na logičku "1".



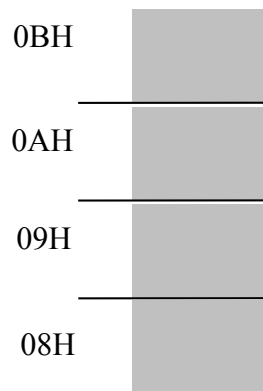
Stack u 8051

- ❑ Registar koji se koristi za pristup stacku zove se SP (stack pointer) registar.
- ❑ Stack pointer je u 8051 osmobaritni registar sto znaci da moze da ima vrijednosti od 00 do FFH. Kada se 8051 pokrene, SP registar sadrži vrijednost 07H.
- ❑ Pri pozivu podrutine i pri generisanju interapta, prije nego što ode na adresu podrutine ili adresu obrade interapta, 8051 upiše PC na stack. Pri završetku podrutine ili interapta, 8051 upisuje u PC vrijednost koju je sačuvao na stacku i nastavlja izvršavanje programa od adrese na kojoj je bio prekinut.

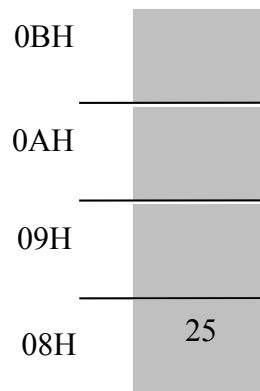


Primjer:

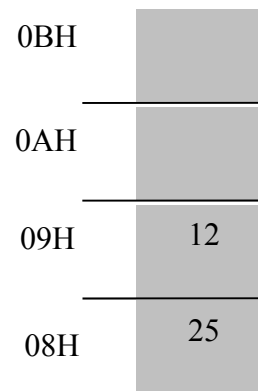
```
MOV    R6,#25H
MOV    R1,#12H
MOV    R4,#0F3H
PUSH   6
PUSH   1
PUSH   4
```



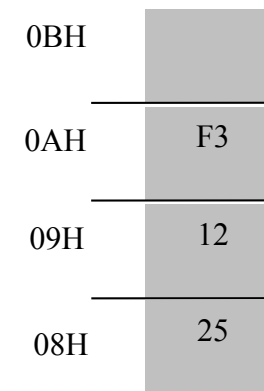
Start SP=07H



SP=08H



SP=09H



SP=0AH

Načini adresiranja

- ☐ **Neposredno adresiranje**
- ☐ **Adresiranje preko registara**
- ☐ **Direktno adresiranje**
- ☐ **Indirektno adresiranje**
- ☐ **Adresiranje eksterne memorije**
- ☐ **Adresiranje ROM memorije**

Neposredni način adresiranja

Neposredno adresiranje se zove tako jer vrijednost koja se upisuje u memoriju neposredno dolazi iza izvršnog koda u memoriji. Na taj način, instrukcija sama diktira koja vrijednost će biti upisana u memoriju

Na primjer, instrukcija:

MOV A,#20h

Ova instrukcija koristi neposredno adresiranje jer će u akumulator biti upisana vrijednost koja neposredno dolazi iza izvršnog koda - u ovom slučaju 20 (hexadecimalno). Na taj način u programskoj memoriji je upisano 7420. Neposredno adresiranje je veoma brzo jer se vrijednost koja se upisuje nalazi u samoj instrukciji. S druge strane, pošto je vrijednost koja se upisuje fiksno upisana pri kompajliranju, nema fleksibilnosti.

Adresiranje preko registara

Podacima u SFR i registrima R_n iz izabrane registar banke u RAM se može pristupiti preko imena registra

Na primjer:

MOV **R_n , A** ; $n=0,...,7$

ADD **A, R_n**

MOV **DPL, R6**

S tim da se ne može koristiti DPTR registar jer je on 16-bitni

Npr.

MOV **DPTR, A**

nije korektno

Takođe, ne mogu se koristiti istovremeno kao izvor i odredište registri iz RAM memorije što znači da

MOV **R_m , R_n**

nije korektno

Direktno adresiranje

Direktno adresiranje se zove tako jer se vrijednost koja se upisuje u memoriju dobija tako što se direktno preuzima sa neke memorijske lokacije.

Na primjer:

MOV A,30h

Ova instrukcija će da pročita podatke iz interne RAM adrese 30 (heksadecimalno) i upisati ih u akumulator. Direktno adresiranje je brzo jer, iako vrijednost koja se upisuje nije uključena direktno u instrukciju, brzo joj se pristupa posto je upisana u interni RAM. Takođe, ovaj način adresiranja je fleksibilniji od neposrednog adresiranja pošto se upisuju podaci koji se nalaze na datoj adresi i koji mogu da se mijenjaju.

Kada se koristi direktno adresiranje, sve instrukcije koje se pristupaju adresama između 00h i 7Fh odnose se na interni RAM. Sve instrukcije koje pristupaju adresama između 80h i FFh odnose se na SFR kontrolne registre koji kontrolišu rad mikrokontrolera.

Indirektno adresiranje

Indirektno adresiranje je veoma koristan način adresiranja koji u mnogim slučajevima pokazuje izuzetan nivo fleksibilnosti. Indirektno adresiranje se koristi na sljedeći način:

MOV A,@R0

Ova instrukcija će dovesti do toga da se u akumulator upiše vrijednost iz internog RAM-a koja se nalazi na adresi koja je upisana u R0. Na primjer, recima da se u R0 nalazi vrijednost 40h i da je na adresi 40h u internom RAM-u upisana vrijednost 67h. Kada se navedena instrukcija izvrši, mikrokontroler će da provjeri vrijednost koja je upisana u R0. Pošto je u R0 upisana vrijednost 40h, mikrokontroler će da preuzme vrijednost sa adrese 40h (na kojoj je upisano 67h) i upisati je u akumulator. Na taj način kada se instrukcija izvrši, u akumulator će biti upisana vrijednost 67h.

Indirektno adresiranje se uvijek odnosi na interni RAM, nikada se ne odnosi na SFR.

Ukoliko bi na osnovu prethodnog primjera željeli upisati vrijednost na port 1 kome se pristupa preko SFR registra P1 koji se nalazi na adresi 90 izgleda kao da bi to moglo da se uradi na sljedeći način:

MOV R0,#90h ;Upisivanje adrese porta 1

MOV @R0,#01h ;Upisivanje 01 na port 1 – GREŠKA!!!!

Ovo nije korektno. Pošto se indirektno adresiranje uvijek odnosi na interni RAM, ove dvije instrukcije će, u slučaju da se koristi kontroler 8051, proizvesti nedefinisan rezultat pošto on ima samo 128 bajtova internog RAM-a (adrese 00h do 7Fh).

Kontroleri bazirani na 8051 koji imaju proširenu funkcionalnost u odnosu na osnovu arhitekturu 8051, uglavnom imaju 256 bajtova internog RAM-a. Kod njih se nižim 128 bajtova, koji su isti kao kod osnovnog 8051 kontrolera, može pristupiti i direktno i indirektno. Memorijskom prostoru u internom RAM-u od 128 do 255 kod ovog kontrolera može da se pristupi samo indirektno. Na taj način, kod ovog kontrolera, kada se pristupa memorijskim lokacijama od 128 do 255 direktno, pristupa se SFR kontrolnim registrima, a kada se ovim lokacijama pristupa indirektno, pristupa se internom RAM-u.

Adresiranje eksterne memorije

Eksternoj memoriji se pristupa indirektno preko 16 bitnog registra DPTR ili registara R0 i R1. Za to se koristi komanda **MOVX**. Vrijednost se iz eksterne memorijske lokacije čija je adresa upisana u DPTR upisuje u akumulator pomoću naredbe:

MOVX A,@DPTR

Vrijednost iz akumulatora se upisuje u eksternu memoriju na adresu koja je upisana u DPTR pomoću naredbe:

MOVX @DPTR,A

Kao što se vidi, ove komande koriste DPTR da bi dobile informaciju o memorijskoj lokaciji kojoj se pristupa. To znači da u DPTR mora prvo da se upiše adresa eksterne memorijske lokacije koja želi da se čita ili na koju želi da se upiše.

Kada se pristupa externim memorijskim lokacijama od 00h do FFh može da se koristi naredba

MOVX @Ri,A

odnosno

MOVX A,@Ri

gdje je $i=0,1$

I u ovom slučaju, adresa eksterne memorijske lokacije kojoj želi da se pristupi mora prvo da se upiše u Ri (R0 ili R1). Pošto su R0 i R1 osmobarbitni registri, u njima mogu da se upišu vrijednosti od 0 do 255 tako da na ovaj način može da se pristupi samo najnižim 256 bajtova eksterne memorije.

Adresiranje ROM memorije

Ovaj način adresiranja se koristi za pristup look-up tabelama sa podacima koje se nalaze u ROM memoriji 8051

MOVC A,@A+DPTR

Preuzima se bajt koji se nalazi na ROM adresi koja se dobije kada se sabere sadržaj iz akumulatora sa sadržajem 16 bitnog registra DPTR i smješta u akumulator

Pinovi I/O portova

- ❑ 8051 ima četiri I/O porta
 - Port 0 – pinovi 36 do 43: P0.0 - P0.7
 - Port 1 – pinovi 2 do 9: P1.0 - P1.7)
 - Port 2 – pinovi 24 do 31: P2.0 - P2.7
 - Port 3 – pinovi 11, 13 do 19 : P3.0 - P3.7
 - Svaki port ima 8 pinova
 - Označavaju se P0.X ; X=0,1,...,7; P1.X, P2.X, P3.X
 - Npr. P0.0 je bit 0 – LSB od P0
 - Npr : P0.7 je bit 7 – MSB od P0
 - Ovih 8 bita formiraju byte.
- ❑ Svaki port može da se koristi kao ulaz ili izlaz (dvosmjerno)

Portovi

Port 0 – pristup eksternoj memoriji

- Niži bajt adrese / podaci

Port 2 – pristup eksternoj memoriji

- Viši bajt adrese

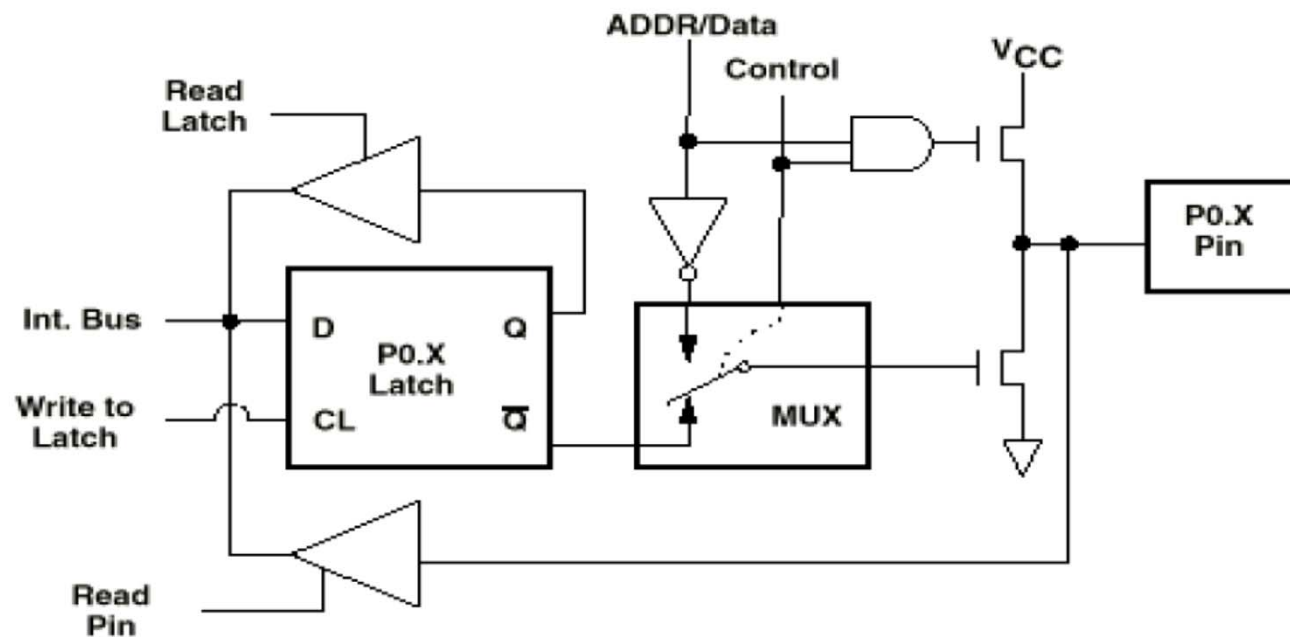
Port 1 – ulazi/izlazi opšte namjene

Port 3 – posebna namjena

- 0 - RxD: serial input
- 1 - TxD: serial output
- 2 - INT0: external interrupt 0
- 3 - INT1: external interrupt 1
- 4 - T0: timer/counter 0 external input
- 5 - T1: timer/counter 1 external input
- 6 - WR: external data memory write strobe
- 7 - RD: external data memory read strobe

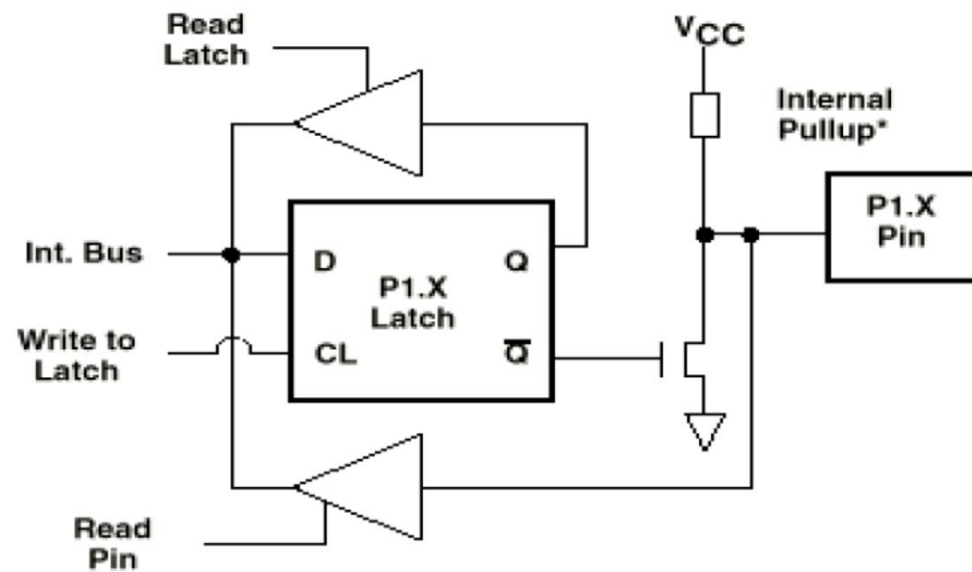
Da bi se pin porta koristio kao ulaz, u njegov latch mora da se upiše 1

Port 0 - bit

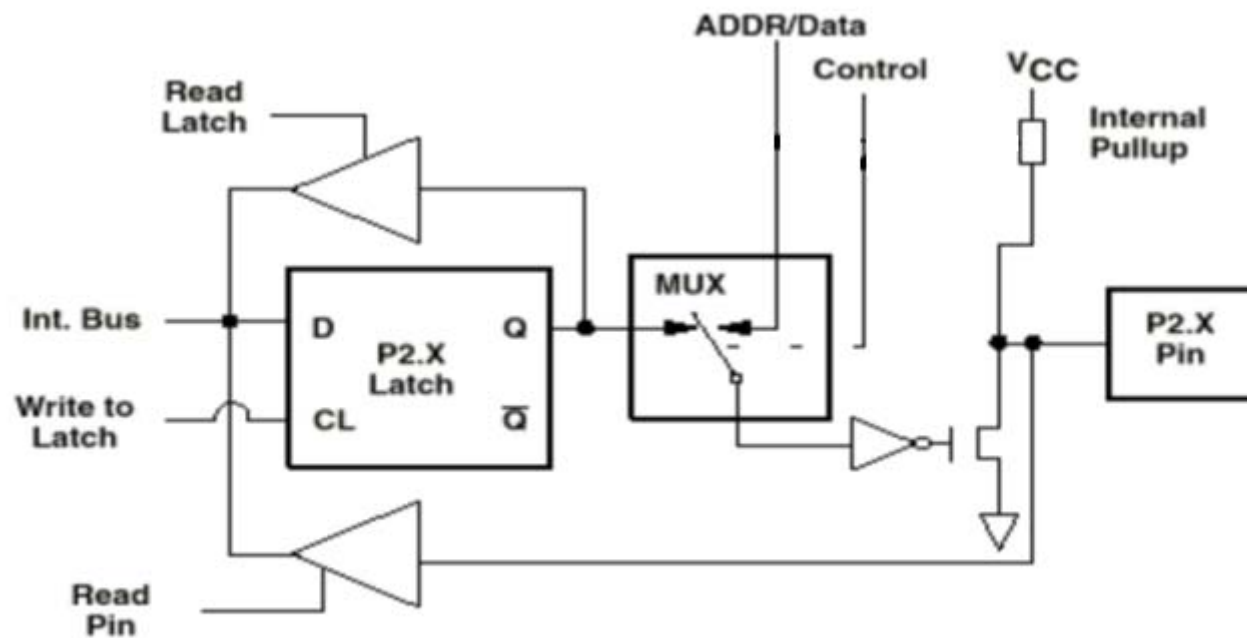


Potreban eksterni pullup

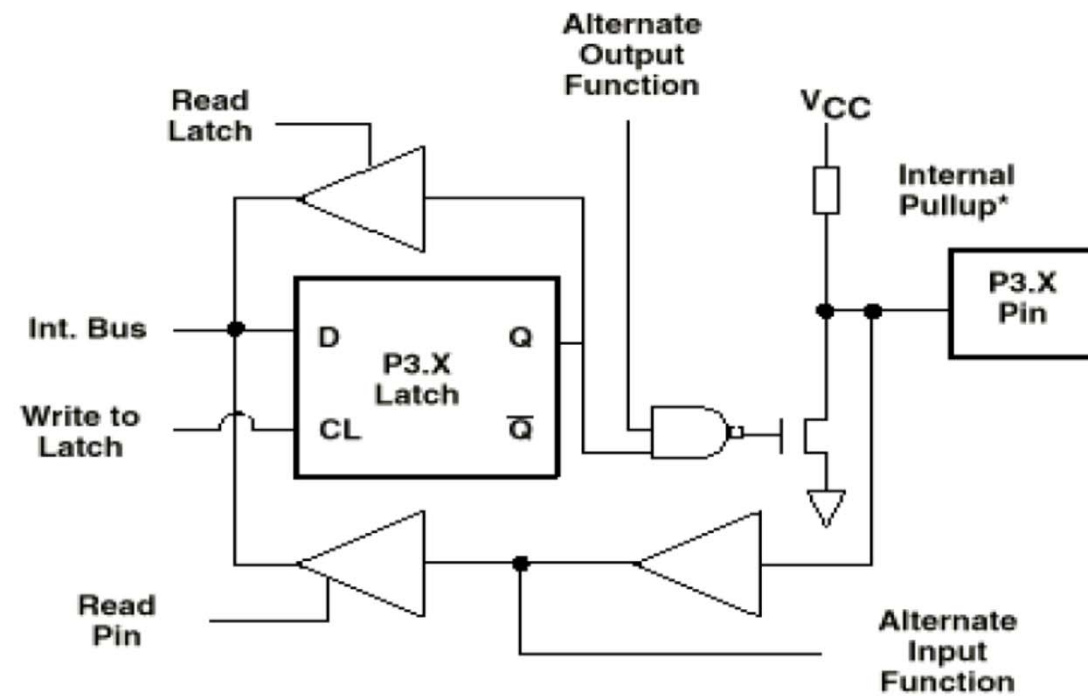
Port 1 - bit



Port 2 - bit



Port 3 - bit



Alternativna funkcija može da se aktivira samo ako je u latch upisana 1

Trajanje mašinskog ciklusa

- Jedan “mašinski ciklus” = 6 stanja (S1 - S6)
- Jedno stanje = 2 perioda oscilatora
 - Za jedan mašinski ciklus potrebno je 12 perioda oscilatora
- Većina instrukcija 8051 izvršava se za jedan mašinski ciklus (ADD, MOV, SETB, ...).
- Neke instrukcije traju 2 mašinska ciklusa (JMP, JZ, ...)
- 4 mašinska ciklusa traju samo množenje i dijeljenje (MUL, DIV)

Primjer: Odrediti trajanje mašinskog ciklusa za oscilatore

(a) XTAL = 11.0592 MHz

(b) XTAL = 16 MHz.

Rješenje:

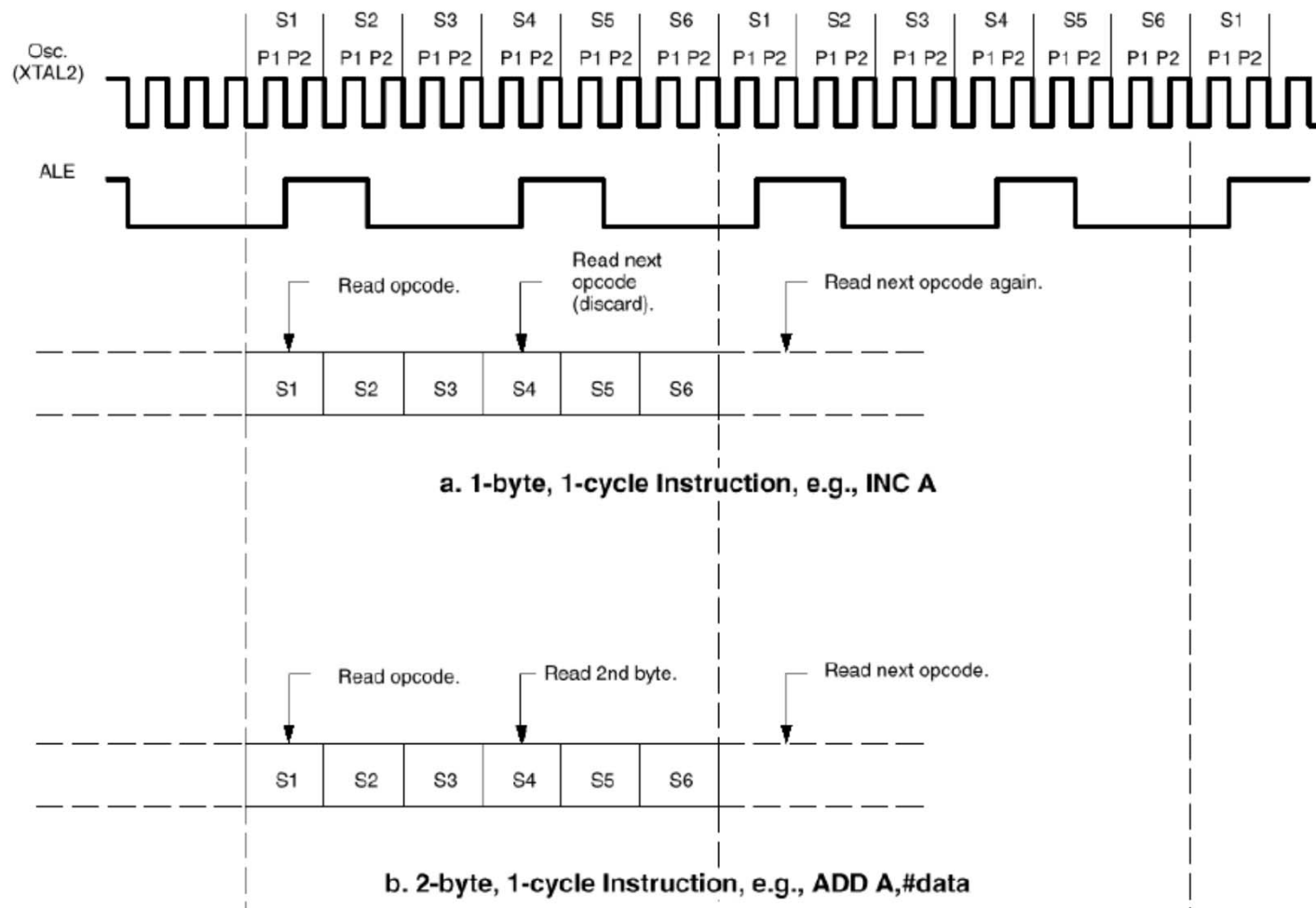
(a) $11.0592 \text{ MHz} / 12 = 921.6 \text{ kHz}$;

$T \text{ mašinskog ciklusa} = 1 / 921.6 \text{ kHz} = 1.085 \mu\text{s}$

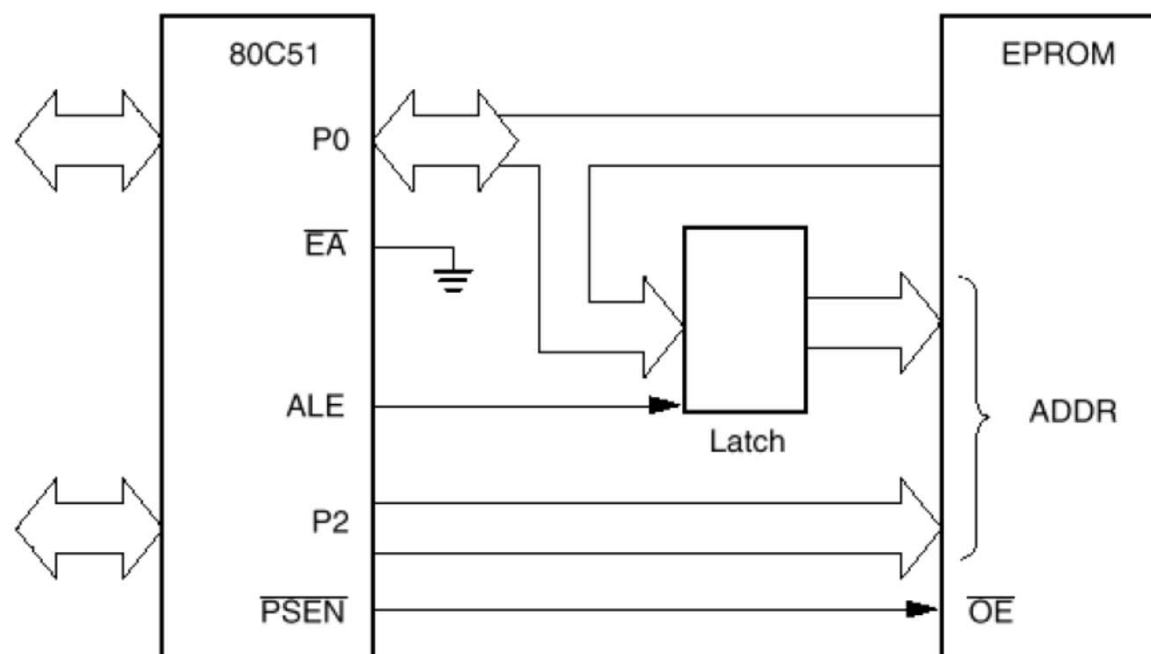
(b) $16 \text{ MHz} / 12 = 1.333 \text{ MHz}$;

$T \text{ mašinskog ciklusa} = 1 / 1.333 \text{ MHz} = 0.75 \mu\text{s}$

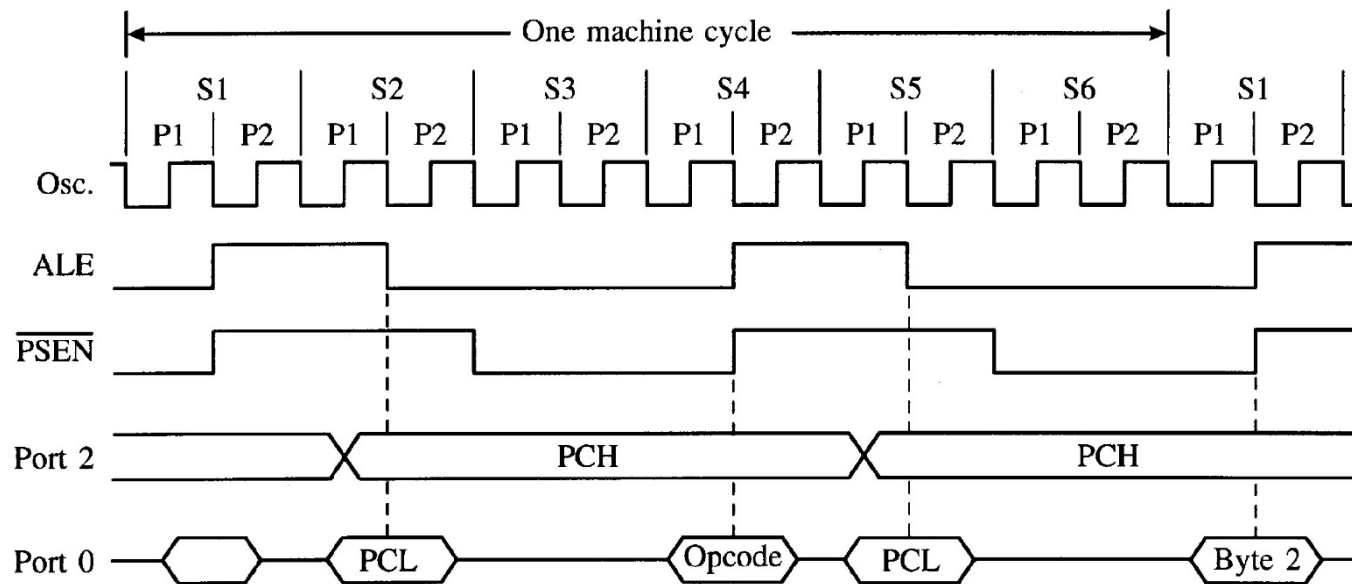
Izvršavanje instrukcije



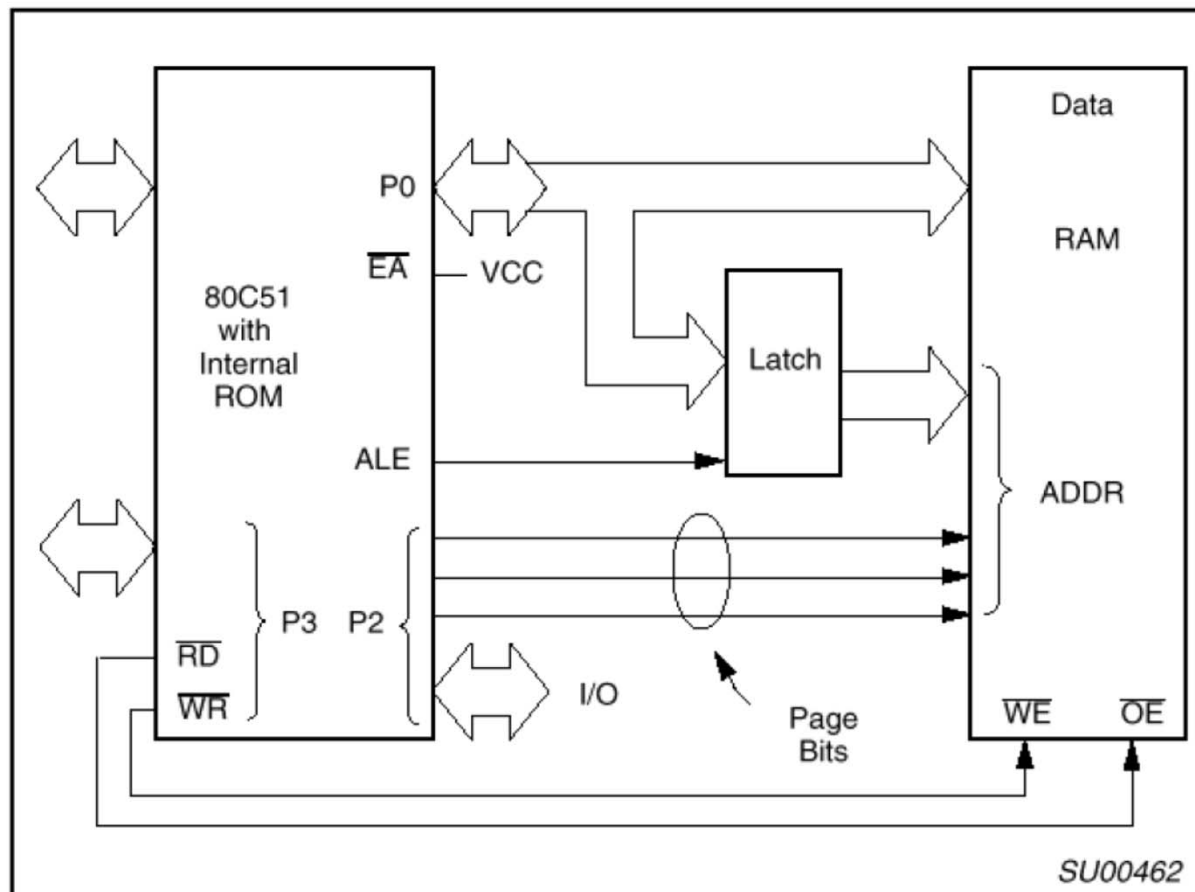
Pristup eksternoj programskoj memoriji 1/2



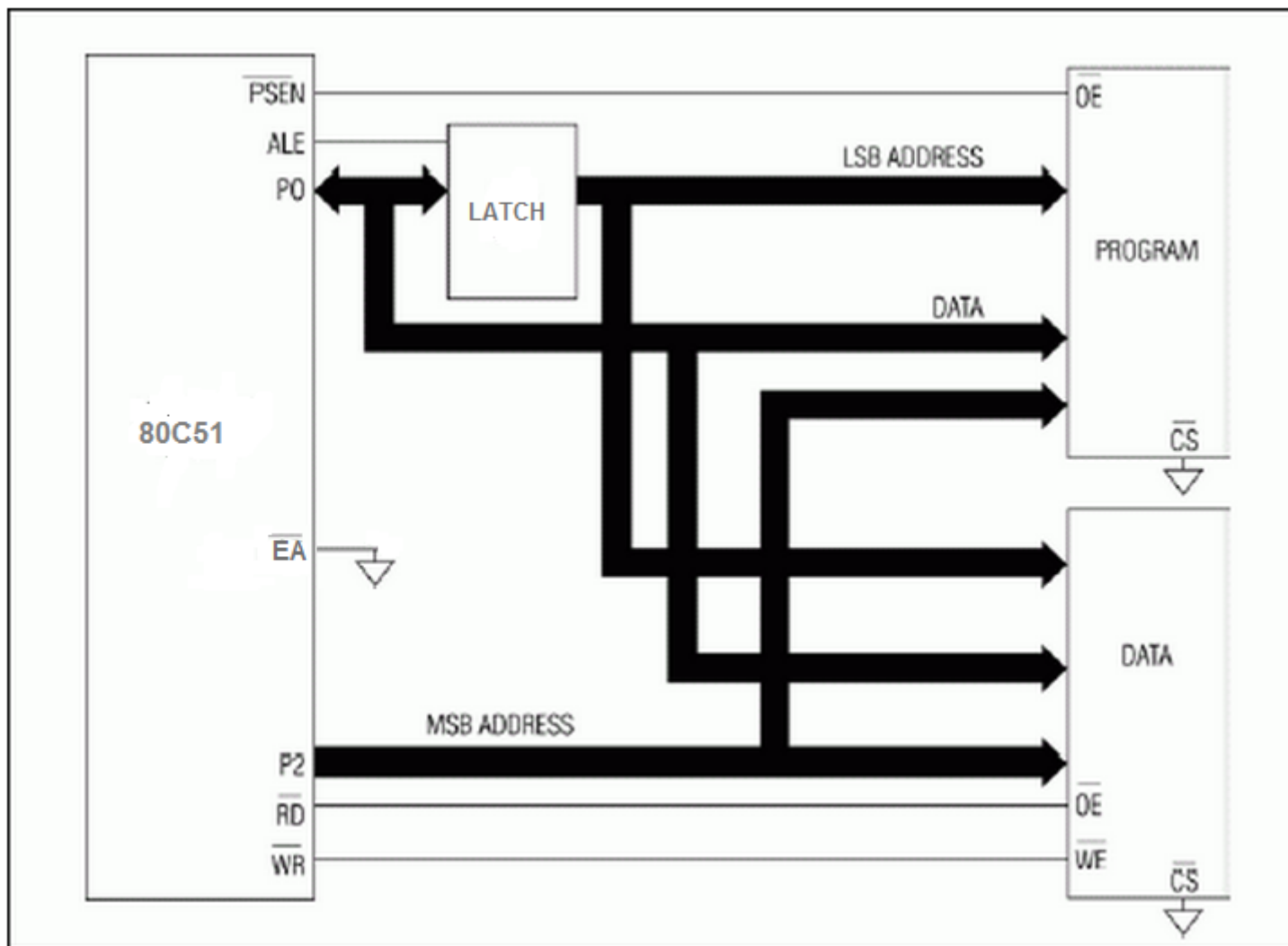
Pristup eksternoj programskoj memoriji 2/2



Pristup eksternoj radnoj memoriji 1/3



Pristup eksternoj radnoj memoriji 2/3



Pristup eksternoj radnoj memoriji 3/3

MOVX instrukcija

