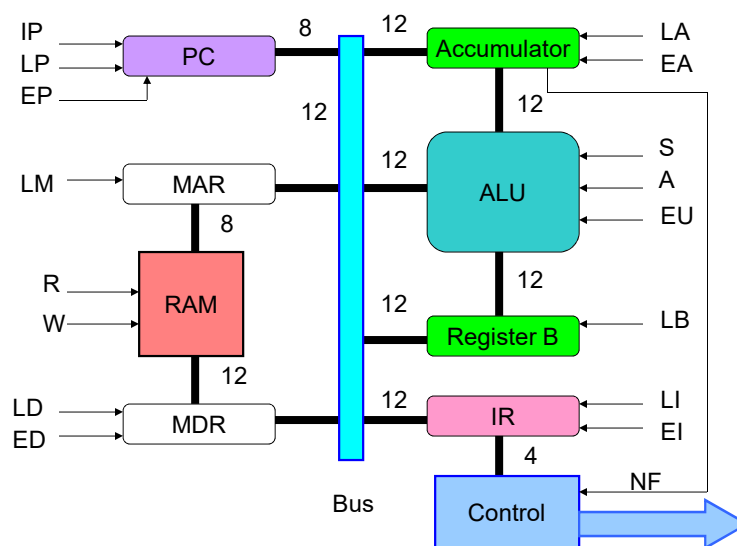


Primer projektovanja CU jednostavnog procesora

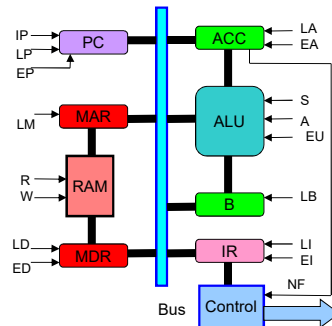
- hardverska CU bazirana na brojaču sekvenci
- mikroprogramska CU

Model jednostavnog računara



Model jednsotavnog računara

- * Jedna 12-bitna magistrala se koristi za prenos informacija između parova registara unutar staze podataka.
- * Registrima i memorijom se upravlja pomoću 16 upravljačkih (control) signala.
- * Većina registara ima Load (L) i Enable (E) ulaze na koje se dovode odgovarajući upravljački signali.
- * Aktivni L signal prisutan na ulazu registra uzrokuje da se sadržaj koji se nalazi na magistrali upiše u registar pri nailasku narednog klok impulsa.
- * Aktivni E signal otvara ulaz registra i omogućava da se njegov sadržaj nađe na magistrali.
- * A, S (Add, Subtract) – sabiranje, oduzimanje
- * Bit znaka (NF –negative flag) iz ACC za control unit
- * IP = Increment PC
 - Tako npr. da bi se preneo sadržaj registra A u registar B neophodno je prvo aktivirati signal EA (koji će omogućiti da se sadržaj registra A nađe na magistrali) a zatim signal LB koji će omogućiti da se sadržaj sa magistrale upiše u registar B.



Model jednsotavnog računara

- * Obradu podataka obavlja ALU (Arithmetic-Logic-Unit, ALU, aritmetičko-logička- jedinica) , kolo koje je u stanju da obavi sabiranje ili oduzimanje dva 12-bitna broja koji se nalaze u dva registra: akumulatoru (ACC) i registru B.
 - Koju će operaciju obaviti ALU definišu upravljački signali A (Add) ili S (Subtract).
 - Akumulator ima jedan flip-flop koji se postavlja kada je rezultat negativan (tj. kada je prvi bit postavljen na 1) .
 - Vrednost ovog bita predstavlja ulaz u kontroler/ sekvencer
 - RAM memoriji računara se pristupa tako što se prvo postavi 8-bitna adresa u Memory Address Register (MAR).
 - Aktivni Read (R) upravljački signal će zatim uzrokovati da se sadržaj adresirane memorijske lokacije prenese u Memory Data Register (MDR).
 - S druge strane, aktivni Write (W) signal uzrokuje da se sadržaj MDR upiše u RAM na adresu specificiranu sadržajem MAR registra.
 - Reč zapamćena u memoriji može biti podatak ili instrukcija

Model jednsotavnog računara

- * Staza podataka sadrži i programski brojač PC čija je funkcija da ukazuje na adresu sledeće instrukcije u memoriji koja treba da se izvrši.
 - Upravljački signal Increment Program Counter (IP) uzrokuje inkrementiranje sadržaja PC za jedan.
 - Ovo omogućava sekvencijalno izvršenje instrukcija.
- * Pored toga sastavni deo staze podataka je i instrukcioni registar (IR) koji pamti instrukciju koja se trenutno izvršava i obezbeđuje opcode bitove za kontroler/sekvencer.

Skup instrukcija

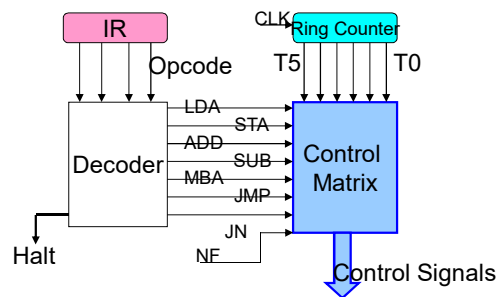
- * Instrukcije su 12-bitne.
 - Prva 4 bita definišu kod operacije (opcode), a preostalih 8 bitova, kada se koriste, definišu memorijsku adresu operanda.
 - Za instrukcije koje imaju dva operanda, jedan je uvek u akumulatoru (uzeli smo primer jedno-adresne mašine).

tabela 1: skup instrukcija

Mnemonic	Opcode	Action	Register Transfers	Active Controls
LDA Load Accumulator	1	$A \leftarrow (\text{Mem})$	1. $MAR \leftarrow IR$ 2. $MDR \leftarrow M(MAR)$ 3. $A \leftarrow MDR$	EI,LM R ED,LA
STA Store Accumulator	2	$(\text{Mem}) \leftarrow A$	1. $MAR \leftarrow IR$ 2. $MDR \leftarrow A$ 3. $M(MAR) \leftarrow MDR$	EI,LM EA,LD W
ADD	3	$A \leftarrow A+B$	1. $A \leftarrow ALU(\text{Add})$	A,EU,LA
SUB	4	$A \leftarrow A-B$	1. $A \leftarrow ALU(\text{Sub})$	S,EU,LA
MBA	5	$B \leftarrow A$	1. $B \leftarrow A$	EA,LB
JMP	6	$PC \leftarrow \text{Mem}$	1. $PC \leftarrow IR$	EI,LP
JN (jump if negative)	7	$PC \leftarrow \text{Mem}$ If negative flag is set	1. $PC \leftarrow IR$ if NF is set	NF : EI,LP
HLT	8-15	Stop Clock		
"Fetch"		$IR \leftarrow \text{Next Instruction}$	1. $MAR \leftarrow PC$ 2. $MDR \leftarrow M(MAR)$ 3. $IR \leftarrow MDR$	EP,LM R ED,LI,IP

blok dijagram sa unutrašnjom organizacijom hardvereske CU bazirane na korišćenju kružnog brojača

- * CU se sastoji od sledećih funkcionalnih jedinica:
 - kružni brojač
 - dekodek instrukcija
 - upravljačka matrica (control matrix)
- * Ulaz u CU je 4-bitni kod operacije instrukcije koja se trenutno nalazi u instrukcionom registru, i flag znaka iz akumulatora.
- * Izlaz iz CU je 16 upravljačkih signala koji dosežu do različitih registara i do memorije računara, pored HLT signala koji se aktivira kad god je prvi bit opcode jednak jedan.



implementacija kružnog brojača

- * Kružni brojač generiše sekvencu od 6 uzastopnih aktivnih signala koja se stalno ponavlja.
 - Kružni brojač je sinhronizovan sa sistemskim klokom (taktom) i on prvo aktivira liniju T0, zatim T1, itd.
 - Nakon aktiviranja T5, sekvenca se vraća na početak, tj počinje se od T0.

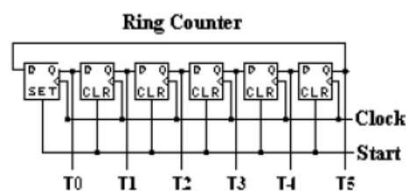


Figure 3. The Internal Organization of the Ring Counter

Dekoder

- * Dekoder instrukcija ima 4-bitni ulaz koji potiče od opcode polja instrukcije, i na osnovu njega aktivira jednu i samo jednu od svojih 8 izlaznih linija.
 - Svaka linija odgovara jednoj instrukciji iz seta instrukcija našeg računara.

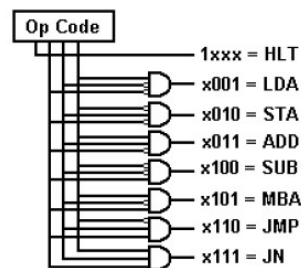


Figure 4. The Internal Organization of the Hard-wired Instruction Decoder

Upravljačka matrica

* Najvažniji deo hardverske CU je upravljačka matrica (control matrix).

- Ona dobija ulaze sa kružnog brojača i dekodera instrukcija i obezbeđuje korektnu sekvencu upravljačkih signala neophodnih za izvršenje svake instrukcije.

trenutci aktiviranja pojedinih signala

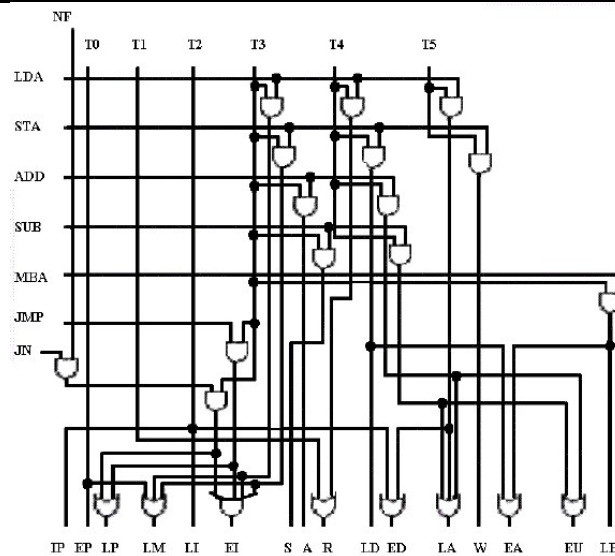
	IP	LP	EP	LM	R	W	LD	ED	LI	EI	LA	EA	A	S	EU	LB
Fetch	T2		T0	T0	T1			T2	T2							
LDA				T3	T4			T5		T3	T5					
STA				T3		T5	T4			T3		T4				
MBA												T3				T3
ADD											T3		T3		T3	
SUB											T3			T3	T3	
JMP		T3								T3						
JN		T3* NF								T3* NF						

$IP = T2;$ $R = T1 + T4 * LDA;$ $LI = T2;$
 $LP = T3 * JMP + T3 * JN * NF;$ $W = T5 * STA;$ $A = T3 * ADD;$
 $EP = T0;$ $LD = T4 * STA;$ $S = T3 * SUB;$
 $LM = T0 + T3 * LDA + T3 * STA$ $ED = T2 + T5 * LDA;$

- Prva kolona tabele prikazuje instrukcije (uključujući i fetch fazu).
- Ostale kolone prikazuju koji signali se aktiviraju za svaku instrukciju u kom stanju kružnog brojača (T0, T1, T2, T3, T4 ili T5).

• Tako, npr., Fetch operacija zahteva aktiviranje signala EP i LM u stanju T0, signala R u stanju T1, i aktiviranje signala IP, Ed i LI u stanju T2.

izgled upravljačke matrice



re 5. The Internal Organization of the Hard-wired Control Matrix

jednačine na osnovu kojih se dobija upravljačka matrica

```

IP = T2
W = T5*STA
LP = T3*JMP + T3*NJ*JN
LD = T4*STA
LA = T5*LDA + T4*ADD + T4*SUB
EA = T4*STA + T3*MBA
EP = T0
S = T3*SUB
A = T3*ADD
LI = T2
LM = T0 + T3*LDA + T3*STA
ED = T2 + T5*LDA
R = T1 + T4*LDA
EU = T3*ADD+T3*SUB
EI = T3*LDA + T3*STA + T3*JMP + T3*NJ*JN
LB = T3*MBA
    
```

Figure 6. The logical equations required for each of the hardwired control signals on the basic computer. The machine's control matrix is designed from these equations.

Mikroprogramska jedinica

- * Skup upravljačkih signala koji dovode do izvršenja specifične mikro-operacije predstavlja mikroinstrukciju koja može biti zapamćena u memoriji.
 - Svaki bit mikroinstrukcije može odgovarati jednom upravljačkom signalu.
 - Ako je bit postavljen na 1, to znači da je odgovarajući upravljački signal aktivan.
- * Niz mikroinstrukcija koji implementira jednu mašinsku instrukciju čini mikro-program.
 - Sekvenca mikroinstrukcija je zapamćena u posebno upravljačkoj (control) memoriji.
 - Izvršenje jedne mašinske instrukcije se postiže pribavljanjem odgovarajuće sekvence mikroinstrukcija iz upravljačke memorije koja se koristi za postavljanje odgovarajućih signala u stazi podataka.

Mikroprogramska jedinica

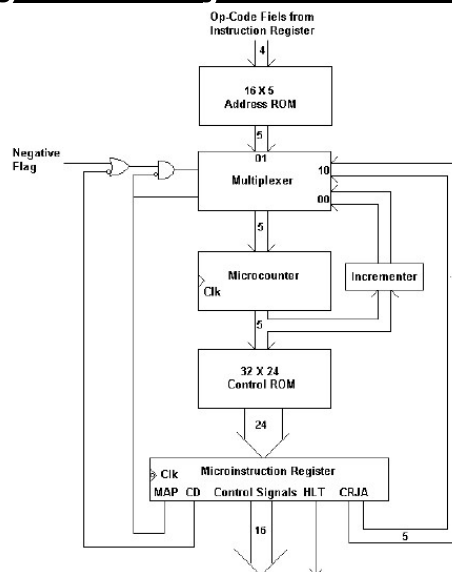
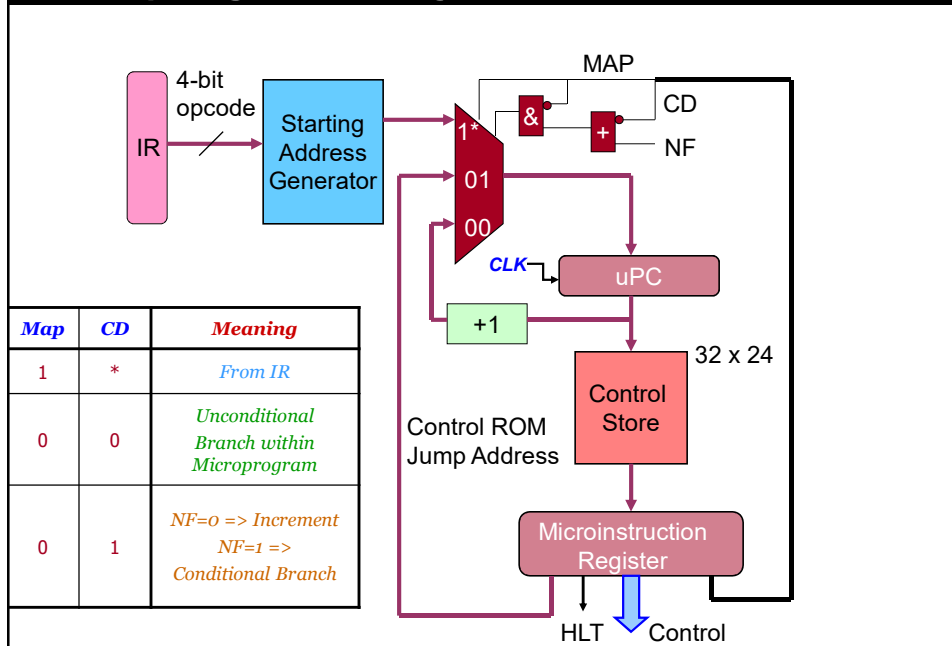


Figure 7. A Microprogrammed Control Unit for the Simple Computer

Mikroprogramska jedinica



Mikroprogramska CU

- * Srce CU je upravljačka ROM memorija kapaciteta 32x24 u koju se mogu zapamtiti do 32 mikroinstrukcije dužine 24 bita.
- * Reč pribavljena iz ROM memorije se pamti u mikroinstrukcioni registar (24-bitni registar).

- Svaka mikro-instrukcija ima dva polja: 16-bitno polje upravljačkih signala i 8-bitno polje sledeće adrese (next-address).
 - Prvih 16 bitova predstavljaju upravljačke signale koji se vode na upravljačke tačke staze podataka.
 - Polje sledeće adrese sadrži bitove koji određuju adresu sledeće mikroinstrukcije koja treba da se pribavi iz upravljačke ROM.

1	1	1	5
CD	MAP	HLT	CRJA

Figure 8. Next address field of the microinstruction register. CD is the condition bit, MAP causes the address of the next microinstruction to be obtained from the address ROM, HLT stops the clock, and CRJA is the control ROM jump address field.

- Adresa koja se koristi za pristup upravljačkoj ROM memoriji dolazi iz mikrobrojača, koji je analogan programskom brojaču.
- Mikrobrojač, dobija podatke preko multipleksera koji selektuje ili (1) izlaz iz adresne ROM, (2) vrednost dobijenu iz inkrementera adrese, ili (3) adresu zapamćenu u polju sledeće adrese mikroinstrukcije.

Mikroprogramska CU

* Adresni ROM se puni adresom startne instrukcije mikro-programa za svaku instrukciju iz seta instrukcija.

- Ova adresa se dobija na osnovu koda operacije (opcode polja mašinske instrukcije) .
- Adresa nula u adresnom ROM sadrži adresu fetch rutine;
- svaka druga adresa u adresnom ROM odgovara jednom od op-kodova iz skupa instrukcija.

Control Store							
Instruction	Op-Code	Instruction Address	Control Signals	CD	MAP	HLT	Addr. Of Next
Fetch	0	00	0011000000000000	0	0	0	01
		01	0000100000000000	0	0	0	02
		02	1000000110000000	0	1	0	XX
LDA	1	03	0001000001000000	0	0	0	04
		04	0000100000000000	0	0	0	05
		05	0000000100100000	0	0	0	00
STA	2	06	0001000001000000	0	0	0	07
		07	0000001000010000	0	0	0	08
		08	0000010000000000	0	0	0	00
ADD	3	09	000000000101010	0	0	0	00
SUB	4	0A	000000000100110	0	0	0	00
MBA	5	0B	000000000010001	0	0	0	00
JMP	6	0C	0100000001000000	0	0	0	00
JN	7	0D	0000000000000000	1	0	0	0F
		0E	0000000000000000	0	0	0	00
		0F	0100000001000000	0	0	0	00
Expansion	8-E	10-1E					
HLT	F	1F	0000000000000000	0	0	1	XX
Control Word							

Example 1 – MBA followed by ADD

			I	L	E	L	E	L	E	A	S	E	L					
			P	P	P	M	R	W	D	D	I	I	A	A	S	U	B	
Fetch	0	00	0011000000000000										0	0	0	01		
		01	0000100000000000										0	0	0	02		
		02	1000000110000000										0	1	0	XX09		
LDA	1	03	0001000001000000										0	0	0	04		
		04	0000100000000000										0	0	0	05		
		05	0000000100100000										0	0	0	00		
STA	2	06	0001000001000000										0	0	0	07		
		07	0000001000010000										0	0	0	08		
		08	0000010000000000										0	0	0	00		
ADD	3	09	0000000000101010										0	0	0	00		
SUB	4	0A	0000000000100110										0	0	0	00		
MBA	5	0B	000000000010001										0	0	0	00		
JMP	6	0C	0100000001000000										0	0	0	00		
JN	7	0D	0000000000000000										1	0	0	0F		
		0E	0000000000000000										0	0	0	00		
		0F	0100000001000000										0	0	0	00		
Expansion	8-E	10-1E																
HLT	F	1F	0000000000000000										0	0	1	Stop clk		

Sequence for MBA,ADD

MOV B,A

- 1. MAR ← PC 0011000000000000
- 2. MDR ← M(MAR) 0000100000000000
- 3. IR ← MDR 1000000110000000
- B ← A 0000000000010001

ADD

- 1. MAR ← PC 0011000000000000
- 2. MDR ← M(MAR) 0000100000000000
- 3. IR ← MDR 1000000110000000
- A ← ALU(Add) 000000000101010

