

**Univerzitet u Kragujevcu
TEHNIČKI FAKULTET
Čačak**



SPECIFIKACIJA ARHITEKTURE PROCESORA

TFCOmin

**Verzija 1.03
Interni izveštaj**

Laboratorija za računarsku tehniku



**Čačak
Jun 2012. godine**

SPECIFIKACIJA ARHITEKTURE PROCESORA TFaComin

Instrukcije procesora TFaComin su fiksne dužine 16 bita. Osnovni format instrukcije je prikazan na slici 1.



Slika 1. Osnovni format instrukcije procesora TFaComin

Shodno usvojenom formatu instrukcije arhitektura procesora TFaCo ima sledeće karakteristike:

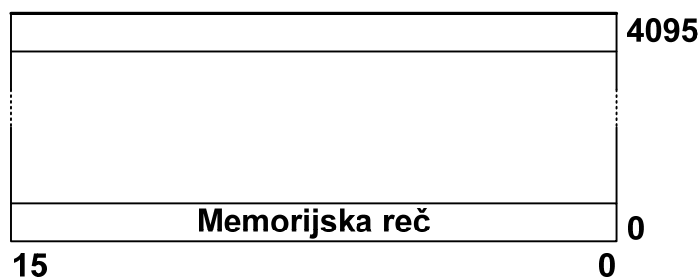
- Dužina operacionog dela instrukcije (COP) je **4 bita** ($I_{12} - I_{15}$).
- Funkcija ostalih bitova instrukcije ($I_0 - I_{11}$) zavisi od vrste instrukcije i primenjenog načina adresiranja.
- Dužina procesorske reči je **16 bita**, a istu dužinu ima i **memorijska reč**, čija je dužina prilagođena dužini instrukcije, koja se na taj način može dohvatati u okviru jednog memorijskog ciklusa.

Arhitektura procesora TFaComin odgovara **akumulatorskoj mašini** tako da su instrukcije jednoadresne. Preko **akumulatora** se vrši prenos podataka u/iz memorije.

Načini adresiranja

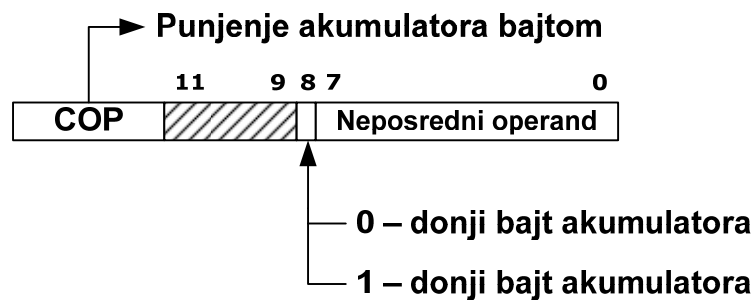
Procesor TFaComin omogućava sledeće načine adresiranja

1. **Direktno memorijsko adresiranje** – Kod ovog načina adresiranja bitovi instrukcije **0 – 11** predstavljaju adresu adresibilne jedinice (memorijske reči dužine 16 bita) unutar stranice od 2^{12} (**4096**) lokacija što je istovremeno i adresni prostor ovog procesora. Struktura memorije sistema TFaComin je prikazana na slici 2. Shodno ovome dužina **memorijskog adresnog registra (MAR)** iznosi 12 bitova. Ovaj način adresiranja primenjuje se i kod instrukcija kojima se vrši prenos podataka između **akumulatora** i **memorije**.



Slika 2. Struktura operativne memorije

2. **Neposredno adresiranje** – Ovaj način adresiranja za punjenje akumulatora neposrednom vrednošću operanda. Usvojeno je da dužina neposrednog operanda bude **8 bita (1 bajt)**. U tom slučaju bitovi instrukcije $I_8 - I_{11}$ ostaju neiskorišćeni. S obzirom, da je dužina akumulatora 16 bita ova instrukcija može da se iskoristi da se bira u koji bajt akumulatora će se upisati neposredni operand. Za tu namenu se koristi **bit 8**. Ako je $I_8 = 0$ tada se neposredni operand upisuje u donji bajt akumulatora, a ako je $I_8 = 1$ tada se neposredni operand upisuje u gornji bajt akumulatora, kao što je prikazano na slici 3.



Slika 3. Neposredno adresiranje

3. **Instrukcije skoka** – Skokovi kod TFaComin mogu biti bezuslovni ili uslovni. Uslovni skokovi se realizuju na bazi statusnog registra RS u koji se po izvršavanju operacija upisuje odgovarajuća statusna informacija, prema tabeli 1 i slici 4.

Tabela 1. Značenje bitova u registru statusa – R₉

Bit	Oznaka	Značenje
(0)	Z – Zero Bit	
	Z=0	Rezultat operacije različit od 0
	Z=1	Rezultat operacije jednak 0
(1)	N – Negative Bit	
	N=0	Rezultat operacije pozitivan
	N=1	Rezultat operacije negativan
(2)	C – Carry Bit	
	C=0	Nema prenosa iz najstarijeg razreda rezultata
	C=1	Postoji prenos iz najstarijeg razreda rezultata
(3)	O – Overflow Bit	
	O=0	Nema prekoračenja opsega računanja
	O=1	Došlo je do prekoračenja opsega računanja

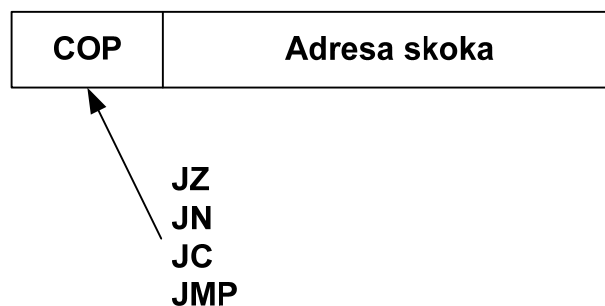
Zbog ograničenog broja bitova namenjenih kodovanju operacija predviđeno je postojanje samo sledećih tipova skoka:

- Bezuslovni skok;
- Uslovni skok po bitu Z;
- Uslovni skok po bitu N;
- Uslovni skok po bitu C.



Slika 4. Struktura statusnog registra RS

Specifikacija uslova koji se testira pri uslovnim skokovima definisana je kodom operacije, a adresa skoka je specificirana u adresnom polju instrukcije, kao što je prikazano na slici 5.



Slika 5. Format instrukcija skoka

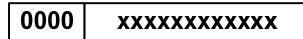
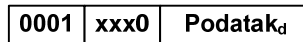
Skup instrukcija TFaCo i opis njihovog izvršavanja

COP	Simbolički kod	Sintaksa	Opis
0000	HALT	HALT	Zaustavljanje programa
0001	LI	LI lb, podatak	<p>Operacija punjenja akumulatora neposrednim operandom</p> <ul style="list-style-type: none"> ▪ lb – Jednobitna informacija, koji se nalazi na lokaciji I_8, i koja definiše u koji bajt se upisuje neposredni operand, koji je dužine 1 bajt. <ul style="list-style-type: none"> ○ lb=0 – podatak se upisuje u donji bajt akumulatora ○ lb=1 – podatak se upisuje u gornji bajt akumulatora ▪ podatak – 8 – bitna informacija. Pomoću ove naredbe, u opštem slučaju, mogu da se formiraju sledeći podaci u akumulatoru: <ul style="list-style-type: none"> ○ Neoznačena veličina dužine 16 bita ○ Označena veličina dužine 16 bita ○ Binarni broj dužine 16 bita ○ Četiri heksadecimalna broja ○ Dva ASCII karaktera ○ Jedan Unicode karakter
0010	LOAD	LOAD adresa	<p>Puni akumulator podatkom iz memorije</p> <ul style="list-style-type: none"> ▪ adresa – memorijska adresa sa koje se čita podatak
0011	STORE	STORE adresa	<p>Pamti sadržaj akumulatora</p> <ul style="list-style-type: none"> ▪ adresa – memorijska adresa na koju se upisuje podatak
0100	ADD	ADD adresa	<p>Sabiranje $acc \leftarrow (acc) + (adresa)$ – Funkcionalni format instrukcije za sabiranje</p>
0101	SUB	SUB adresa	<p>Oduzimanje $acc \leftarrow (acc) - (adresa)$ – Funkcionalni format instrukcije za oduzimanje</p>
0110	NOT	NOT adresa	<p>Logička negacija $acc \leftarrow \text{not}(adresa)$</p>
0111	OR	OR adresa	<p>Logičko ILI $acc \leftarrow (acc) \text{ or } (adresa)$ – Funkcionalni format instrukcije za logičko ILI</p>
1000	AND	AND adresa	<p>Logičko I $acc \leftarrow (acc) \text{ and } (adresa)$ – Funkcionalni format instrukcije za logičko I</p>
1001	XOR	XOR adresa	<p>Ekskluzivno ILI $acc \leftarrow (acc) \text{ or } (adresa)$ – Funkcionalni format instrukcije za ekskluzivno ILI</p>
1010	IN	IN adresa	<p>Puni akumulator podatkom iz I/O uređaja $acc \leftarrow (adresa)$</p> <ul style="list-style-type: none"> ▪ adresa - adresa I/O porta sa koga se čita podatak
1011	OUT	OUT adresa	<p>Šalje podatak iz akumulatora u I/O uređaj $adresa \leftarrow (acc)$</p> <ul style="list-style-type: none"> ▪ adresa - adresa I/O porta na koji se šalje podatak
1100	JZ	JZ adresa	Skok ako je rezultat nula ($Z=1$)
1101	JN	JN adresa	Skok ako je rezultat negativan ($N=1$)
1110	JC	JC adresa	Skok ako postoji prenos ($C=1$)
1111	JMP	JMP adresa	Instrukcija bezuslovnog skoka

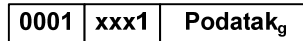
Opisi binarnog formata instrukcija TFaCo

Pri objašnjavanju načina kodovanja instrukcija koristiće se sledeća pravila za označavanje

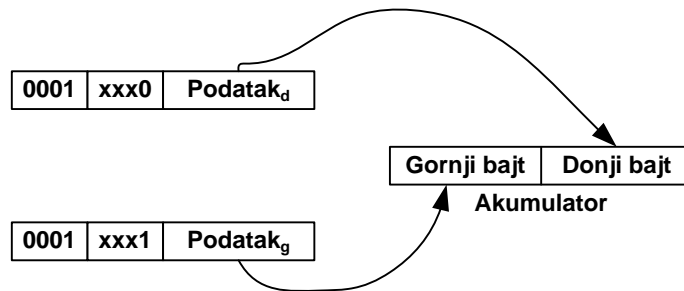
- 0 – Vrednost bita jednaka 0
- 1 – Vrednost bita jednaka 1
- x – Vrednost bita nije bitna

HALT – Zaustavljanje programa**LI – Punjenje akumulatora neposrednim operandom**

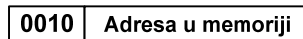
Upis podatka u donji bajt akumulatora



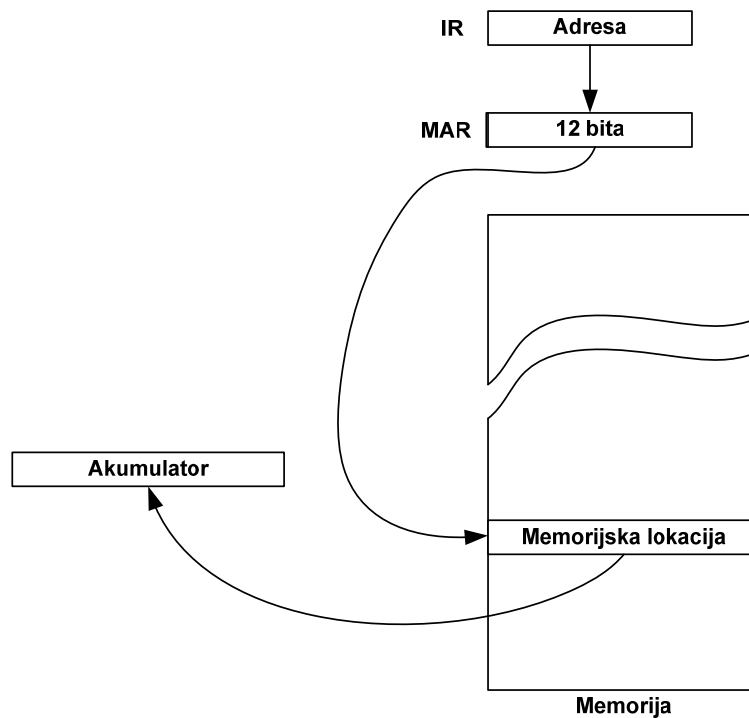
Upis podatka u gornji bajt akumulatora



Način izvršenja LI instrukcije

LOAD – Punjenje akumulatora sadržajem memorije

Punjenje akumulatora sadržajem adresirane memorijske lokacije – specificirana adresa odgovara memorijskoj lokaciji sa koje se čita podatak

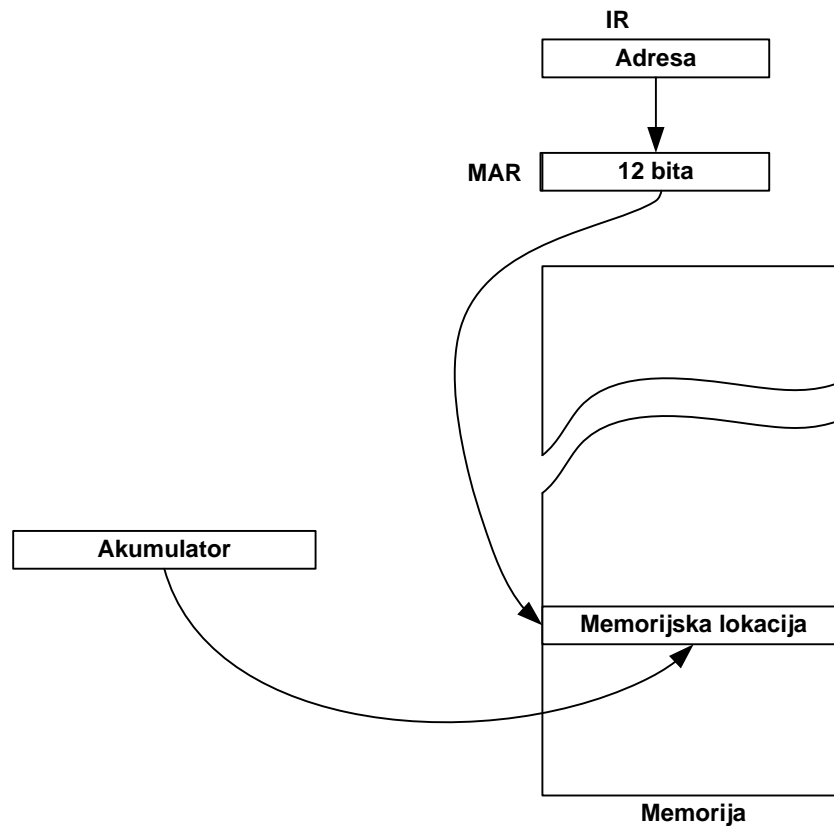


Način izvršavanja instrukcije LOAD

STORE – Pamćenje sadržaja akumulatora u memoriji

0011	Adresa u stranici
------	-------------------

Pamćenje sadržaja akumulatora u adresiranoj memorijskoj lokaciji – specificirana adresa odgovara memorijskoj lokaciji na kojoj se pamti podatak



Način izvršenja instrukcije STORE

ADD – Sabiranje

0100	Adresa
------	--------

Format instrukcije ADD

SUB – Oduzimanje

0101	Adresa
------	--------

Format instrukcije SUB

NOT – Logička negacija

0110	Adresa
------	--------

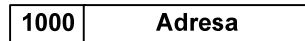
Format instrukcije NOT

OR – Logičko ILI

0111	Adresa
------	--------

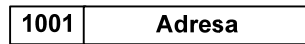
Format instrukcije OR

AND – Logičko I



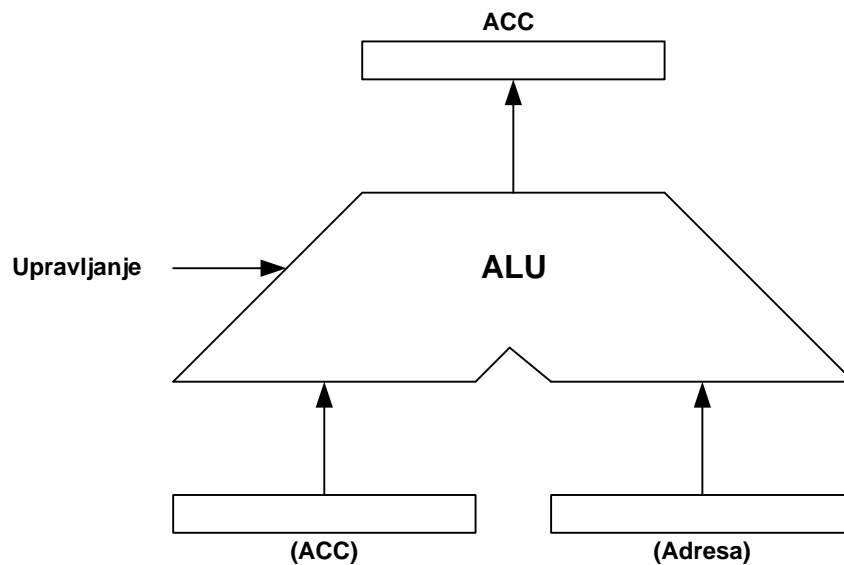
Format instrukcije AND

XOR – Ekskluzivno ILI

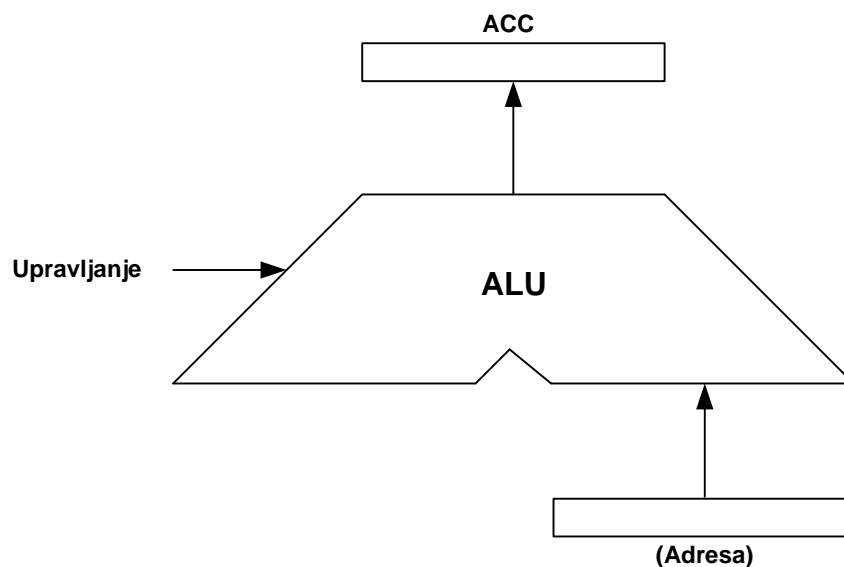


Format instrukcije XOR

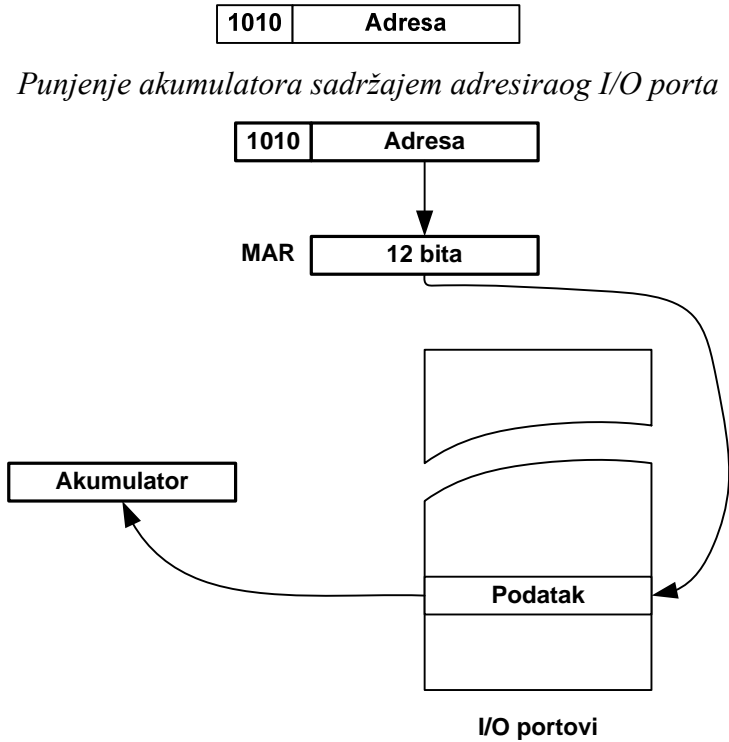
Aritmetičko – logičke instrukcije ADD, SUB, OR, AND i XOR izvršavaju se prema šemi na sledećoj slici:



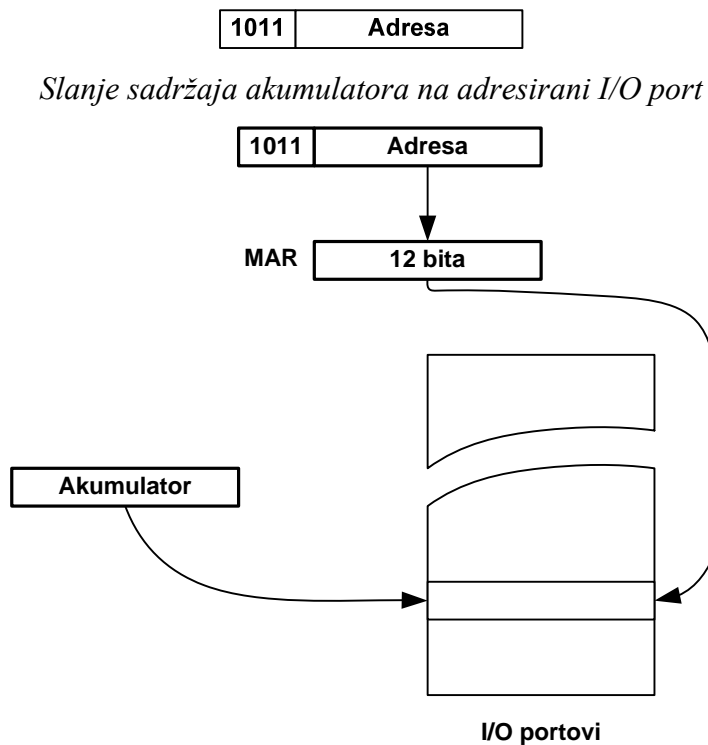
Šema izvršavanja aritmetičko – logičkih operacija



Šema izvršavanja instrukcije NOT

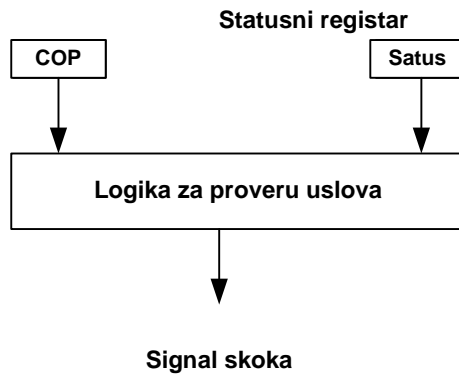
IN – Punjenje akumulatora podatkom iz I/O uređaja

Način izvršavanja instrukcije IN

OUT – Slanje sadržaja akumulatora u I/O uređaj

Način izvršenja instrukcije OUT

JZ, JN, JC, JMP – Uslovni/bezuslovni skok



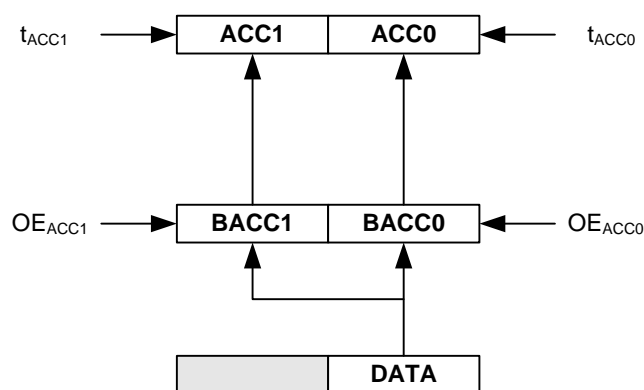
Šema izvršenja instrukcije skoka

PRIKAZ MIKROPERACIJA INSTRUKCIJA TFaComin

LI – Punjenje akumulatora neposrednim operandom

Neposredni operand se pre unošenja u akumulator nalazi u registru instrukcije (IR) i dužine je 8 bita. Ovaj podatak može da se upiše u akumulator na nižu ili višu poziciju, što znači da je potrebno izvršiti odgovarajuće demultipleksiranje. U principu ovde može da se napravi proširenje dejstva ove naredbe tako što bi se omogućila operacija umetanja, tj. da se drugi deo sadržaja akumulatora ne menja. Demultipleksiranje se realizuje postavljanjem dvostrukog bafera sa izlazom sa tri stanja na izlaz registra instrukcije IR7 – IR0. U zavisnosti od vrednosti bita **lb** otvara se izlaz bafera koji propušta informacije iz IR na ulaz donjeg ili gornjeg dela akumulatora.

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
	If lb = 0 then OE _{ACC0}	OE _{ACC0} – Upravlja three – state baferom na ulaz u donji bajt akumulatora	BACC0 – Three – state bafer za multipleksiranje neposrednog operanda na ulaz donjeg bajta akumulatora	
	If lb = 1 then OE _{ACC1}	OE _{ACC1} – Upravlja three – state baferom na ulaz u gornji bajt akumulatora	BACC1 – Three – state bafer za multipleksiranje neposrednog operanda na ulaz gornjeg bajta akumulatora	
ACC←(IR)₇₋₀				Na željeno mesto u akumulatoru upisuje se neposredni operand definisan donjim bajtom registra instrukcije.
			Statusni bitovi:	



Upis neposrednog operanda u odgovarajuće bajtove akumulatora

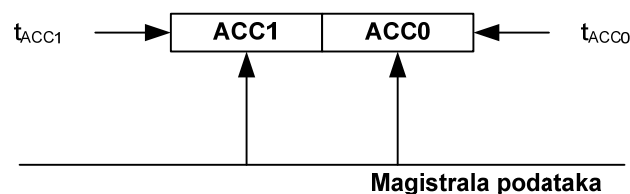
$$OE_{ACC0} = (\overline{IR}_{15}\overline{IR}_{14}\overline{IR}_{13}IR_{12})\overline{lb}$$

$$OE_{ACC0} = (\overline{IR}_{15}\overline{IR}_{14}\overline{IR}_{13}IR_{12})lb$$

Logičke funkcije signala OE_{ACC0} i OE_{ACC1}

LOAD – Punjenje akumulatora sadržajem memorijske lokacije

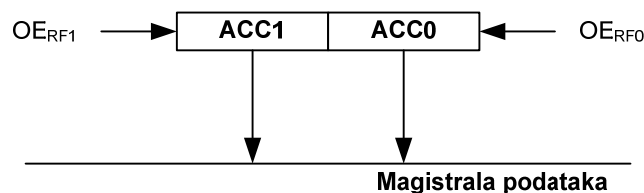
MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj upisuje u akumulator.
D₁₅–D₀←(Memorija)				Pročitani sadržaje se postavlja na magistralu podataka
ACC←D₁₅–D₀				Podaci magistrale se upisuju u akumulator
			Statusni bitovi: Z, N	



Upis podataka sa magistrale podataka u akumulator

STORE – Prenos sadržaja akumulatora u memoriju

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije u koju se upisuje sadržaj akumulatora.
D₁₅–D₀←(Akumulator)				Pročitani sadržaje se postavlja na magistralu podataka
Memorija←D₁₅–D₀				Podaci magistrale se upisuju u memoriju. Generiše se signal t_{MW} .
			Statusni bitovi:	



Čitanje podataka iz akumulatora na magistralu podataka

$$OE_{RF1} = OE_{RF0} = ARF_3\overline{ARF_2}\overline{ARF_1}\overline{ARF_0}$$

Formiranje signala dozvole izlaska podataka iz RF na magistralu podataka

ADD – Sabiranje

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_A←(Akumulator)				Postavljanje sadržaja akumulatora na ulaz u ALU
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na ulaz u ALU
ADD: (Akumulator)+(Memorija)				Izvršavanje operacije sabiranja nad sadržajem akumulatora i memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z, N, C, O	

SUB – Oduzimanje

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_A←(Akumulator)				Postavljanje sadržaja akumulatora na ulaz u ALU
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na ulaz u ALU
ADD: (Akumulator)-(Memorija)				Izvršavanje operacije oduzimanja nad sadržajem akumulatora i memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z, N, C, O	

NOT Logička negacija

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na

				ulaz u ALU
ADD: not(Memorija)				Izvršavanje operacije negacije na sadržajem memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z	

OR – Logičko ILI

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_A←(Akumulator)				Postavljanje sadržaja akumulatora na ulaz u ALU
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na ulaz u ALU
ADD: (Akumulator) or (Memorija)				Izvršavanje operacije Logičkog ILI nad sadržajem akumulatora i memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z	

AND logičko I

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_A←(Akumulator)				Postavljanje sadržaja akumulatora na ulaz u ALU
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na ulaz u ALU
ADD: (Akumulator) and (Memorija)				Izvršavanje operacije logičkog I nad sadržajem akumulatora i memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z	

XOR – Ekskluzivno ILI

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
ALU←Upravljački signali (operacija)				Generisanje upravljačkih signala za ALU jedinicu
MAR←Adresa				U MAR se upisuje adresa memorijske lokacije čiji se sadržaj dovodi u operaciju sa sadržajem akumulatora
ALU_A←(Akumulator)				Postavljanje sadržaja akumulatora na ulaz u ALU
ALU_B←(Memorija)				Postavljanje sadržaja memorijske lokacije na ulaz u ALU
ADD: (Akumulator) xor (Memorija)				Izvršavanje operacije Ekskluzivno ILI nad sadržajem akumulatora i memorijske lokacije
Akumulator←Rezultat				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z	

IN – Ulaz podatka sa I/O uređaja

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
IO←Upravljački signali (operacija)				Generisanje upravljačkih signala za IO blok
MAR←Adresa				U MAR se upisuje adresa IO lokacije čiji se sadržaj unosi u akumulator
IO←(Akumulator)				Pamćenje rezultata u akumulatoru
			Statusni bitovi: Z, N	

OUT – Izlaz podatka na I/O uređaj

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
IO←Upravljački signali (operacija)				Generisanje upravljačkih signala za IO blok
MAR←Adresa				U MAR se upisuje adresa IO lokacije u koju se upisuje sadržaj akumulatora
Akumulator←(IO)				Pamćenje rezultata u akumulatoru
			Statusni bitovi:	

JZ, JN, JC – Uslovni skokovi

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
IR←(Memorija)				Po adresi u MAR se čita instrukcija
Test				Testiranje zadatog uslova
MAR←Adresa/(PC)				U MAR se upisuje adresa skoka u slučaju uspešnosti testa. U suprotnom upisuje se sadržaj PC.
PC←(MAR)				U slučaju da je test uspešan (postoji skok) sadržaj MAR – a se prepisuje u PC.
PC←(PC)+1				U slučaju da je test uspešan (postoji skok) – programski brojač se uvećava za 1.

JMP – Bezuslovni skok

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
MAR←Adresa				U MAR se upisuje adresa bezuslovnog skoka
PC←(MAR)				Sadržaj MAR – a se prepisuje u PC
PC←(PC)+1				Programski brojač se uvećava za 1.

HALT – Zaustavljanje programa

MAR←Adresa				U MAR se upisuje adresa instrukcije
PC←(PC)+1				Uvećava se sadržaj programskog brojača
Halt				Generisanje signala zaustavljanja sinhro jezgra