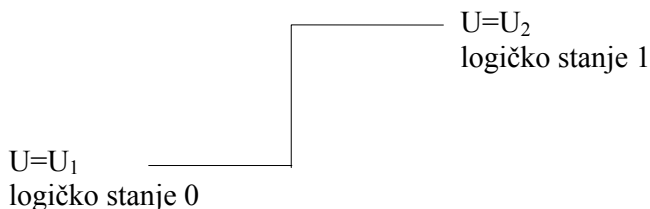


LOGIČKA KOLA

Kao što smo već istakli, obrada podataka u digitalnom računaru se realizuje pomoću električnih veličina (napon, struja), odnosno elektronski sklopovi računara obrađuju električne veličine kojima su predstavljeni podaci. Najpogodnije je podatke binarno kodirati, odnosno predstavljati ih pomoću dva definisana stanja elektronskih sklopova, koji se stoga nazivaju **digitalni sklopovi**, a pošto se radi o elektronskim kolima češće se koristi termin **digitalna kola**.

Dva moguća stanja digitalnog kola su najčešće dva nivoa napona U_1 i U_2 . Recimo, $U_1=0V$, a $U_2=5V$. Fizičkim stanjima 0V i 5V odgovaraju dve logičke vrednosti \perp (laž) i T (istina) koja se u digitalnoj elektronici označavaju kao logička nula (0) i logička jedinica (1). (vidi sl. 3.1)



Logičko i fizičko stanje digitalnog kola

Data korespodencija između fizičkih i logičkih stanja odgovara tzv. **pozitivnoj logici**. Moguće je suprotno, nižem naponu U_1 dodeliti logičku 1, a višem naponu U_2 logičku 0 i tada se radi o **negativnoj logici**. Stanja i funkcije digitalnih kola se dakle mogu opisati pomoću logičkih vrednosti i logičkih operacija, pa se zato umesto termina digitalno kolo najčešće koristi termin **logičko kolo**.

Ponašanje logičkih kola može se opisati pomoću **prekidačkih** ili **Bulovih funkcija** koje su predmet izučavanja **Bulove** (ili prekidačke) **algebre**.

Bulova algebra

Za razliku od klasične algebre, promenljiva veličina u Bulovoj algebri može da ima samo dve vrednosti - logička nula (0) i logička jedinica (1):

$$x = 0 \text{ ili } x = 1$$

Tri osnovne operacije, pomoću kojih može da se definiše bilo koja **Bulova funkcija**, su:

1. operacija **logičkog sabiranja** (disjunkcija) ili **ILI (OR) operacija**
2. operacija **logičkog množenja** (konjunkcija) ili **I (AND) operacija**
3. operacija **komplementiranja** tj. **inverzije** (negacija) ili **NE (NOT) operacija**

Pri tome Bulova funkcija predstavlja rezultat izraza koji se sastoji od operandata i operacija nad tim operandima. Naravno, i Bulova funkcija ima rezultat 0 ili 1.

Osnovne operacije se mogu definisati pomoću tablice stanja ili **tablice istinitosti** (Tab. 3.1), iz koje vidimo da su prve dve operacije binarne (dva operandata), a treća je unarna (jedan operand).

Tab. 3.1. Osnovne logičke operacije

operandi x	operacija:			
	Y	ILI (OR) $f = x + y$	I(AND) $f = x \cdot y$	NE(NOT) $f = \bar{x}$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Vidimo da,

1. Logički zbir ima vrednost nula ako i samo ako oba sabirka imaju vrednost nula.
2. Logički proizvod ima vrednost 1 ako i samo ako oba činioca imaju vrednost 1.
3. Komplement ili negacija (inverzija) nule je jedinica, a komplement jedinice je nula.

Ako je u Bulovoj funkciji prisutno više binarnih operacija onda se Bulov izraz izračunava s desna na levo pri čemu se definiše da logičko množenje ima prioritet u odnosu na logičko sabiranje. Prioritet operacija se može promeniti zagradama.

Osnovne teoreme Bulove algebre date su u Tabeli 3.2.

Tab. 3.2. Teorema Bulove algebre

	a)	b)	opis:
1.	$x + 0 = x$	$x \cdot 1 = x$	operacije sa konstantnim vrednostima
2.	$x + 1 = 1$	$x \cdot 0 = 0$	
3.	$x + x = x$	$x \cdot x = x$	zakon idempotentnosti
4.	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$	operacije sa komplementima
5.	$\overline{\bar{x}} = x$		dvostruka negacija
6.	$x + y = y + x$	$x \cdot y = y \cdot x$	komutativnost
7.	$x + (y + z) = (x + y) + z = x + y + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	asocijativnost
8.	$x + y \cdot z = (x + y) \cdot (x + z)$	$x \cdot (y + z) = x \cdot y + x \cdot z$	distributivnost
9.	$x + x \cdot y = x$	$x \cdot (x + y) = x$	zakoni
10.	$x + \bar{x} \cdot y = x + y$	$x \cdot (\bar{x} + y) = x \cdot y$	apsorpcije
11.	$\overline{(x + y)} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$	De Morganova pravila invertovanja

Zakoni De Morgana i zakon distributivnosti se mogu uopštiti:

$$12a) \overline{(x_1 + x_2 + \dots + x_n)} = \bar{x}_1 \cdot \bar{x}_2 \cdot \dots \cdot \bar{x}_n$$

$$12b) \overline{(x_1 \cdot x_2 \cdot \dots \cdot x_n)} = \bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n$$

$$13a) x + y_1 \cdot y_2 \cdot \dots \cdot y_n = (x + y_1)(x + y_2) \cdot \dots \cdot (x + y_n)$$

$$13b) x(y_1 + y_2 + \dots + y_n) = xy_1 + xy_2 + \dots + xy_n$$

Osnovna logička kola

Logičkim operacijama I, ILI i NE (Tab.3.1) odgovaraju **elementarna logička kola** čije su oznake date u tabeli 3.3.

Tab.3.3 Oznake logičkih kola

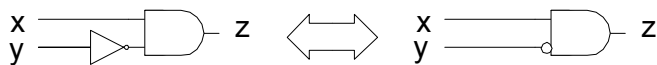
Kolo	Oznaka	Bulova funkcija
NE (NOT)	$x \rightarrow \text{inverter} \rightarrow f$	$f = \bar{x}$
I (AND)	$x, y \rightarrow \text{AND} \rightarrow f$	$f = x \cdot y$
ILI (OR)	$x, y \rightarrow \text{OR} \rightarrow f$	$f = x + y$
NI (NAND)	$x, y \rightarrow \text{AND} \rightarrow \text{inverter} \rightarrow f$	$f = \overline{xy}$
NILI (NOR)	$x, y \rightarrow \text{OR} \rightarrow \text{inverter} \rightarrow f$	$f = \overline{x+y}$
EXILI (EXOR)	$x, y \rightarrow \text{EXOR} \rightarrow f$	$f = x \oplus y$
NEXILI (NEXOR)	$x, y \rightarrow \text{EXOR} \rightarrow \text{inverter} \rightarrow f$	$f = \overline{x \oplus y}$

Elementarnim kolima su pridodata još tri kola: NI, NILI , EXILI i NEXILI (Tab.3.3. i 3.4.), koja zajedno sa njima čine **osnovna kola** u digitalnoj elektronici. Ako u Bulovoj funkciji figurišu više binarnih operacija tj. ta funkcija predstavlja kombinaciju osnovnih logičkih kola sa dva ulaza tada je prioritet operacija sledeći: NI, NILI , EXILI. Naravno, prioritet se može promeniti zagradama.

Tab.3.4. Definicije NI, NILI, EXILI i NEXILI kola

x	y	\overline{xy}	$\overline{x+y}$	$x \oplus y$	$\overline{x \oplus y}$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1

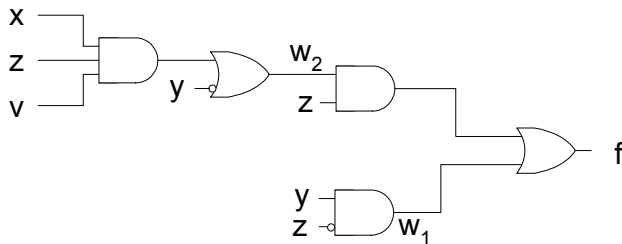
Invertor ulaznog signala u neko kolo označava se **skraćeno** kružićem. Na primer:



Složena logička kola i Bulove funkcije

Složena logička kola su sastavljena iz više osnovnih kola. Svako logičko kolo se može opisati nekom Bulovom funkcijom i obratno, svaka Bulova funkcija se može generisati pomoću nekog logičkog kola. Ilustrovaćemo to sledećim primerima.

PRIMER Formirati Bulovu funkciju koja opisuje sledeće logičko kolo sa četiri ulaza: x,y,z i v



Na slici su uvedene oznake međusignala: w_1 i w_2 . Idući od desnog kraja šeme (izlaz iz kola) prema ulazima, zapažamo:

$$f = w_1 + zw_2$$

$$w_1 = y\bar{z}, w_2 = xzv + \bar{y}$$

Smenom izraza za w_1 i w_2 u izraz za f dobijamo konačno:

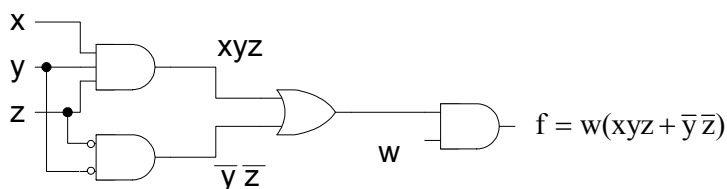
$$f = y\bar{z} + z(\bar{y} + xzv) = y\bar{z} + z\bar{y} + xzv$$

$$f = y\bar{z} + z(\bar{y} + xv)$$

PRIMER Formirati logičko kolo, koje generiše Bulovu funkciju:

$$f(x, y, z, w) = w(xyz + \bar{y}\bar{z})$$

Kolo formiramo postupno-sleva udesno, poštujući pravila o **redosledu izračunavanja složenog Bulovog izraza**.



Sinteza logičkih kola zadate namene

Zadatak sinteze logičkih kola se rešava u četiri etape:

1. Formulisanje tablice istinitosti na osnovu zadate namene kola,
2. Generisanje odgovarajuće Bulove funkcije
3. Uprošćavanje ili **minimizacija** dobijene Bulove funkcije
4. Realizacija minimizovane Bulove funkcije pomoću raspoloživih osnovnih kola

Osnovna kola u računarskim sistemima

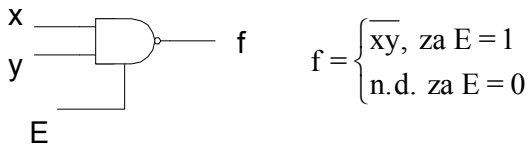
Kola iz kojih je izgrađen računarski hardver mogu se podeliti u dve klase:

1. Kombinaciona
2. Sekvencijalna

Izlazni signal iz **kombinacionih kola zavise od trenutne kombinacije vrednosti ulaznih signala**. Sva kola, koja smo sreli u dosadašnjem izlaganju pripadaju ovom tipu logičkih kola. Za realizaciju vrlo važne funkcije - **memorisanja podataka**, neophodna su i kola koja mogu da proizvoljno dugo zadrže dato stanje tj. vrednost svog izlaza (memorisanje bita 0 ili 1) kao i da uz pomoć ulaznih signala promene stanje. (unos novog sadržaja u 1-bitnu memorijsku lokaciju). Takva kola se nazivaju **sekvencijalna** (ili memorijska) i njihovi **izlazi ne zavise samo od trenutnih vrednosti ulaznih signala, već i od prethodnog stanja**.

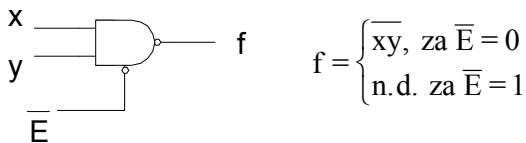
Treće logičko stanje

Logičkim kolima u sklopu hardvera se najčešće dodeljuje tzv. **treće logičko stanje** pri kome su **izlazi iz kola "odvezani"**, tj. **nemaju veze sa ulazima**. Drugim rečima, u trećem logičkom stanju, koje se još i zove i **stanje velike impedanse**, logičko kolo **ne obavlja svoju funkciju**. Za aktiviranje i deaktiviranje kola služi dodatni ulazni signal - **signal dozvole** (enable signal). Na primer NI kolo sa mogućnošću trećeg logičkog stanja, dato je na Sl.4.1a.



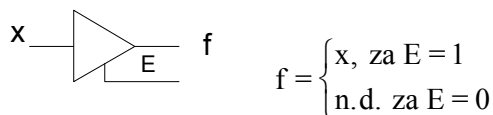
Sl.4.1a NI kolo sa tri logička stanja

Kada signal dozvole, E, ima logičku vrednost 1, kolo je aktivno, a pri E=0 ono prelazi u treće logičko stanje, kad signal f nije definisan. Tačnije, f nema veze sa vrednostima ulaza x i y, već uzima onu vrednost napona koji je trenutno na liniji na koju je kratko vezana f linija - plivajuća vrednost. Aktivna vrednost signala dozvole može da bude nulti napon i tada se on označava sa \overline{E} , što ukazuje na taj uslov (Sl.4.1b).



Sl. 4.1b NI kolo sa tri logička stanja

Za kolo sa trećim logičkim stanjem se koristi termin **trostabilna** ili **trostatička** kola, a ako se kao signal dozvole koristi takti signal i **sinhronizovana** kola. Najjednostavnije trostabilno kolo u računarskoj tehnici je formirač signala dat na Sl.4.2., analogan električnom prekidaču.

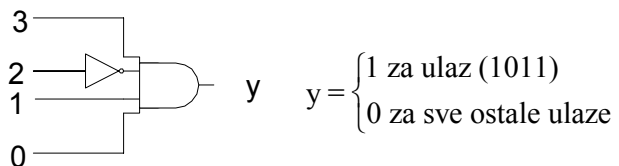


Sl. 4.2. Formirač signala

Dekoder

Dekoder je kombinaciono kolo koje služi za dekodiranje ili prepoznavanje stanja na ulazu. Tako se kao elementi hardvera sreću: adresni dekodeer, dekodeer instrukcija, dekodeer binarnih brojeva itd.

Najjednostavniji dekodeer je **dekodeer sa jednim izlazom**, koji dekodira ili **prepoznaje samo jednu** od mogućih kombinacija ulaznih signala. “Znak prepoznavanja” posmatrane kombinacije, tj. logička vrednost izlaza koja ukazuje na nju, može da bude 1 ili 0. Na primer, dekodeer binarne kombinacije tj. binarnog broja (1011) je I kolo sa 4 ulaza (cifre u binarnom broju), pri čemu je na ulazu sa adresom 2 (ulaz cifre na cifarskom mestu 2^2) stavljen inverter (sl.4.2.)

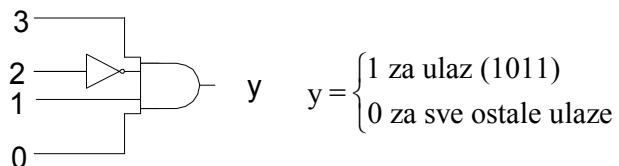


Sl.4.3. Dekoder kombinacije (1011)

Na prisustvo kombinacije (1011) na ulazu ukazuje jedinična vrednost izlaznog signala.

Dekoder 1 od 2^n

Dekoder 1 od 2^n ima n ulaza i 2^n izlaza i dekodira sve moguće (ukupno 2^n) ulazne binarne kombinacije. Svaka od 2^n izlaznih linija jednoznačno odgovara jednoj od mogućih binarnih kombinacija, tj. dobija logičku vrednost 1 samo pri toj kombinaciji, dok istovremeno svi ostali izlazi imaju nulte vrednosti. Na slici 4.3. dat je blok dijagram dekodera 1 od 2^3 sa trećim logičkim stanjem.



Sl. 4.3 Dekoder 1 od 2^3

Brojne oznake ili adrese izlaznih linija predstavljaju dekadne vrednosti odgovarajućih binarnih kombinacija.

PRIMER 4.3. Za vrednosti ulaznih signala u dekodeer 1 od 2^3 na Sl. 4.3.:

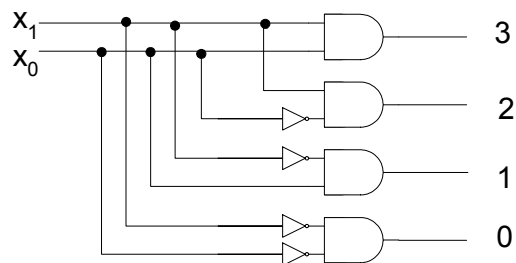
0	1	2	\bar{E}
0	1	1	0

kakve su vrednosti na izlaznim linijama?

U pitanju je linearna kombinacija $110_2 = 6_{10}$ pa će samo na izlaznoj liniji sa adresom 6 biti jedinična vrednost, dok će na ostalim linijama biti 0. Jasno je da dekodeer 1 od 2^n u stvari dekodira tj. prevodi u dekadni oblik n-to cifrene binarne brojeve, pri čemu dekadnu vrednost daje brojna oznaka jedinog aktivnog (vrednost 1) izlaza.

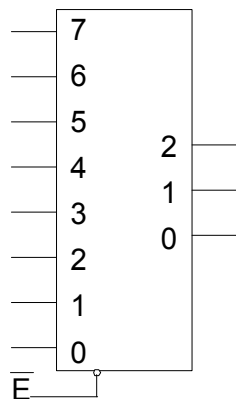
Važna primenu dekodera 1 od 2^n je kao adresnog dekodera kada se na ulaz u dekodeer dovode signali sa adresne sabirnice. Kao rezultat dekodiranja adrese biće aktivna samo jedna od izlaznih linija i ona će aktivirati memorijsku lokaciju ili registar radi unosa ili čitanja podataka.

PRIMER Pomoću I kola i NE kola formirati dekodeer 1 od 2^2 .



Koder

Koder je kombinaciono kolo sa **suprotnom funkcijom od dekodera 1 od 2^n** . Tako, njegovi **izlazi** (ukupno n) daju **binarni kod jedinog aktivnog** (jedinična vrednost) od ukupno 2^n **ulaza** (Sl.4.4).



Sl.4.4 Blok dijagram kodera sa 3 izlaza

PRIMER 4.4. Ako je $\bar{E} = 0$ i jedini aktivni ulaz u koder na Sl.4.4. je ulaz sa adresom 6, koje su vrednosti na izlaznim linijama?

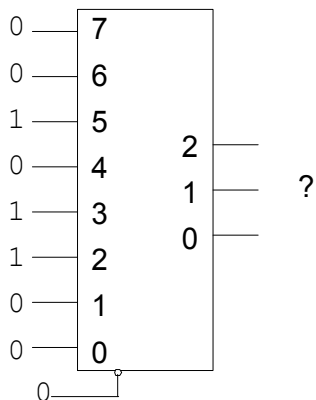
Izlazni signali daju binarni kod broja 6 pa su vrednosti

Izlaz	2	1	0
Vrednost	1	1	0

Koder prioriteta

Kao što smo konstatovali, kod kodera je dovoljno da samo jedna od ulaznih linija bude aktivna. Kod **kodera prioriteta** više ulaznih linija mogu da imaju jediničnu vrednost a **izlazi daju binarni kod najprioritetnijeg od aktivnih ulaza**.

PRIMER 4.5. Ako prioritet ulaza u dati koder prioriteta raste od 0 prema 7 (najprioritetniji je ulaz 7) odrediti vrednosti izlaza.

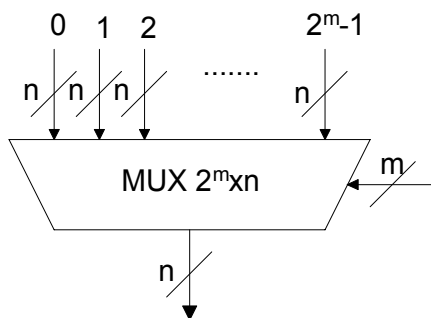


Pošto je najprioritetniji aktivni ulaz onaj sa adresom 5, izlazi daju binarni kod broja 5:

Izlaz	2	1	0
Vrednost	1	0	1

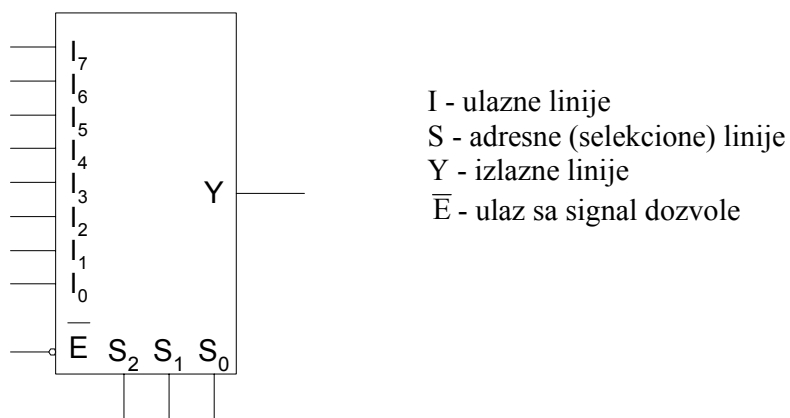
Multiplekser

Multiplekser ili **selektor izvorišta podataka** je kombinaciono kolo sa 2^m ulaznih priključaka, m adresnih ulaznih linija i jednim izlazom. U opštem slučaju, na svaki od ulaznih priključaka (ili portova) kao i na izlazni port može da se veže po n linija i takav multiplekser se zove multiplekser MUX $2^m \times n$ (Sl.4.5)



Sl.4.5. Multiplekser $2^m \times n$

Funkcija multipleksera je da se **pomoću m adresnih ulaza bira koji od 2^m ulaza će biti povezan sa izlazom**. Blok dijagram multipleksera MUX $2^3 \times 1$ sa tri logička stanja dat je na Sl.4.6.



Sl.4.6. MUX $2^3 \times 1$

PRIMER 4.6. Logičke vrednosti ulaza u MUX $2^3 \times 1$ su

Ulaz	0	1	2	3	4	5	6	7
Vrednosti	0	1	1	1	0	0	0	1

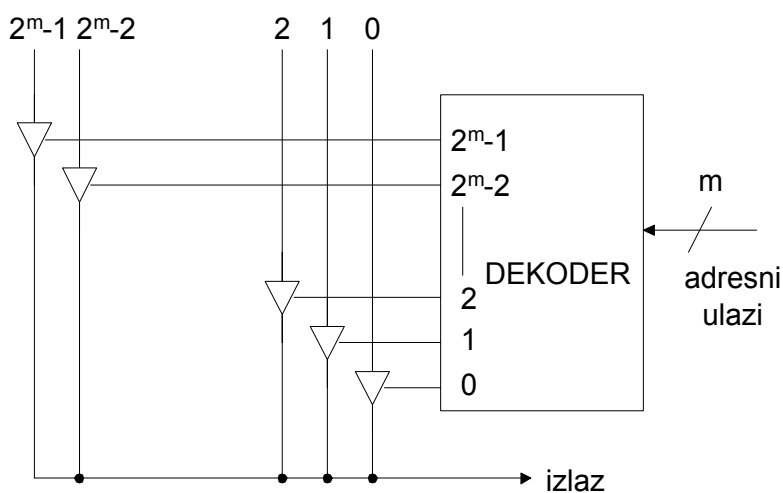
Za svaku od kombinacija vrednosti dozvole i adresnih ulaza odrediti vrednosti izlaza

\bar{E}	S_0	S_1	S_2
0	0	0	0
0	1	1	0
0	0	1	1
0	1	0	0
1	0	1	1

Vrednost (podatak) na onom od ulaza, čija adresa je postavljena na adresnim ulazima, pojaviće se na izlazu, ako je multiplekser aktivan ($\bar{E} = 0$). Tako je rešenje:

\bar{E}	S_0	S_1	S_2	Y
0	0	0	0	0
0	1	1	0	1
0	0	1	1	0
0	1	0	0	1
1	0	1	1	n.d.

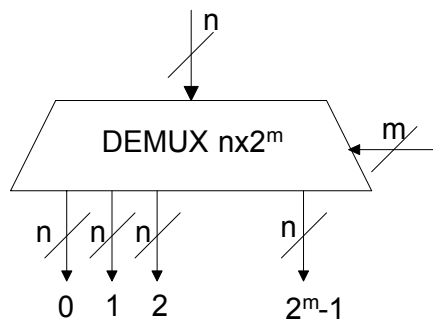
Na slici 4.7. data je realizacija multipleksera MUX $2^m \times 1$ pomoću dekodera 1 od 2^m i 2^m formirača signala:



Sl.4.7. Realizacija multipleksera pomoću dekodera

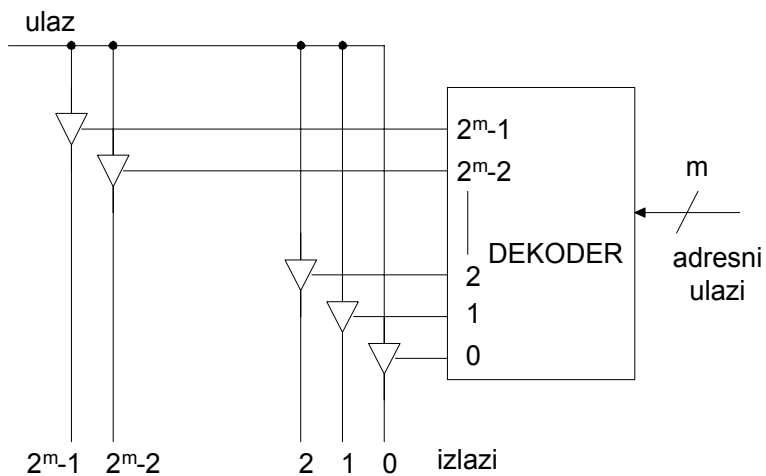
Demultiplekser

Demultiplekser predstavlja **selektor odredišta podataka** jer omogućuje **upućivanje podataka** sa ulaznog porta (linije) na **odabrani izlazni** port uz pomoć adresnih ulaza (Sl.4.8.)



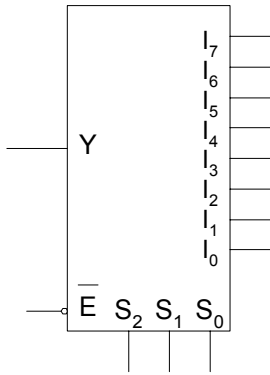
Sl.4.8. Šema demultiplekseara DEMUX $n \times 2^m$

Kao i multiplekser, demultiplekser se može lako realizovati pomoću dekodera 1 od 2^n (Sl.4.9.)



Sl.4.9. Realizacija pomoću dekodera

Na Sl.4.10. Dat je blok dijagram demultiplekseara sa 8 izlaza (DEMUX 1×2^3)

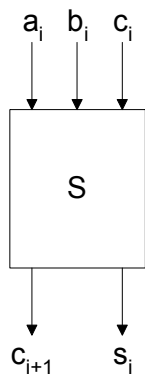


I - izlazne linije
 S - adresne (selekcione) linije
 Y - uazna linija
 \bar{E} - ulaz sa signal dozvole

Sl. 4.10 DEMUX 1×2^3

Binarni sabirač

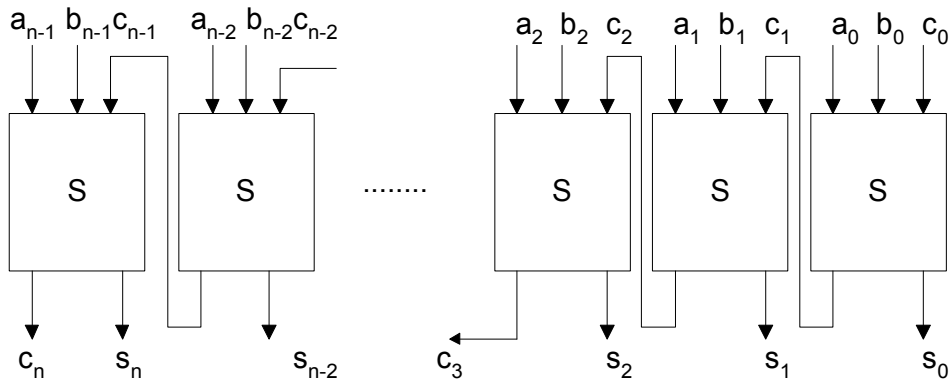
Ovo kombinaciono kolo realizuje sabiranje dva cela binarna broja u ALU.



c_i - prethodni prenos
 c_{i+1} - prenos pri sabiranju cifara a_i i b_i
 s_i - cifra zbir

Sl. 4.11. Binarni sabirač

Binarni sabirač se dobija jednostavno, vezivanjem n potpunih sabirača.



Sl.4.12. Šema binarnog sabirača

Kao što smo već rekli, binarni sabirač je samo jedno od više kola u sklopu ALU. Pored sabirača, tu su i kolo za komplementiranje, kolo za različite logičke operacije sa podacima, a kod složenih ALU i kolo za hardversko izvođenje oduzimanja, množenja i deljenja.

Na linije a_i , $i=0,n-1$, cifre prvog operanda A, koji se nalazi u akumulatoru centralne procesne jedinice (mikroprocesor) dolaze internom sabirnicom. Signali b_i , koji predstavljaju cifre drugog operanda dolaze sabirnicom podataka iz memorijske jedinice mikroracunara, a onda internom sabirnicom mikroprocesora, ili iz nekog registra za privremeno pamćenje podataka u okviru mikroprocesora. Signali s_i linijama interne sabirnice idu u akumulator, pa rezultat S zamenjuje u akumulatoru prethodno smešten prvi operand. Preostaje da objasnimo odakle dolazi signal početnog prenosa c_0 i gde ide konačan prenos c_n . Prenos (carry) kao vrlo važna informacija se čuva u mikroprocesoru u posebnom registru - registar uslova ili status registar. Tako konačan prenos c_n ide u registar uslova (kao bit ili zastavica C). Kakav smisao kod sabiranja ima početni prenos c_0 ? To će biti jasno, ako se podsetimo da se u mikroracunaru celi brojevi registruju kao dvobajtni (16-cifreni). Tako se u 8 bitnom mikroprocesoru sabiranje dva broja vrši u dva koraka: u prvom se, bez početnog prenosa, sabiraju niži bajtovi, a u drugom koraku se sabiraju viši bajtovi gde je početni prenos, c_0 jednak konačnom prenosu c_n prethodnog sabiranja iz registra stanja.

Bistabilna kola (flip-flop)

Bistabilna kola služe za **memorisanje bita**. To su **sekvencijalna kola** koja mogu da budu u dva stabilna stanja, 0 ili 1, otuda naziv **bistabilna kola** ili **bistabili**. Stabilno stanje (memorisani bit) se može održavati proizvoljno dugo, a može se delovanjem ulaznih signala promeniti (unos novog sadržaja).

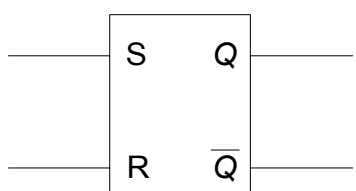
Bistabil ima **bar jednu izlaznu liniju** na kojoj se dobija njegovo stanje (memorisani bit). Ako je na izlaznoj liniji **jedinica** kažemo da je bistabil **setovan**, a ako je **nula** kažemo da je **resetovan**.

Bistabil ima bar jednu ulaznu liniju i vrednost signala na ulazu, zajedno sa njegovim satnjem (signal na izlaznoj liniji), određuje njegovo novo stanje, tj. novu vrednost izlaza.

Bistabili se sreću i pod nazivom flip-flopovi.

R-S flip-flop

Blok dijagram R-S flip-flopa je dat na Sl.4.13. a njegovu funkciju objašnjava tablica istinitosti (tab.4.1.)



Q - izlaz na kome se dobija stanje
 \bar{Q} - drugi izlaz, čija vrednost je uvek invertovano stanje
 S - ulaz za setovanje (unos jedinice)
 R - ulaz za resetovanje (unos nule)

Sl. 4.13. Blok dijagram R-S flip flopa

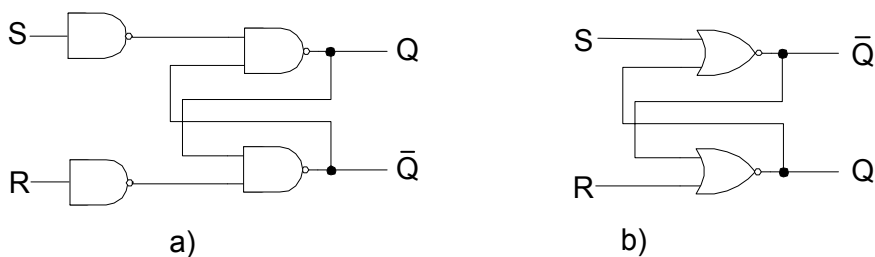
Tab. 4.1. Tablica istinitosti R-S flip flopa

R	S	Q
0	0	Q^*
0	1	1
1	0	0
1	1	n.d.

Q^* - postojeće stanje

n.d. - ova kombinacija nije dozvoljena

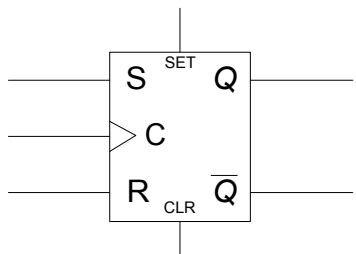
Na Sl.4.14. data je uprošćena realizacija R-S flip flopa pomoću NI odnosno NILI kola



Sl.4.14. Realizacija R-S flip-flopa pomoću a) NI i b) NILI kola

Pretpostavimo da su trenutne vrednosti signala Q i \bar{Q} u šemi na Sl.4.14.: $Q = 0$, $\bar{Q} = 1$. Sve dok su R i S jednaki nuli stanje Q se ne menja (Tab 4.1.). Zaista, na ulazu u gornje NI kolo su dve jedinice što daju što daje $Q = 0$, dok u donje NI kolo ulaze nula (Q) i jedinica (\bar{R}) što daje $\bar{Q} = 1$. Ako želimo da promenimo stanje tj. da u bistabil unesemo bit 1 (setovanje) treba na ulazu primeniti: $S = 1$, $R = 0$ (Tab.4.1). U gornje NI kolo sada ulaze jedna jedinica (\bar{Q}) i jedna nula (\bar{S}) što daje novu vrednost za Q , $Q = 1$. Ona zajedno sa $\bar{R} = 1$ u donjem NI kolu daje $\bar{Q} = 0$. Ta nova vrednost za \bar{Q} ne menja izlaz iz gornjeg NI kola (dve nule daju opet 1). Slično razmatranje važi i za R-S flip-flop realizovan sa 2 NILI kola.

U praksi se češće koriste sinhronizovani R-S bistabili za dodatnim priključkom za signal dozvole C i postavljanje Q na 1 (SET) ili na 0 (CLR)(Sl.4.15.)



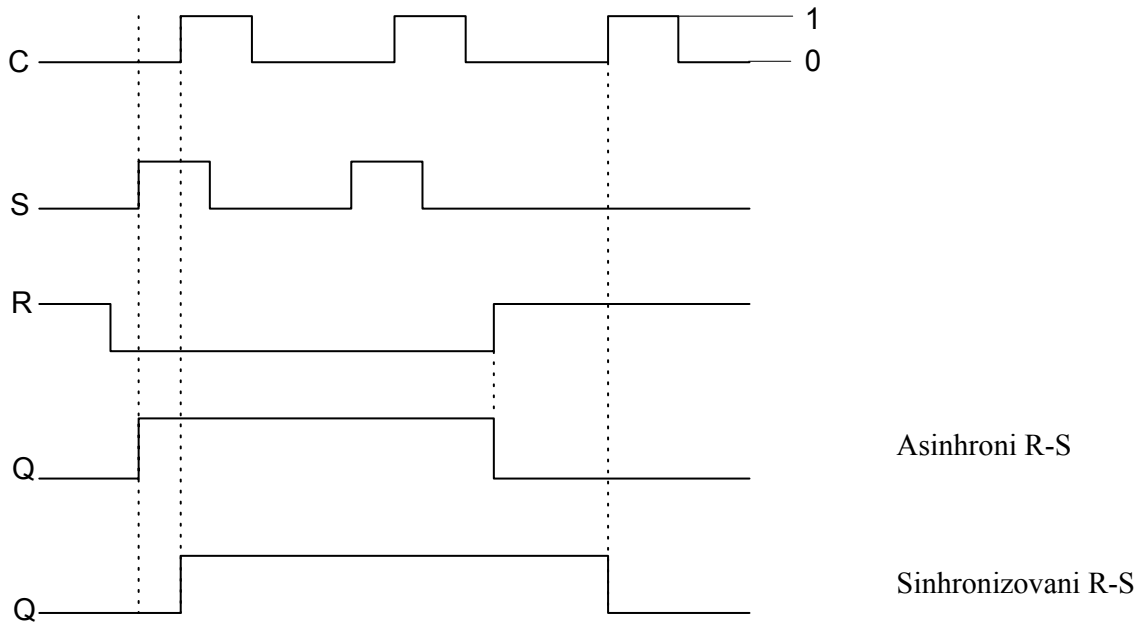
C - ulaz za signal dozvole, po pravilu taktni signal (clock signal)
 SET - postavljanje Q na 1 bez obzira na ostale signale (setovanje)
 CLR - postavljanje Q na 0 bez obzira na ostale signale (resetovanje)

Sl.4.15. Sinhronizovani R-S flip-flop

C	R	S	Q
1	0	0	Q*
1	0	1	1
1	1	0	0
0	x	x	Q*

x - bilo koja vrednost (0 ili 1)
 Q* - postojeće stanje

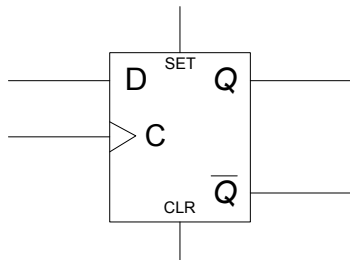
Razlika u funkciji asinhronog i sinhronizovanog R-S bistabila jasna je sa vremenskog dijagrama na slici 4.16.



Sl.4.16. Vremenski dijagram asinhronog i sinhronizovanog R-S flip-flopa

D flip-flop

D flip-flop (Sl.4.17.) je sinhronizovan bistabil sa jednim ulazom D. Vrednost sa ulaza prelazi na izlaz Q, sinhronizovano sa taktim signalom (Tab.4.2. i Sl.4.18.)

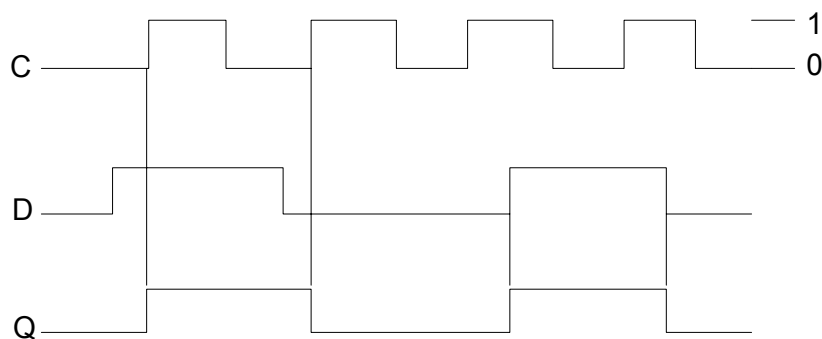


Sl.4.17. D flip-flop

Tab.4.2 Tablica istinitosti D flip-flopa

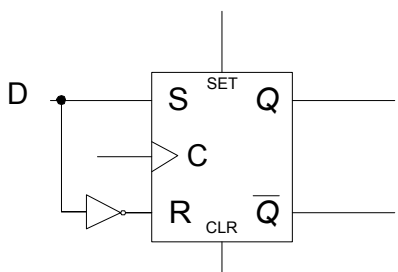
C	D	Q
1	0	0
1	1	1
0	x	Q*

x - bilo koja vrednost (0 ili 1)
 Q* - postojeće stanje



Sl. 4.18. Vremenski dijagram D - flip flopa

D flip flop se može jednostavno formirati od R-S flip flopa (Sl.4.19)



Sl.4.19. Realizacija D flip flopa pomoću R-S flip flopa

Registri

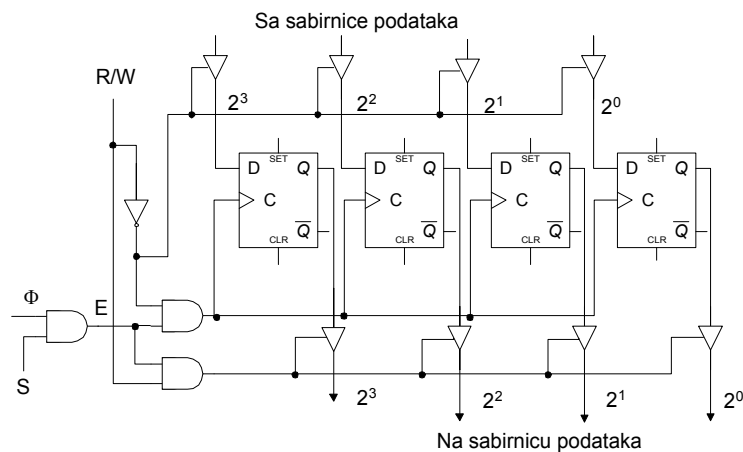
Osnovna namena registra je **privremeno pamćenje** nekog višebitnog podatka (kod 8-bitnog procesora, njegov kapacitet je 8 bitova). U tom svojstvu, sem u memoriji (memorijski registri) oni se sreću u **međusklopovima**, preko kojih su periferne ulazno-izlazne jedinice vezane za sabirnicu mikroračunara, i tu imaju funkciju **međumemorisanja** ili baferovanja (baferni registri) podataka na putu od ulazne jedinice do mikroprocesora ili od mikroprocesora do ulazno-izlazne jedinice. Registri koji imaju samo opisani način **čuvanja** (registrovanja) podataka zovu se **statički** ili **stacionarni registri**.

Složeniji registri od statičkih su oni koji omogućavaju različite **manipulacije** sa registrovanim podatkom i to su:

- pomerački (šift) registri
- ciklični ili kružni registri
- brojački registri ili kaunteri (counter)

Statički registri

S obzirom da čuvanje 1-bita omogućuju bistabili, jasno je da se n-bitni registar dobija **paralelnim vezivanjem n bistabila**. Šema jednog 4-bitnog registra data je na slici 4.20.



$$R/W = \begin{cases} 1 & \text{za čitanje (read)} \\ 0 & \text{za pisanje (write)} \end{cases}$$

Φ - takti signal iz generatora taktnog signala
 S - signal izbora registra, iz adresnog dekodera

Sl.4.20. Šema 4-bitnog statičkog registra

Da bi se u neki registar uneo (write) podatak sa sabirnice podataka ili iz njega pročitao (read) podatak, tj. postavio na sabirnicu podataka neophodno je da budu ispunjena 2 uslova.

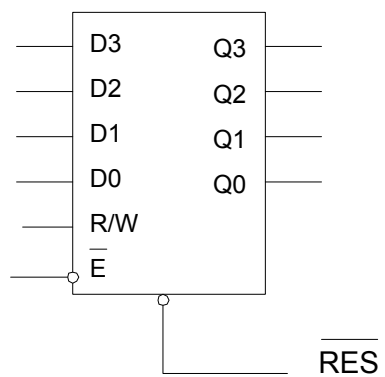
- odabran je posmatrani registar, tj. njegova adresa je postavljena (od strane mikroprocesora) na adresnu sabirnicu,
- vrednost taktnog signala Φ je jednak 1

Ako je mikroprocesor odabrao dati registar, signal S koji ide iz adresnog dekodera imaće vrednost 1 (vidi poglavlje 4.2.1), inače njegova vrednost je 0. Ako su ispunjena oba opisana uslova, tj. $E = \Phi \cdot S = 1$ (I kolo na levom kraju šeme), zavisno od vrednosti R/W signala biće realizovano čitanje ili upisivanje.

Ako R/W signal ima vrednost 1, izlaz iz I kola u koji se uvode R/W i $E = \Phi \cdot S$ signal biće aktivan i otvoriće formirače signala pa će se bitovi (Q signali) iz pojedinih D bistabila naći na sabirnici podataka (operacija čitanja).

Ako je $R/W = 0$ (upisivanje) izlaz iz I kola u koji se uvode $\overline{R/W}$ i E signal biće aktivan i on će aktivirati D bistabile, pa će u njih biti uneti bitovi sa sabirnice podataka.

Na slici 4.21. dat je blok dijagram 4-bitnog statičkog registra, kome je pridodat i priključak za resetovanje ili brisanje registra (postavljanje svih izlaza Q_i na nulu).



Sl.4.21. Blok dijagram 4-bitnog statičkog registra sa mogućnošću resetovanja

PITANJA

Dekoder, Koder

Multiplexer, Demultiplexer

Potpuni sabirač, Binarni sabirač

Flip-Flopovi, RS, D

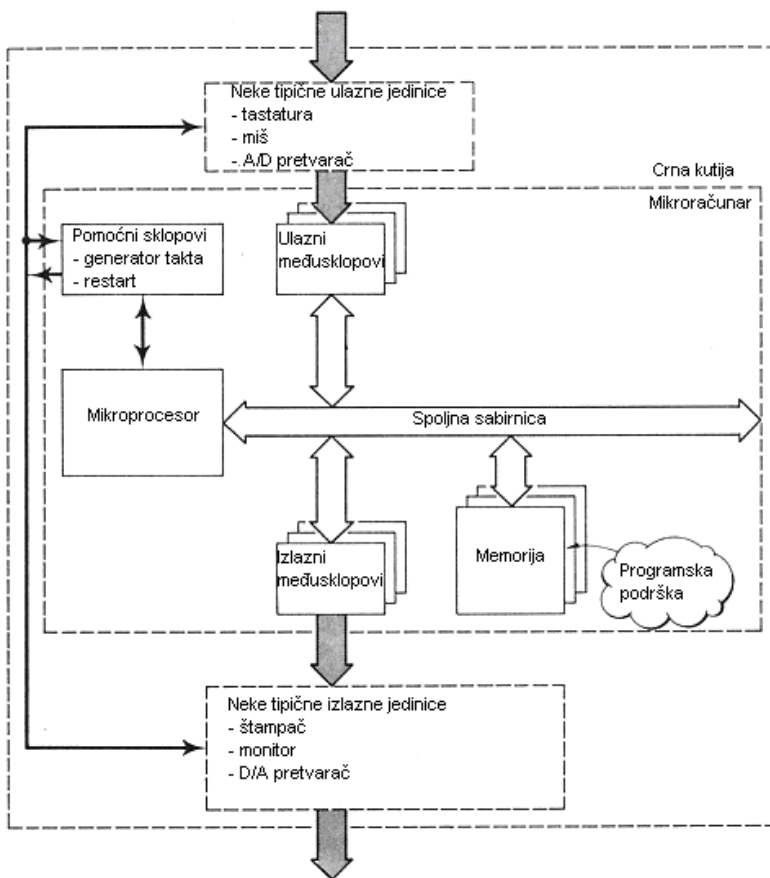
Registri

MIKRORAČUNAR

Mikroračunar je sastavljen od četiri osnovna bloka (Sl.1) - to zovemo hardver:

- mikroprocesora
- memorije
- ulaznog međusklopa
- izlaznog međusklopa

Programska podrška (to zovemo softver) je “vezivna materija” koja ta četiri bloka osmišljava i drži ih zajedno. *Mikroprocesor* sjedinjuje u sebi sposobnost računskog dela i dela koji je sposoban da na temelju dobijenih rezultata donosi odluke-izabere jedan od alternativnih smerova daljnjeg delovanje ili akcije. U *memoriji* se smeštaju podaci u binarnom obliku, međurezultati i rezultati. U memoriji su takođe smešteni programi koji određuju mikroprocesoru koje operacije mora izvršiti. Preko *ulaznog međusklopa* ostvaruje se put za prenos binarnih podataka (u paralelnom ili serijskom obliku) od ulaznih jedinica prema mikroprocesoru ili memoriji. *Izlazni međusklop* omogućava prenos binarnih podataka od mikroprocesora prema izlaznim jedinicama. Sastavni blokovi (mikroprocesor, ulazno-izlazni međusklopovi, memorija) su preko spoljne sabirnice povezani i čine mikroračunar (Sl.1.). Spoljna sabirnica je skup linija preko kojih se saobraća između sastavnih delova mikroračunara.

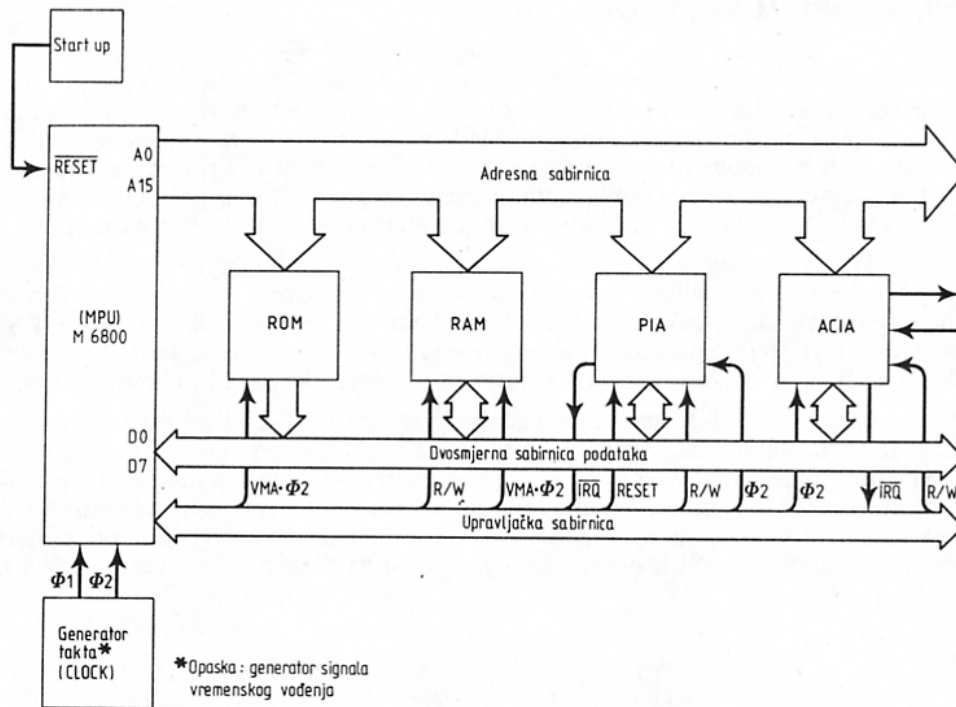


Sl.1. - Sklop mikroračunara kao crna kutija

Sl.2. prikazuje strukturu mikroračunara zasnovanog na porodici Motorola M6800,gde su:

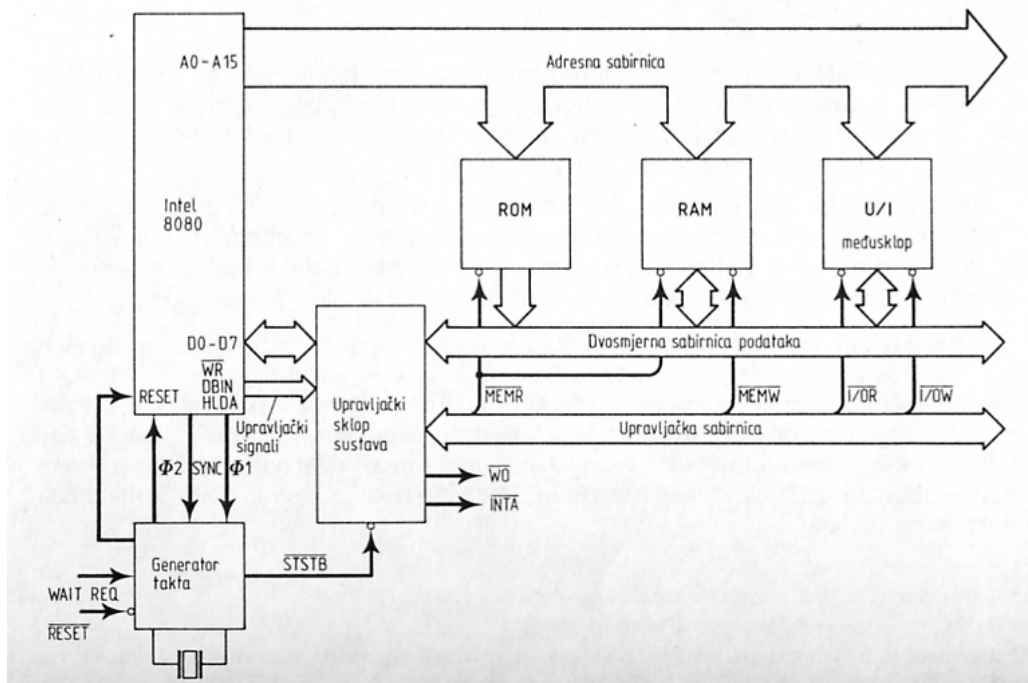
- MPU: mikroprocesor,
- RAM, ROM: memorija sa direktnim pristupom, ispisna memorija,
- START UP, CLOCK: pomoćni sklopovi,
- PIA: paralelni U/I međusklop,
- ACIA; serijski međusklop.

Spoljna sabirnica se sastoji od upravljačke (CONTROL BUS) i adresne (ADDRESS BUS) sabirnice i sabirnice podataka (DATA BUS).



Sl.2. - Struktura mikroračunara zasnovanog na mikroprocesorskoj porodici M6800

Sl.3 prikazuje građu mikroračunara na osnovi mikroprocesora Intel 8080, a sastoji se od memorije RAM, ROM, U/I međusklopa i specijalne jedinice nazvane upravljački sklop (system controller), generatora takta, kao pomoćnog sklopa. Spoljna sabirnica se sastoji od adresne sabirnice, sabirnice podataka i upravljačke sabirnice.



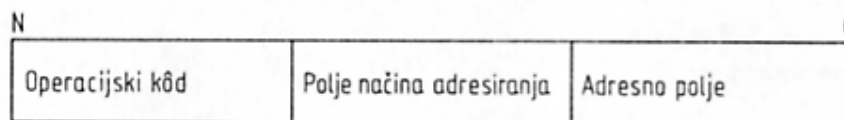
SI.3 - Struktura mikroračunara na bazi mikroprocesora Intel 8080

MODEL MIKROPROCESORA

Mikroprocesor je osnovni sastavni blok mikroračunara. On je centralno procesorska jedinica realizovana u tehnologiji LSI. Sastoji se od sklopova koji dekodiraju instrukcije pribavljene iz memorije i u skladu s tim generišu sekvence upravljačkih signala (koji određuju niz prenosa preko interne sabirnice, prenose kroz aritmetičko-logičku jedinicu, pobuđivanje internih sklopova i sl.), potrebnih za izvođenje instrukcije. To je jedan od osnovnih zadataka upravljačke jedinice.

Mikroprocesorski čip sadrži aritmetičko-logičku jedinicu (ALU). U njoj se izvode aritmetičke i logičke operacije na binarnim podacima. Registri za privremeno skladištenje i rukovanje podacima također su sastavni deo procesora LSI. Budući da većina standardnih mikroprocesora ima reč dužine osam bita (bajt), dat je opis njegovog rada na jednostavnom modelu 8-bitnog mikroprocesora.

Dužina reči od 8 bita onemogućava da se instrukcijska reč razdeli na klasičan način kao kod računara: na polje operacijskog koda, na polje načina adresiranja i na adresno polje (SI.4). Samo polje operacijskog koda zauzelo bi veći deo 8-bitne reči. Na primer, 8-bitni mikroprocesor M-6800 ima osnovni skup od 72 instrukcije, što zahteva operacijski kod od 7 bita ($2^7 > 72 > 2^6$), ako privremeno zanemarimo 197 različitih operacijskih kodova u zavisnosti od načina adresiranja.



SI.4 - Organizacija instrukcijske reči

Adresno polje bilo bi svedeno, u najboljem slučaju na jedan bit. To znači da bi bilo moguće direktno adresirati samo dve memorijske lokacije- što je očito premalo. Problem dužine reči mora biti kod mikroprocesora rešen pribavljanjem više 8-bitnih reči za konstrukciju jedne instrukcijske reči.

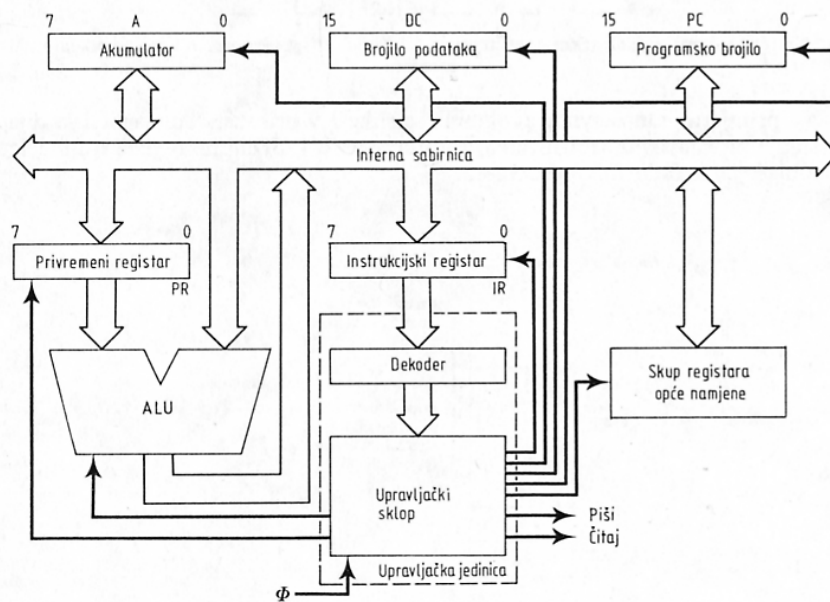
Zbog čega reč mikroprocesora nije duža (npr. 24 ili 32 bita)? Većina razloga je u tehnološkim ograničenjima, na primer u broju izvoda na integrisanom kućištu DIP (sa izvodima u dve linije), problemu gustoće integracije komponenti, postojećoj opremi za testiranje i ispitivanje čipova u fazi proizvodnje i sl..

Jednostavni model na kojem će biti opisan princip rada prikazan je na *Sl.5*. Model mikroprocesora ima *akumulator A* koji se upotrebljava za privremeno skladištenje jednog od operanada. Akumulator učestvuje pri izvršavanju aritmetičkih i logičkih operacija na podacima, te ima i središnju ulogu u prenosu podataka unutar mikroračunara ili sklopa mikroračunara. *Programsko brojilo (programski brojač)* - registar PC - sadrži adresu sledećeg bajta koji će biti pribavljen u narednom ciklusu.

Operacijski kod instrukcije upisuje se u *instrukcijski registar* - registar IR. U 16-bitnom *brojilu podataka (registar podataka)* registru DC, sadržana je adresa memorijske lokacije u kojoj se nalazi operand. Izvođenje svake instrukcije deli se na:

- fazu pribavljanja instrukcije -PRIBAVI (*fetch*),
- fazu izvršavanje instrukcije -IZVRŠI (*execute*).

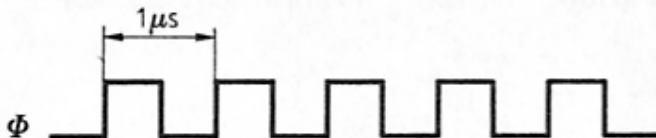
Mikroprocesor za vreme faze PRIBAVI postavlja sadržaj programskog brojila preko *interne sabirnice* na spoljnu *adresnu sabirnicu*. Ujedno šalje i odgovarajuće upravljačke signale (signal ČITAJ) na spoljnu upravljačku sabirnicu (u našem slučaju pojednostavljeni model imće samo dva upravljačka signala: ČITAJ i PIŠI). Memorijski sklop dekodira postavljenu adresu (prisutnu na adresnoj sabirnici) u cilju pristupa do odgovarajuće memorijske reči. Za nekoliko stotina *ns* (npr. 500 *ns*) sadržaj specificirane memorijske lokacije pojaviće se na spoljnoj sabirnici podataka. Taj se sadržaj skladišti u instrukcijskom registru IR, i to je operacijski kod instrukcije. Za vreme faze PRIBAVI mikroprocesor upotrebljava svoju internu logiku i povećava sadržaj programskog brojila.



Sl.5 - Pojednostavljeni model mikroprocesora

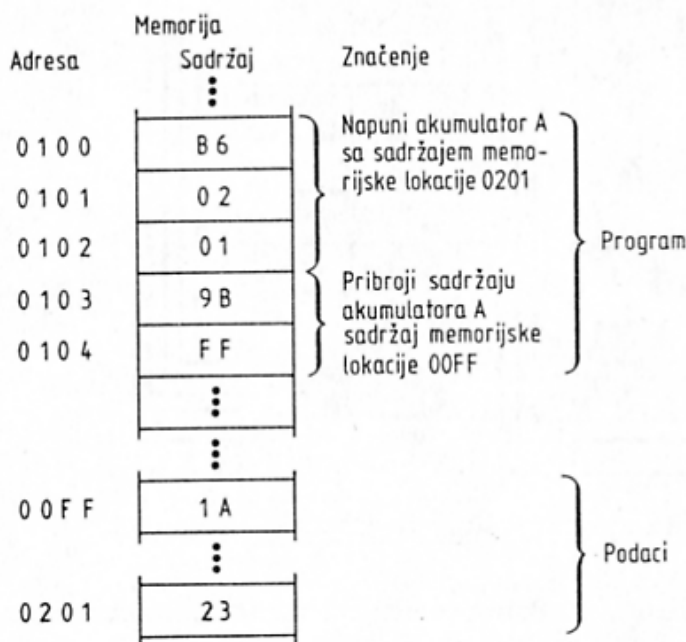
U fazi IZVRŠI upravljačka jedinica, u skladu s operacijskim kodom koji je skladišten u instrukcijskom registru, stvara niz upravljačkih signala. Rezultat tog niza signala su odgovarajući prenosi podataka, odnosno operacije (npr. aktiviranje pojedinih sklopova unutar aritmetičko-logičke jedinice, izvršavanje (izvođenja) zadate instrukcije.)

Operacije unutar mikroprocesora (često se nazivaju mikrooperacijama) sinhronizovane su generatorom takta. Perioda generatora takta može biti, u zavisnosti od tipa mikroprocesora, od 100 ns do nekoliko μ s. Signali generatora takta mogu se sastojati od jednog ili više signala (to je onda višefazni generator takta, npr. mikroprocesor M6800 ima signale $\phi 1$ i $\phi 2$). Za pojednostavnjeni model izabran je jednofazni generator takta (Sl.6). ϕ je obično oznaka za signal generatora takta.



Sl.6 - Jednofazni signal takta (vremenskog vođenja) Φ pojednostavljenog modela mikroprocesora

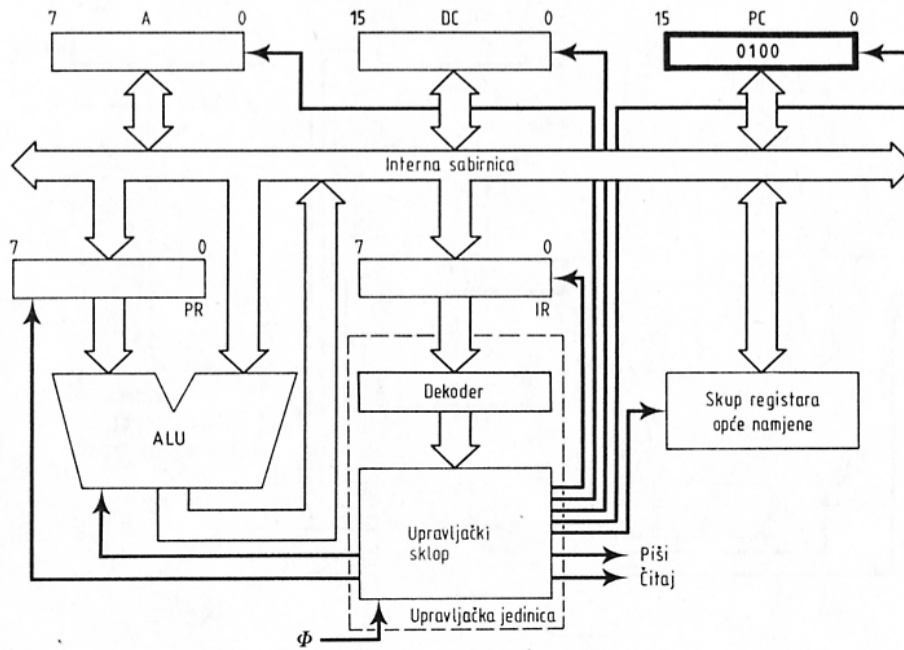
Na primeru jednostavnog programa sa Sl.7 prikazani su vremenski dijagram stanja na spoljnim sabirnicama i promene u sadržajima registara modela u cilju objašnjenja rada mikroprocesora.



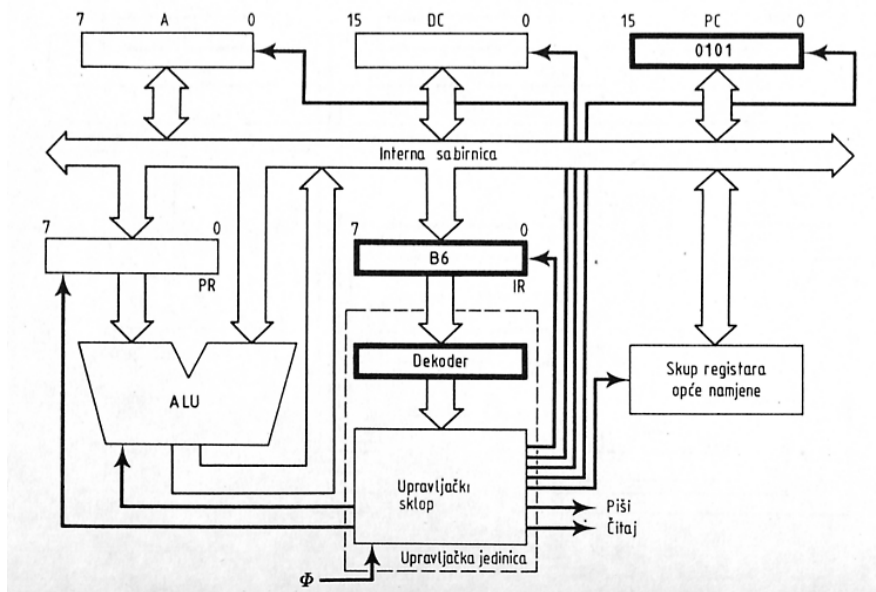
Sl.7 - Primer programa

Napomena: Adrese i sadržaji memorijski lokacija na Sl.7 prikazani u heksadekadnom sistemu.

Početni uslov (sadržaj registara) prikazuje Sl.8. Na slici su označeni samo oni registri koji učestvuju u izvođenju programa. U programskom brojilu postavljena je adresa prve instrukcije.



SI.8 - Početni uslovi

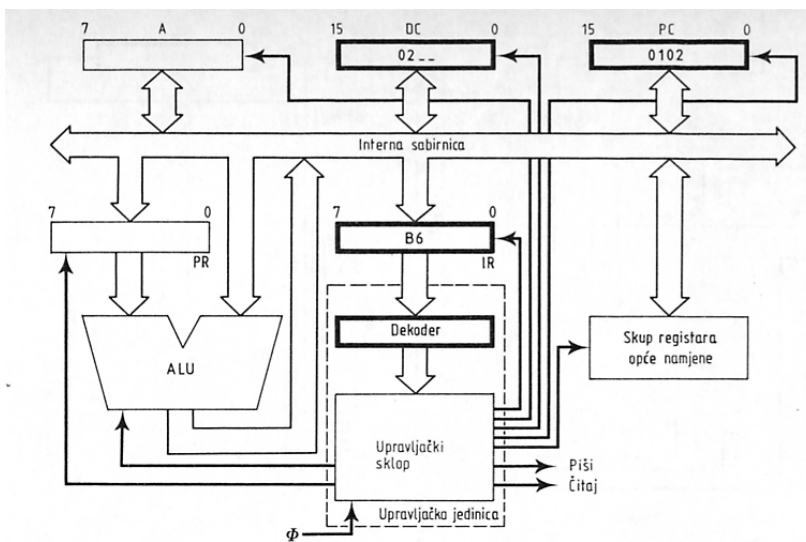


SI.9 - Stanje nakon faze PRIBAVI: Prvi ciklus prve instrukcije

SI.9 prikazuje stanje nakon faze PRIBAVI: u instrukcijskom registru skladišten je operacijski kod instrukcije, sadržaj programskog brojila povećan je za 1. Sadržaj instrukcijskog registra 10110110 (B6-heksadekadno) dekodiran je kao: napuni akumulator A sadržajem memorijske lokacije koje je adresa sadržana u sledeća dva bajta. Mikroprocesor pribavlja sledeći bajt postavljanjem sadržaja programskog brojila na adresnu sabirnicu i generiranjem upravljačkog signala ČITAJ. Pribavljeni bajt se smeštava u brojilo podataka.

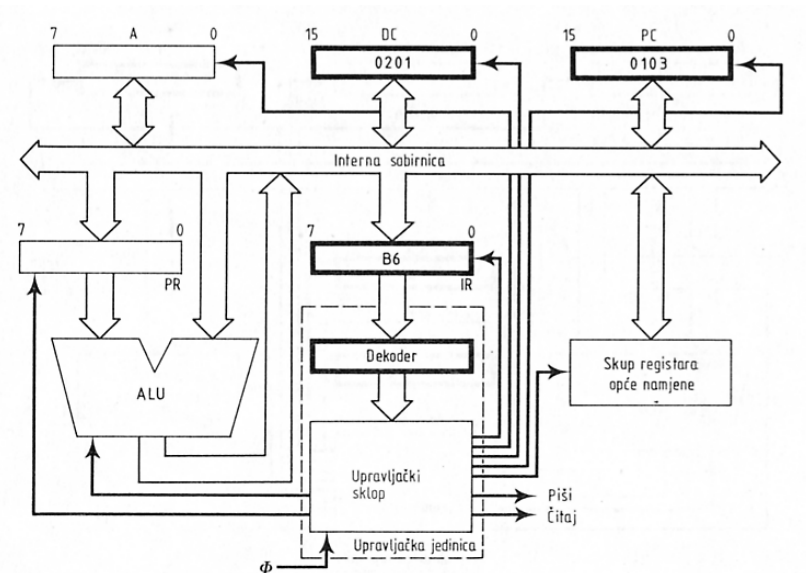
SI.10 prikazuje stanje nakon pribavljanja značajnijeg bajta adrese operanda (02..); programsko je brojilo uvećano za 1. Mikroprocesor pribavlja treći bajt instrukcijske reči-manje značajni bajt adrese

operanda (.01) i smešta ga u brojilo podataka; programsko brojilo uvećava se za 1 i pokazuje na sledeću instrukciju 9B (Sl.7).



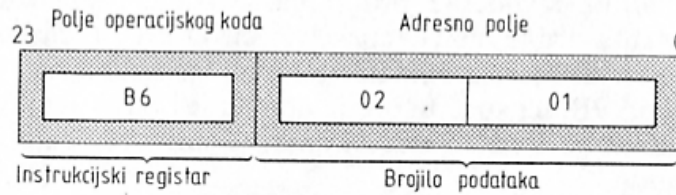
Sl.10. Stanje nakon pribavljanja značajnijeg bajta adrese operanda: Drugi ciklus prve instrukcije

Sl.11 prikazuje stanje nakon pribavljanja manje značajnog bajta adrese operanda.



Sl.11. Stanje nakon pribavljanja manje značajnog bajta adrese operanda: Treći ciklus prve instrukcije

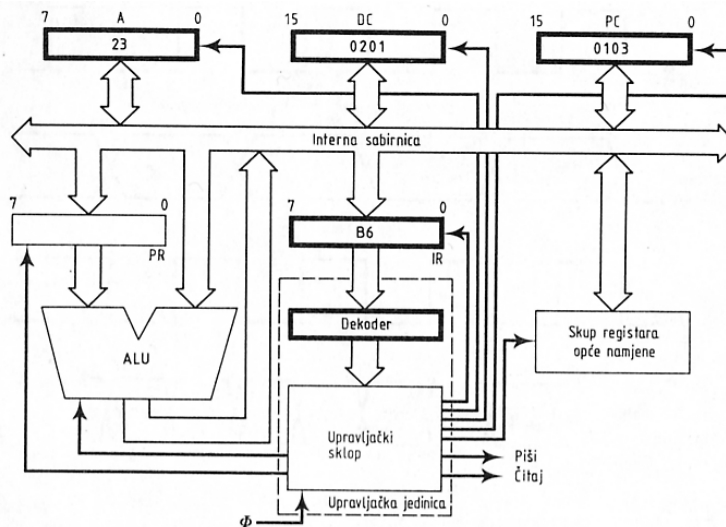
Ako se na trenutak pretpostavi da instrukcijski registar IR (0-7) i brojilo podataka DC (0-15) čine jedan 24-bitni registar, može se instrukcijski registar smatrati poljem operacijskog koda, a brojilo podataka adresnim poljem (Sl.12).



Sl.12 - Instrukcijska reč sastavljena iz tri bajta

Postupku pribavljanja kompletne instrukcijske reči mikroprocesora su bila tri ciklusa (3x1 bajt), dok bi računar sa dužinom reči od 24 bita taj isti postupak obavio u jednom ciklusu. Mikroprocesor pribavlja operand postavljanjem sadržaja brojila podataka na adresnu sabirnicu i generisanjem upravljačkog signala ČITAJ.

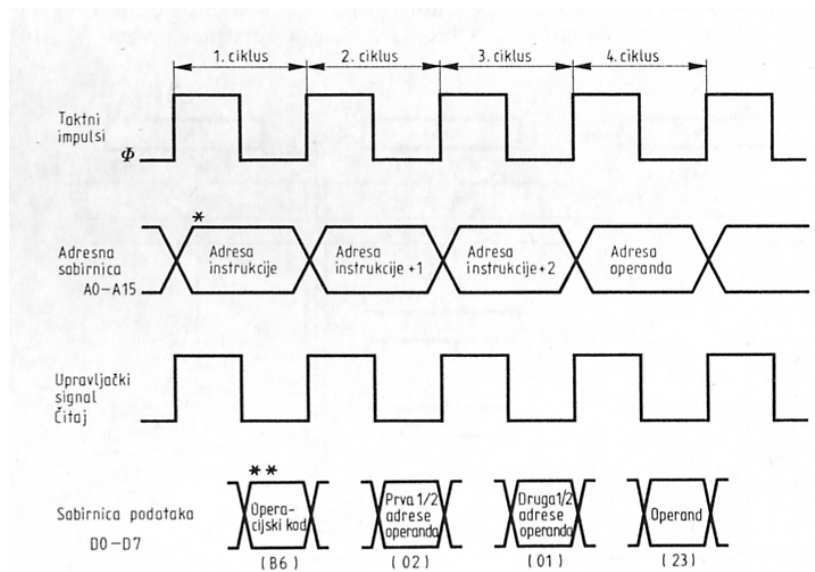
Sl.13 prikazuje konačni rezultat izvođenja prve instrukcije: akumulator A napunjen je sadržajem memorijske lokacije 0201. Sadržaj programskog brojila nije povećan, jer je pribavljen operand a ne instrukcija - mikroprocesor je bio u fazi IZVRŠI.



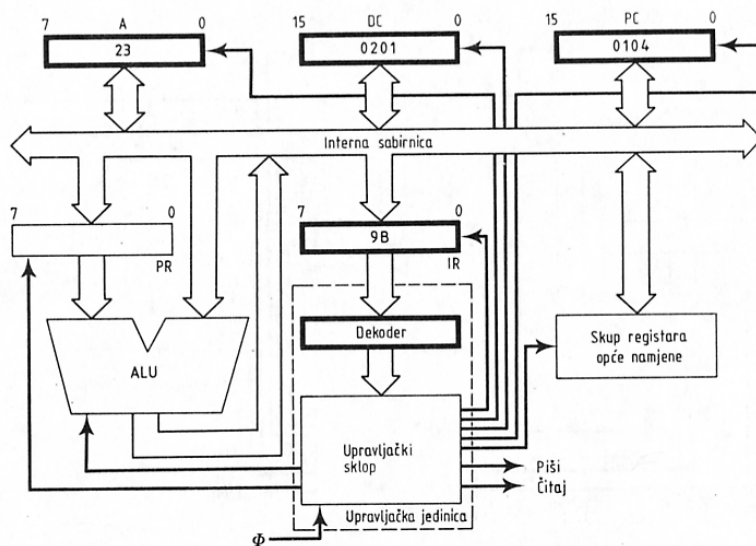
Sl.14 Stanje nakon izvođenja prve instrukcije: Četvrti ciklus prve instrukcije

Sl.15. prikazuje pojednostavnjeni vremenski dijagram izvođenja prve instrukcije. Instrukcija se izvodi u četiri ciklusa (periode) generatora takta ϕ .

Nastavlja se dalje izvođenje programa. Sl.15 prikazuje stanje nakon pribavljanja prvog bajta druge insrukcijske reči. U instrukcijskom registru smešten je operacijski kod druge instrukcije. Sadržaj programskog brojila povećan je za 1. Instrukcijski kod 9B, uz specificiranje operacije (pribrajanja operanda sadržaju akumulatora A), određuje i način adresiranja - sledeći bajt je manje značajan bajt adrese operanda.

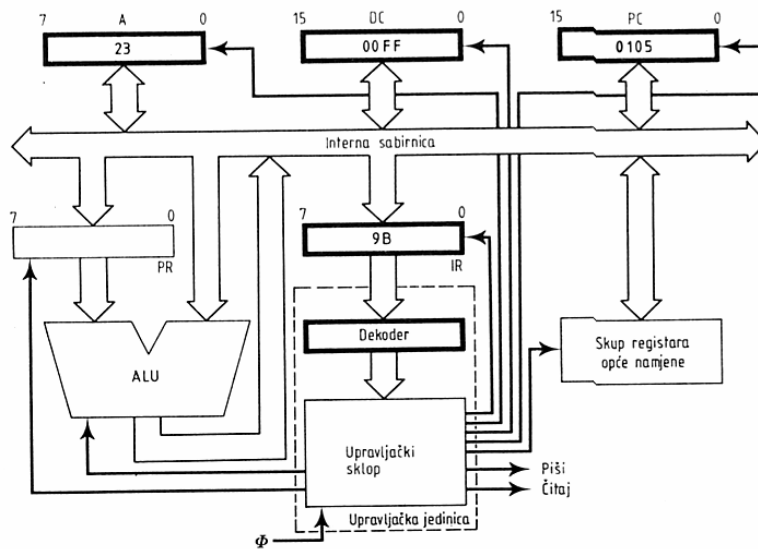


SI.15 - Pojednostavljeni vremenski dijagram izvođenja prve instrukcije

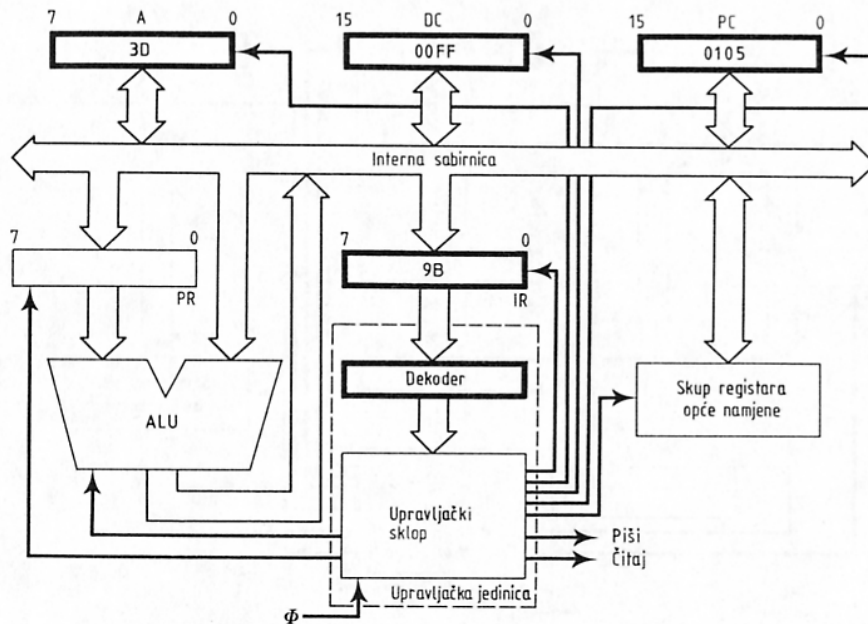


SI.16 - Stanje nakon pribavljanja prvog bajta druge instrukcije: Prvi ciklus druge instrukcije

SI.17 prikazuje stanje nakon pribavljanja drugog bajta instrukcijske reči. Programsko brojiilo povećano je za 1, a u brojiilo podataka smeštava se adresa operanda. SI.18 prikazuje rezultat konačnog izvođenja instrukcije-sadržaju akumulatora A pribrojen je sadržaj sa memorijske lokacije 00FF (1A). Programsko brojiilo nije povećano za 1 - mikroprocesor je bio u fazi IZVRŠI.



SI. 17 - Stanje nakon pribavljanja drugog bajta instrukcijske reči: Drugi ciklus druge instrukcije



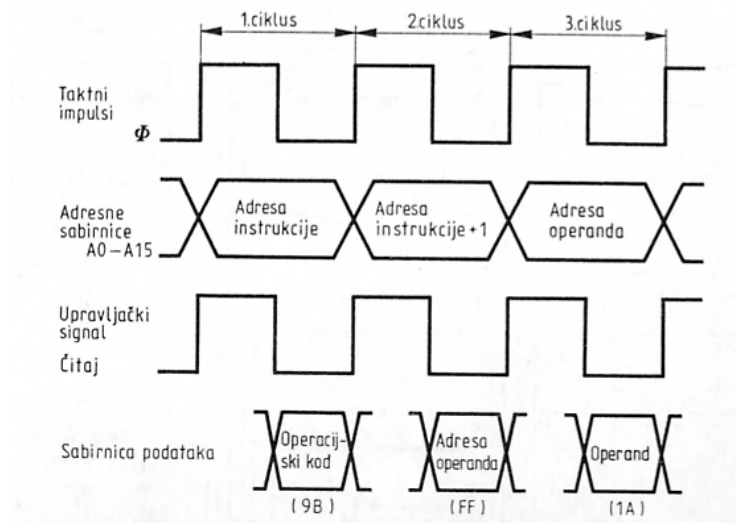
SI.18 - Prikaz konačnog izvođenja instrukcije

SI.19 prikazuje vremenski dijagram izvođenja druge instrukcije. Tabela 1 prikazuje stanje na sabirnici podataka i adresnoj sabirnici za model mikroprocesora prilikom izvođenja zadatog programa sa SI. 7. U mnemoničkom kodu program bi imao (za mikroprocesor M6800) sledeći oblik:

LDAA \$0201

ADDA \$FF

(Napomena: \$ je oznaka za heksadekadni broj)



SL.19 - Vremenski dijagram izvođenja druge instrukcije

Tabela 1

Stanje na sabirnici podataka i adresnoj sabirnici

Instrukcija	Broj ciklusa	Ciklus	Signal (Čitaj)	Adresne sabirnice	Sabirnice podataka
B6	4	1	1	Adresa instrukcije	Operacijski kod
		2	1	Adresa instrukcije +1	Adresa značajnijeg bajta operanda
		3	1	Adresa instrukcije +2	Adresa manje značajnog bajta operanda
		4	1	Adresa operanda	Operand
9B	3	1	1	Adresa instrukcije	Operacijski kod
		2	1	Adresa instrukcije +1	Adresa operanda
		3	1	Adresa operanda	Operand

Broj ciklusa potreban mikroprocesoru M6800 za izvođenje prve instrukcije je 4, a druge 3 ciklusa. Vidimo da se razmatranja na jednostavnom modelu u ovom primeru podudaraju s izvođenjem u stvarnim mikroprocesorima.

PITANJA

Osnovni delovi mikroračunara

Vrste sabirnica

Model mikroprocesora - registri i osnovni delovi

Centralna memorija (Instrukcije, podaci - program)

Faze u kojima se mikroprocesor nalazi (pribavi, izvrši)

Faza pribavi

Faza izvrši