

**ЈОВАН ЂОРЂЕВИЋ**

**АРХИТЕКТУРА  
И  
ОРГАНИЗАЦИЈА  
РАЧУНАРА**

**ГЛАВА I**

**УВОД**

**БЕОГРАД, 2013.**



ДҖМ



# ПРЕДГОВОР

Ова књига је написана као основни уџбеник из архитектуре и организације рачунара и покрива основне концепте из архитектуре рачунара и организације процесора, магистрале, улаза/излаза и меморије.

Аутор

Београд

септембра 2013.



# САДРЖАЈ

ПРЕДГОВОР .....	I
САДРЖАЈ .....	3
<b>1 УВОД.....</b>	<b>1</b>
1.1 СТРУКТУРА РАЧУНАРА .....	1
1.1.1 Меморија .....	1
1.1.2 Процесор.....	3
1.1.3 Улазно/излазни уређаји.....	6
1.1.4 Магистрала.....	9
1.2 ФАЗЕ ИЗВРШАВАЊА ИНСТРУКЦИЈЕ .....	10
1.3 АРХИТЕКТУРА И ОРГАНИЗАЦИЈА РАЧУНАРА.....	12





# 1 УВОД

У овој глави се разматрају неки елементи концепта рада и структуре рачунара. У оквиру тога се, најпре, дефинишу типови информација у рачунару, и то инструкције и операнди, дају функције модула рачунара, и то меморије, процесора, улазно/излазних уређаја и магистрале, и приказује структура рачунара. Потом се дефинишу фазе извршавања инструкције. На крају се дефинишу елементи архитектуре и организације рачунара.

## 1.1 СТРУКТУРА РАЧУНАРА

Рачунари су електронски уређаји у којима се решавање одређених проблема реализује извршавањем одређеног скупа аритметичких, логичких и померачких операција. Операције које се у рачунару извршавају се представљају помоћу бинарних речи које се називају инструкције, команде или наредбе. Скуп операција које рачунар може да извршава је такав да било који проблем који треба да се решава у рачунару може да се разложи на уређени низ инструкција рачунара који се назива програм. Подаци над којима се операције извршавају се, такође, представљају помоћу бинарних речи које се називају операнди.

### 1.1.1 Меморија

За складиштење оба типа бинарних речи користи се модул рачунара који се назива меморија.

Бинарне речи које представљају инструкције имају структуру дату на слици 1.

OC	SRC1	SRC2	DST
----	------	------	-----

Слика 1 Структура инструкције

Група битова означена са OC специфицира операцију коју треба извршити. Групе битова означене са SRC1 и SRC2 представљају адресе локација у меморији са којих треба прочитати операнде и над њима извршити операцију. Група битова означена са DST представља адресу локације у меморији у коју треба сместити резултат. Ове групе битова се још називају: OC – поље кода операције (operation code), SRC1 и SRC2 – поља два изворишна (source) операнда и DST – поље одредишног (destination) операнда. Поља SRC1, SRC2 и DST се називају и адресна поља инструкције. Бинарне речи које представљају инструкције се могу интерпретирати и на друге начине у ситуацијама када адресе неких операнда нису дате експлицитно одговарајућим адресним пољима инструкције, већ су њихове адресе имплицитно одређене.

Бинарне речи које представљају операнде имају различиту структуру у зависности од усвојеног начина интерпретирања битова. Тако, на пример, ако се бинарна реч дужине  $n$  битова, у којој су битови означени са  $a_{n-1}a_{n-2}...a_1a_0$ , интерпретира као целобројна величина без знака, онда она представља податак  $A$  чија се вредност израчунава помоћу израза

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Уз такав начин интерпретирања битова бинарне речи, представљају се целобројне величине без знака у опсегу 0 до  $2^n - 1$ . Међутим, ако се иста бинарна реч интерпретира као целобројна величина са знаком у другом комплементу, онда она представља податак А чија се вредност израчунава помоћу израза

$$A = -2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

Уз овакав начин интерпретирања битова бинарне речи, представљају се целобројне вредности са знаком у опсегу  $-2^{n-1}$  до  $2^{n-1} - 1$ . Бинарне речи које представљају операнде се могу интерпретирати и на друге начине као, на пример, целобројне величине са знаком представљене као знак и величина, целобројне величине са знаком у првом комплементу, величине у покретном зарезу, бинарно кодирани децимални бројеви, низ алфанумеричких знакова итд.

Нека је проблем који треба решити у рачунару дат изразом

$$Y = (A - B) / [C + (D \cdot E)]$$

Да би се овај проблем решио у рачунару потребно га је разложити на уређени низ инструкција којима се реализују операције које рачунар може да извршава. У овом случају то су операције сабирања, одузимања, множења и дељења. Тиме се добија следећи програм:

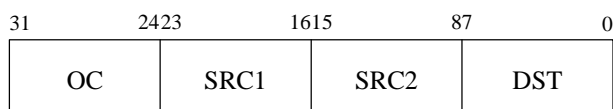
```

SUB  A,  B,  Y
MUL  D,  E,  T
ADD  T,  C,  T
DIV  Y,  T,  Y

```

У инструкцијама програма су кодови операција сабирања, одузимања, множења и дељења означени са ADD, SUB, MUL и DIV, а адресе меморијских локација изворишних и одредишних операнда са А, В, С, D, Е, Y и Т. Извршавањем овог програма у рачунару се, најпре, читају садржаји меморијских локација са адреса А и В, над њима реализује операција одузимања и резултат смешта у меморијску локацију на адреси Y. Затим се читају садржаји меморијских локација са адреса D и Е, над њима реализује операција множења и резултат смешта у помоћну меморијску локацију на адреси Т. Потом се читају садржаји меморијских локација са адреса Т и С, над њима реализује операција сабирања и резултат смешта у меморијску локацију на адреси Т. На крају се читају садржаји меморијских локација са адреса Т и С, над њима реализује операција дељења и резултат смешта у меморијску локацију на адреси Y. Програм је тако написан да садржаји меморијских локација на адресама А, В, С, D и Е остају неизмењени на крају извршавања овог програма. Због тога је и било потребно да се користи помоћна меморијска локација на адреси Т за привремено смештање резултата операција множења и сабирања.

Меморија у коју се смештају инструкције и операнди има, на пример, 256 локација, па се за њихово адресирање користи 8 битова. Инструкције су смештене почев од адресе 10 хексадецимално, а подаци почев од адресе 30 хексадецимално. Инструкције су представљене бинарним речима као на слици 2.



Слика 2 Формат инструкције

Битови 31 до 24 представљају поље кода операције (OC), док битови 23 до 16, 15 до 8 и 7 до 0 представљају поља два изворишна (SRC1 и SRC2) и одредишног (DST) операнда. Дужина инструкције је 32 бита. Операнди су, ради једноставности, представљени као 32 битне целобројне величине без знака и њихове вредности А се интерпретирају према изразу

$$A = \sum_{i=0}^{31} 2^i a_i$$

Како су дужине и инструкција и операнда 32 бита, узето је и да је ширина меморијске речи 32 бита.

Кодови операција додељени операцијама сабирања (ADD), одузимања (SUB), множења (MUL) и дељења (DIV) представљени у хексадецималном облику су: 01 за сабирање, 02 за одузимање, 03 за множење и 04 за дељење. Адресе меморијских локација представљене у хексадецималном облику додељене операндима су 30 за А, 31 за В, 32 за С, 33 за D, 34 за Е, 35 за Y и 36 за Т. Садржаји меморијских локација на адресама 30 до 36, представљени у хексадецималном облику, су 00000009, 00000003, 00000002, 00000004, 00000001, 00000000 и 00000000.

На основу тога делови меморије са инструкцијама и операндима изгледају као на слици 3.

...				
10	02	30	31	35
11	03	33	34	36
12	01	36	32	36
13	04	35	36	35
...				
30	00	00	00	09
31	00	00	00	03
32	00	00	00	02
33	00	00	00	04
34	00	00	00	01
35	00	00	00	00
36	00	00	00	00
...				

Слика 3 Део меморије са инструкцијама и операндима на почетку

### 1.1.2 Процесор

За извршавање овог програма потребан је модул рачунара који се назива процесор. Он мора, најпре, да чита бинарну реч са адресе 10 и да је интерпретира као инструкцију сагласно формату инструкције са слике 2. Затим треба из инструкције да користи поље адресе првог изворишног операнда SRC1 и са адресе 30 прочита бинарну реч, коју треба да интерпретира као операнд представљен као целобројна величина без знака са

вредношћу 9. Потом, из инструкције треба да користи поље адресе другог изворишног операнда SRC2 и са адресе 31 прочита бинарну реч коју треба да интерпретира као операнд представљен као целобројна величина без знака са вредношћу 3. Пошто су оба операнда сада расположива из инструкције треба да користи поље кода операције ОС и на основу вредности 02 овог поља изврши операцију одузимања вредности 3 од 00000009 и резултат 00000006 упише у меморијску локацију на адреси 35 одређеној пољем адресе одредишног операнда DST. Садржај дела меморије у коме се налазе бинарне речи које представљају податке је дат на слици 4. Процесор, затим, мора да за бинарну реч са адресе 11 понови све активности као и за бинарну реч са адресе 10. Процесор ову бинарну реч интерпретира као инструкцију множења, па најпре са адреса 33 и 34 чита бинарне речи које интерпретира као целобројне величине без знака са вредностима 00000004 и 00000001, затим реализује њихово множење и резултат 00000004 уписује у меморијску локацију на адреси 36. Садржај дела меморије у коме се налазе бинарне речи које представљају податке је дат на слици 5.

...				
30	00	00	00	09
31	00	00	00	03
32	00	00	00	02
33	00	00	00	04
34	00	00	00	01
35	00	00	00	06
36	00	00	00	00
...				

Слика 4 Део меморије са инструкцијама и операндима после инструкције SUB

...				
30	00	00	00	09
31	00	00	00	03
32	00	00	00	02
33	00	00	00	04
34	00	00	00	01
35	00	00	00	06
36	00	00	00	04
...				

Слика 5 Део меморије са инструкцијама и операндима после инструкције MUL

Ове активности процесор понавља и за бинарну реч са адресе 12. Процесор ову бинарну реч интерпретира као инструкцију сабирања, па најпре са адреса 36 и 32 чита бинарне речи које интерпретира као целобројне величине без знака са вредностима 00000004 и 00000002, затим реализује њихово сабирање и резултат 00000006 уписује у меморијску локацију на адреси 36. Садржај дела меморије у коме се налазе бинарне речи које представљају податке је дат на слици 6.

...				
30	00	00	00	09
31	00	00	00	03
32	00	00	00	02
33	00	00	00	04
34	00	00	00	01

35	00	00	00	06
36	00	00	00	06
...				

Слика 6 Део меморије са инструкцијама и операндима после инструкције ADD

За случај разматраног програма, процесор понавља ове активности и за бинарну реч са адресе 13. Процесор ову бинарну реч интерпретира као инструкцију дељења, па најпре са адреса 35 и 36 чита бинарне речи које интерпретира као целобројне величине без знака са вредностима 00000006 и 00000006, затим реализује њихово дељење и резултат 00000001 уписује у меморијску локацију на адреси 35. Садржај дела меморије у коме се налазе бинарне речи које представљају податке је дат на слици 7.

...				
30	00	00	00	09
31	00	00	00	03
32	00	00	00	02
33	00	00	00	04
34	00	00	00	01
35	00	00	00	01
36	00	00	00	06
...				

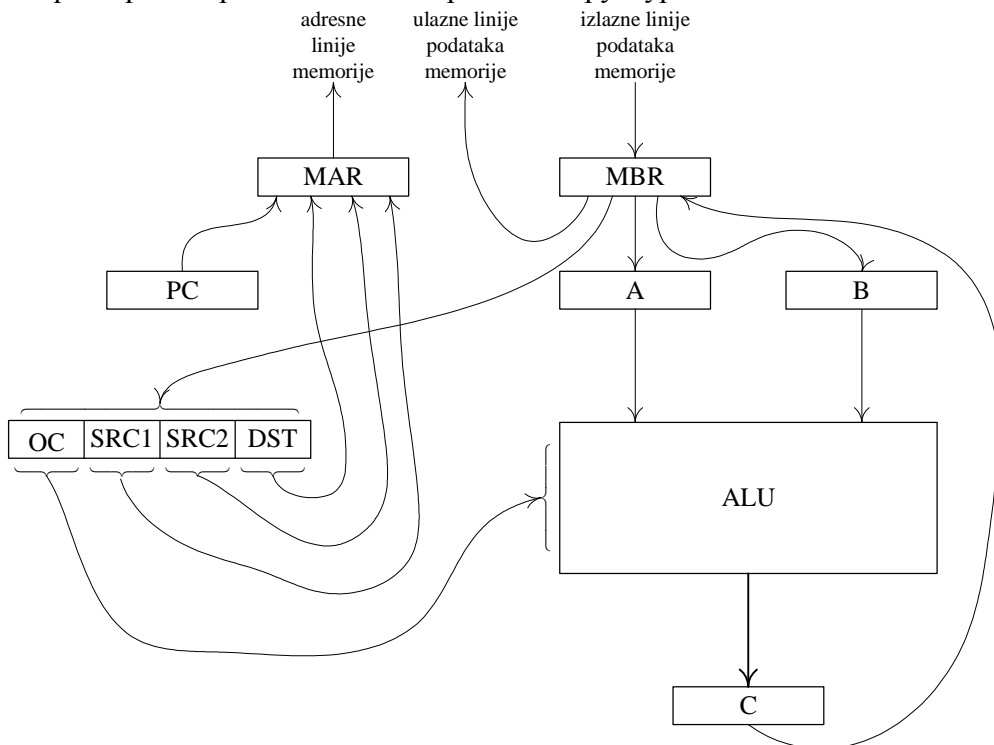
Слика 7 Део меморије са инструкцијама и операндима после инструкције DIV

Резултат проблема срачунавања израза

$$Y=(A-B)/[C+(D-E)]$$

ако вредности променљивих A, B, C, D и E износе 9, 3, 2, 4, и 1, добијен извршавањем програма са слике 3, је 00000001 и налази се у меморијској локацији на адреси 35.

Овакав рад процесора се може остварити са структуром на слици 8.



Слика 8 Структура процесора

На слици 8 је са РС означен бројачки регистар који се назива програмски бројач. Његов садржаја се користи као адреса меморијске локације са које треба читати бинарну реч која се интерпретира као инструкција. С обзиром да су бинарне речи које се интерпретирају као инструкције смештене једна иза друге у меморијским локацијама и да стога треба да се читај секвенцијално, садржај програмског бројача РС се, најпре, користи као адреса меморијске локације са које се чита бинарна реч, па се, затим, његов садржај инкрементира.

Са MAR је означен адресни регистар меморије. У регистар MAR се смешта садржај који представља адресу меморијске локације са које треба прочитати или у коју треба уписати бинарну реч. Садржај регистра MAR се води на адресне линије меморије.

Са MBR је означен прихватни регистар податка меморије. У регистар MBR се смешта садржај који је прочитан из меморијске локације и који долази са излазних линија података меморије. У регистар MBR се смешта садржај који треба уписати у меморијску локацију. Садржај регистра MBR се води на улазне линије података меморије.

Са IR је означен прихватни регистар инструкције. У регистар IR се смешта садржај регистра MBR прочитан са меморијске локације чија је адреса одређена садржајем регистра РС. Бинарну реч у регистру IR треба интерпретирати сагласно формату инструкције (слика 1). Групе битова поља SRC1 и SRC2 треба водити у регистар MAR и користити као адресе меморијских локација са којих прочитане бинарне речи треба интерпретирати као два изворишна операнда. Прву прочитану бинарну реч треба из регистра MBR сместити у регистар А, а другу у регистар В. Регистри А и В представљају прихватне регистре изворишних операнда. Садржаји регистара А и В су везани на улазне линије података аритметичко логичке јединице ALU, док је садржај групе битова поља ОС регистра IR везан на управљачке линије ALU. На излазима ALU се појављује садржај који представља резултат операције специфициране пољем ОС инструкције над операндима чије су адресе у меморији специфициране пољима SRC1 и SRC2 инструкције. Овај садржај се уписује у регистар С, који представља прихватни регистар резултата. Садржај регистра С се води у регистар MBR, док се група битова DST регистра IR води у регистар MAR. Садржај регистра MAR се користи као адреса меморијске локације у коју се уписује садржај регистра MBR. Тиме се у меморијску локацију специфицирану пољем DST инструкције уписује резултат операције специфициране пољем ОС инструкције над операндима чије су адресе у меморији специфициране пољима SRC1 и SRC2 инструкције.

### **1.1.3 Улазно/излазни уређаји**

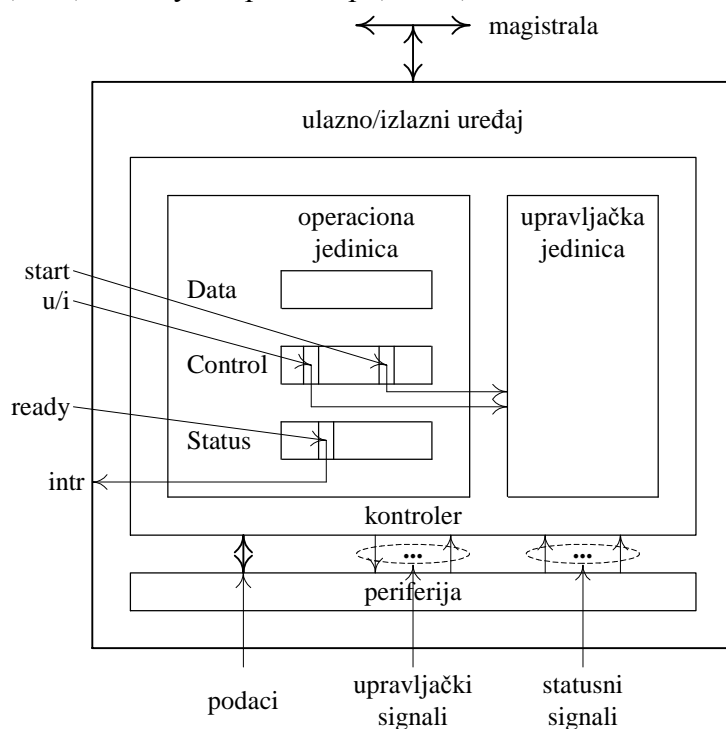
За унос бинарних речи у делове меморије са инструкцијама и операндима, користе се улазни уређаји. Они омогућавају да се на различите, корисницима рачунара погодне, начине специфицирају инструкције и операнди и пренесу у одговарајуће делове меморије као бинарне речи. Бинарне речи које представљају резултате решавањег проблема се помоћу излазних уређаја преносе из одговарајућих делова меморије и корисницима рачунара презентирају у некој погодној форми. Структура улазно/излазног уређаја је дата на слици 9.

Улазно/излазни уређај се састоји од улазно/излазне периферије и контролера. Периферија се реализује као стандардни уређај, на пример оптички читач, ласерски штампач, итд., који могу да се пребацују са једног рачунара на други рачунар. Периферије имају линије за пренос података у форми специфичној за дату периферију и управљачке и статусне линије. Ове линије се користе да се по протоколу дефинисаном за дату периферије подаци преносе од периферије или ка периферији.

За сваку периферију одређеног рачунара постоји посебан контролер. Он преко магистрале физички повезује дату периферију са меморијом и процесором као модулима рачунара и омогућује програмску контролу преноса података од периферије или ка периферији.

Физичко повезивање улазне периферије преко магистрале укључује узимање података из периферије сагласно протоколу рада са периферијом и њихово прослеђивање преко магистрале меморији или процесору као модулима рачунара сагласно протоколу размене података на магистрали. Физичко повезивање излазне периферије преко магистрале укључује узимање података из меморији или процесору као модулима рачунара сагласно протоколу размене података на магистрали и њихово слање у периферију сагласно протоколу рада са периферијом.

Програмска контрола рада улазно/излазне периферије укључује могућност да се извршавањем програма стартује и зауставља рад контролера, затим добијају информације о томе да ли је податак пренет из улазне периферије у контролер или да ли је податак пренет из контролера у излазну периферију, у на крају изврши пренос податка из контролера улазне периферије у меморију или процесор или из меморије или процесора у излазну периферију. Ради тога у контролеру постоје три регистра којима се приступа извршавањем инструкција процесора. То су управљачки регистар (Control), регистар податка (Data) и статусни регистар (Status).



Слика 9 Структура улазно/излазног уређаја

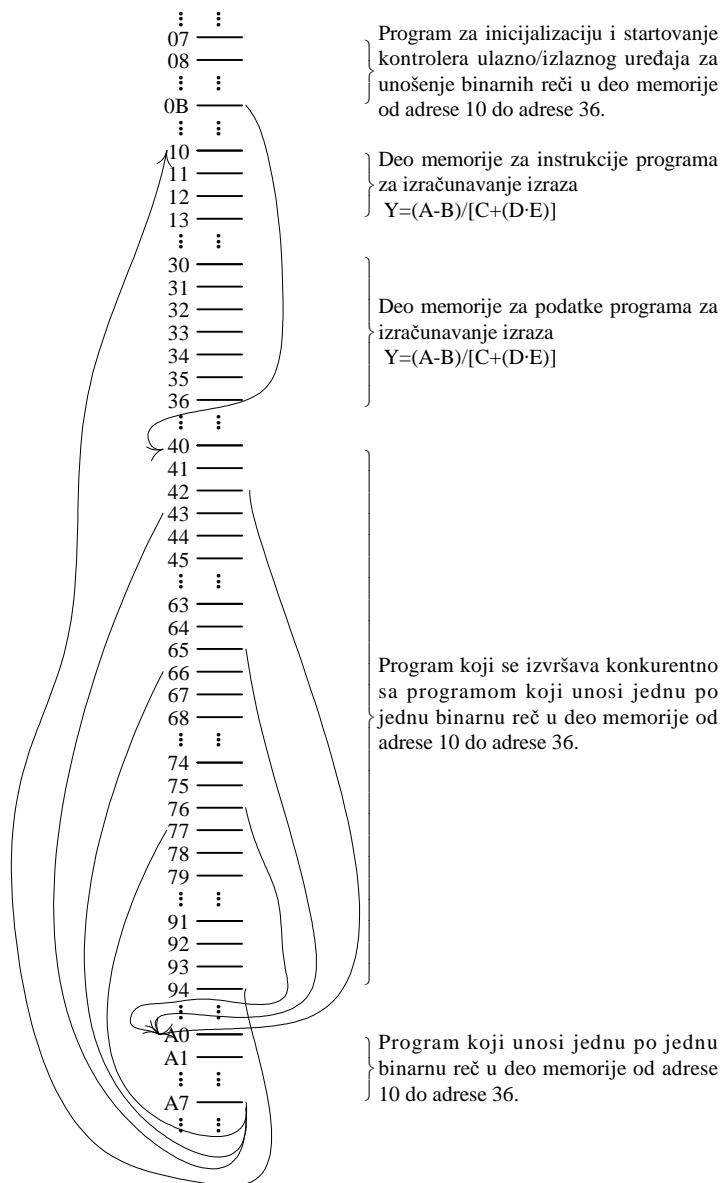
Управљачки регистар има, између осталог, и разред Start. Активна вредност овог разреда активира управљачку јединицу контролера да преноси податке из улазне периферије у контролер или из контролера у излазну периферију, док неактивна вредност зауставља управљачку јединицу.

Регистар податка служи да у њега управљачке јединица контролера смести податак пренет из улазне периферије. Статусни регистар има, између осталог, разред Ready. Активна вредност овог разреда је индикација да се у регистру податка налази податак пренет из улазне периферије и користи се за генерисање сигнала прекида *intr* који контролер шаље процесору. Сигнал прекида треба у процесору да изазове прекид у

извршавању текућег програма и прелазак на извршавање програма који се назива програм прекида. У оквиру овог програма треба да се пренесе садржај регистра податка у одговарајуће меморијску локацију или процесор и потом продужи са извршавањем прекинутог програма.

Регистар податка служи и да из њега управљачка јединица контролера пренесе податак у улазну периферију. Активна вредност разреда Ready статусног регистра је индикација да је податак из регистра податка пренет у излазне периферију и користи се за генерисање сигнала прекида  $intg$  који контролер шаље процесору. Сигнал прекида треба у процесору да изазове прекид у извршавању текућег програма и прелазак на извршавање програма који се назива програм прекида. У оквиру овог програма треба да се пренесе садржај одговарајуће меморијске локације или процесора у регистар податка и потом продужи са извршавањем прекинутог програма.

Уношење бинарних речи које представљају инструкције и податке програма са слике 3 у део меморије од адресе 10 до адресе 36, може се организовати као на слици 10.



Слика 10 Структура програма за уношење бинарних речи

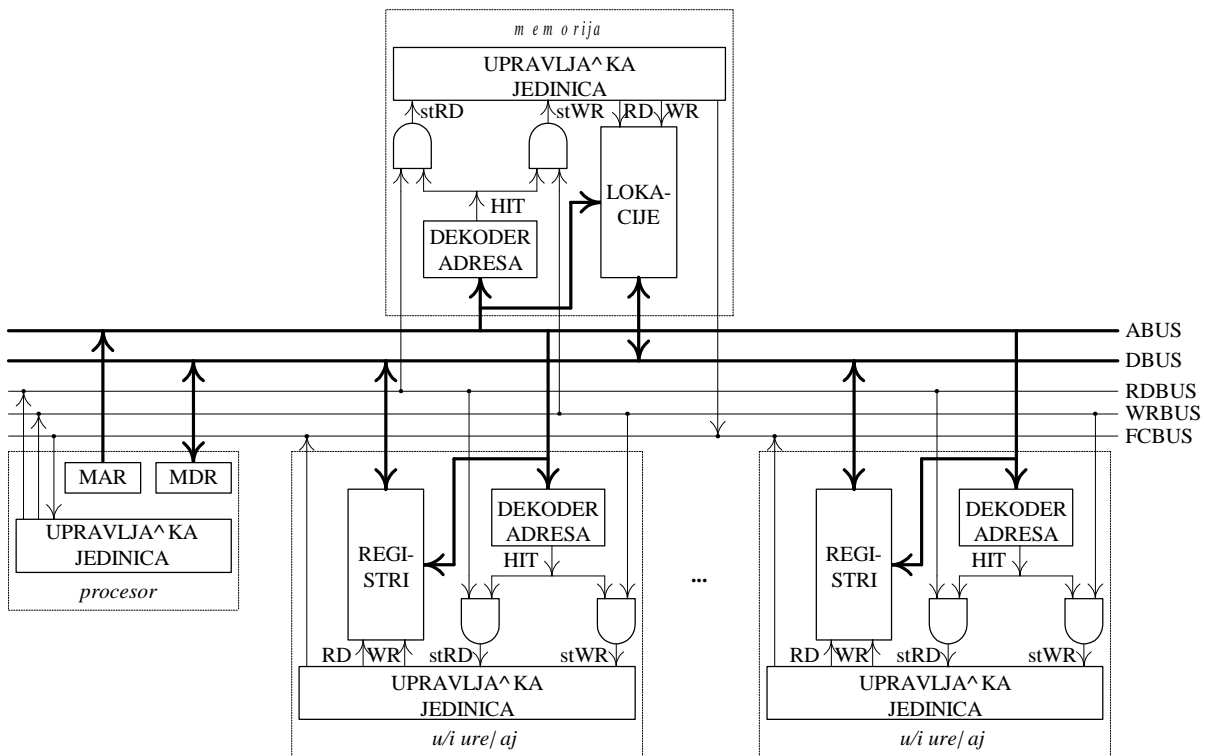


Најпре се програмом од адресе 07 до адресе 0В иницијализује и стартује контролер да се из улазно/излазног уређаја читају бинарне речи. Потом се прелази на извршавање програма од адресе 40 до адресе 94. За време извршавања овог програма контролер преноси једну по једну бинарну реч из периферије у регистар податка и после сваке пренете бинарне речи генерише прекид. По пријему сваког сигнала прекида процесор прекида извршавање овог програма и прелази на извршавање програма од адресе А0 до адресе А7. Овим програмом се преноси једна по једна бинарна реч из регистра податка у меморију редом од адресе 10 до адресе 36. После сваке пренете бинарне речи продужава се са извршавањем прекинутог програма. Узето је да је први прекид контролер генерисао за време извршавања инструкције са адресе 42, па се после ове инструкције прелази на програм од адресе А0 у оквиру кога се преноси бинарна реч из регистра податка контролера у меморију. Задњом инструкцијом овог програма са адресе А7 прелази се на инструкцију са адресе 43 прекинутог програма. Следећи прекид контролер генерише за време извршавања инструкције са адресе 65, па се после ове инструкције прелази на програм од адресе А0 у оквиру кога се преноси следећа бинарна реч из регистра податка контролера у прву следећу меморијску локацију после оне у коју је унета претходна бинарна реч. Задњом инструкцијом овог програма са адресе А7 прелази се сада на инструкцију са адресе 66 прекинутог програма. Слично се дешава и са прекидом који долази и за време извршавања инструкције са адресе 76, при чему се инструкцијом са адресе А7 прелази на инструкцију са адресе 77 прекинутог програма. Ово траје све док се програмом од адресе А0 до адресе А7 не пренесе и задња бинарна реч. Тада се у оквиру овог програма и зауставља контролер.

Овде је претпостављено да извршавање програма од адресе 40 до адресе 94 траје дужи од времена потребног да се извршавањем програма од адресе А0 до адресе А7, и то за сваку бинарну реч, пренесу све бинарне речи у део меморије од адресе 10 до адресе 36. Стога када се инструкцијом са адресе 94 пређе на инструкцију са адресе 10, у делу меморије од адресе 10 до адресе 13 биће бинарне речи које ће се интерпретирати као инструкције програма за израчунавање израза, а у делу меморије од адресе 30 до адресе 36 бинарне речи које ће се интерпретирати као подаци за програм од адресе 10 до адресе 13.

#### **1.1.4 Магистрала**

Магистрала је уређена група линија која служи за повезивање модула рачунарског система и то процесора, меморије и улазно/излазних уређаја (слика 11). Преко магистрале се преносе садржаји између регистара процесора, меморијских локација и регистара улазно/излазних уређаја. Цео ток преноса неког садржаја између два модула, назива се циклус на магистрали. Модул који започиње циклус на магистрали, назива се газда (*master*), а модул са којим газда реализује циклус, назива се слуга (*slave*). Газда може да буде процесор, док слуга може да буде меморија и улазно/излазни уређаји. Процесор чита садржаје меморијских локација и уписује садржаје у меморијске локације приликом читања инструкција и операнда и уписа резултата, као саставног дела извршавања инструкција неког програма. Процесор чита садржаје регистара уређаја и уписује садржаје у регистре уређаја приликом извршавања инструкција којима се добија статус уређаја, врши иницијализација уређаја, задаје режим рада, реализује стартовање и заустављање уређаја, и врши пренос података између меморије и уређаја и обратно.

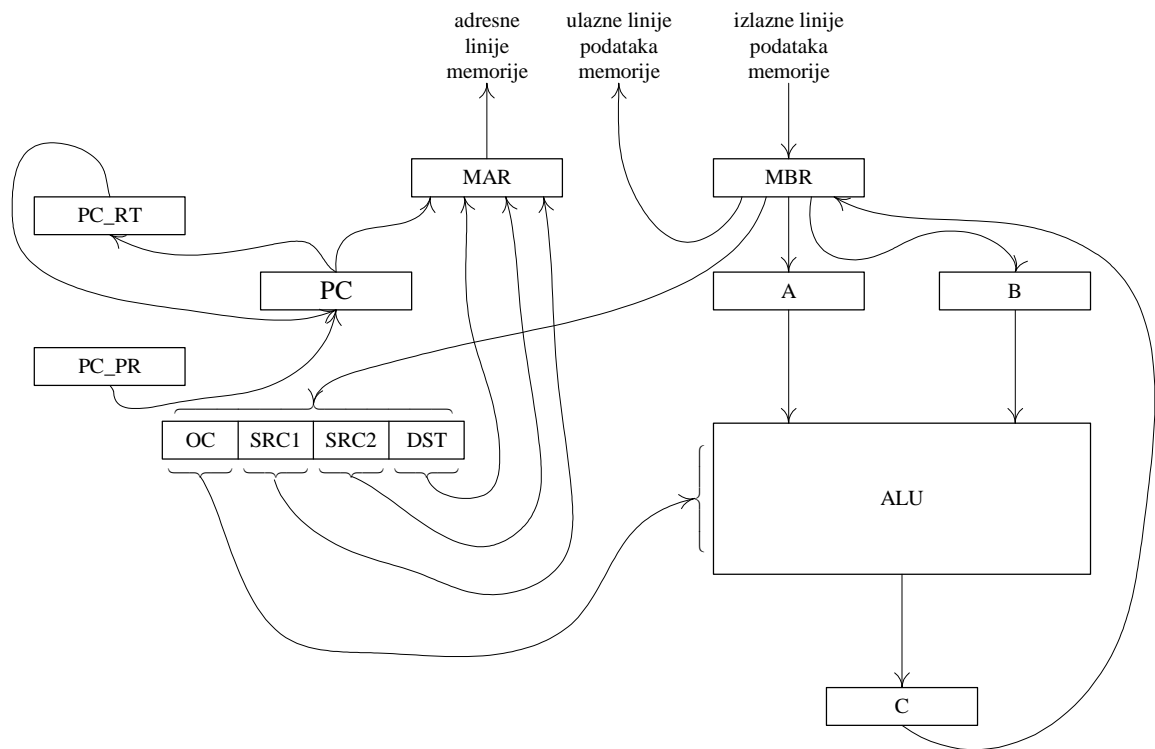


Слика 11 Адресирање на магистралаи са меморијски пресликаним улазно/излазним адресним простором

Магистралу чине три групе линија и то адресне линије (ABUS), линије података (DBUS) и управљачке линије (RDBUS, WRBUS и FCBUS). По адресним линијама газда шаље адресу меморијске локације или регистра уређаја приликом читања садржаја или уписа садржаја. По линијама података газда шаље садржај који треба да се упише у меморијску локацију или регистар уређаја чија адреса се налази на адресним линијама. По линијама података слуга шаље очитани садржај меморијске локације или регистра уређаја чија адреса се налази на адресним линијама. По управљачким линијама газда шаље сигнале којима одређује да ли треба да се реализује читање (RDBUS) или упис (WRBUS) садржаја. У неким реализацијама магистрале по управљачкој линији (FCBUS) и слуга шаље сигнал којим сигнализира газди да су читање или упис садржаја реализовани.

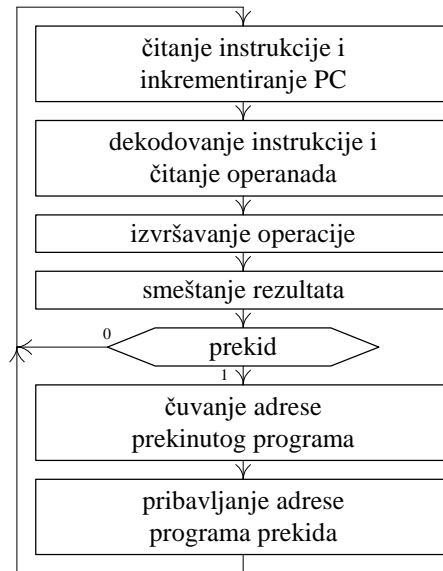
## 1.2 ФАЗЕ ИЗВРШАВАЊА ИНСТРУКЦИЈЕ

Да би се омогућило реаговање у процесору на сигнал прекида и скок на програм прекида, структуру процесора са слике 8 треба допунити са два регистра и то PC\_RT и PC\_PR, чиме се долази до структуре процесора са слике 12. Регистар PC\_RT треба да чува адресу повратка из програма прекида у прекинути програм, док регистар PC\_PR треба да садржи адресу програма прекида. На крају извршавања инструкције треба, уколико је током њеног извршавања генерисан прекид, пребацити садржај регистра PC у регистар PC\_RT и у регистар PC уписати садржај регистра PC\_PR. Тиме се обезбеђује да када се сада садржај регистра PC користи као адреса са које ће се читати инструкција, то буде прва инструкција програма прекида. Тиме се прелази на програм прекида. Поред тога, на крају програма прекида мора увек да буде посебна инструкција која пребацује садржај регистра PC\_RT у регистар PC. Тиме се обезбеђује да када се сада садржај регистра PC користи као адреса са које ће се читати инструкција, то буде инструкција прекинутог програма која би се извршавала да није било прекида и преласка на програм прекида.



Слика 12 Структура процесора са реакцијом на прекид

На основу свега изнетог извршавање инструкције се састоји из следећих фаза: читање инструкције и инкрементирање регистра PC, декодовање инструкције и читање операнда, извршавање операције, смештање резултата, чување адресе прекинутог програма и прибављање адресе програма прекида (слика 13). Кроз фазе чување адресе прекинутог програма и прибављање адресе програма прекида се пролази само уколико је током проласка кроз претходне фазе генерисан прекид.



Слика 13 Дијаграм тока извршавања инструкције

Током извршавања фазе читање инструкције и инкрементирање регистра PC најпре се садржај регистра PC пребацује у регистар MAR и врши инкрементирање садржаја регистра PC, затим се садржај меморијске локације чија је адреса одређена садржајем

регистра MAR уписује у регистар MBR и на крају садржај регистра MBR уписује у регистар IR.

Током извршавања фазе декодовање инструкције и читање операнда најпре се група битова из поља SRC1 регистра IR пребацује у регистар MAR, затим се садржај меморијске локације чија је адреса одређена садржајем регистра MAR уписује у регистар MBR и на крају садржај регистра MBR уписује у регистар A. Ово се понавља тако што се група битова из поља SRC2 регистра IR пребацује у регистар MAR, затим се садржај меморијске локације чија је адреса одређена садржајем регистра MAR уписује у регистар MBR и на крају садржај регистра MBR уписује у регистар B.

Током извршавања фазе извршавања инструкције се над садржајима регистара A и B у аритметичко логичкој јединици ALU реализује операција одређена групом битова из поља OC регистра IR и резултат уписује у регистар C.

Током извршавања фазе смештање резултата најпре се група битова из поља DST регистра IR пребацује у регистар MAR, затим садржај регистра C пребацује у регистар MBR и на крају садржај регистра MBR уписује у меморијску локацију чија је адреса одређена садржајем регистра MAR.

Уколико током претходних фаза није генерисан прекид прелази се на фазу читање инструкције и инкрементирање регистра PC. Тиме се завршава извршавања текуће инструкције и прелази на извршавања прве следеће инструкције у програму. Уколико је током претходних фаза генерисан прекид прелази се на фазе чување адресе прекинутог програма и прибављање адресе програма прекида.

Током извршавања фазе чување адресе прекинутог програма, садржај регистра PC се пребацује у регистар PC\_RT.

Током извршавања фазе прибављање адресе програма прекида, садржај регистра PC\_PR се пребацује у регистар PC и прелази се на фазу читање инструкције и инкрементирање регистра PC. Тиме се завршава извршавања текуће инструкције и прелази на извршавања прве инструкције програма прекида.

### **1.3 АРХИТЕКТУРА И ОРГАНИЗАЦИЈА РАЧУНАРА**

Код разматрања рачунарских система користе се појмови архитектуре рачунара и организације рачунара. Под архитектуром рачунара се подразумева све оно што треба да се зна о рачунару да би за њега могли да се напишу програми који ће се успешно извршавати и увек давати идентичне резултате без обзира на то како је рачунар реализован. Под организацијом рачунара се подразумевају различити начини реализације рачунара.

Елементи архитектуре рачунара су програмски доступни регистри, меморијске локације и регистри контролера периферија, типови података, формати инструкција, начини адресирања, скуп инструкција и механизам прекида.

Елементи организације рачунара су могуће технике реализације модула рачунара, као на пример реализације операционих јединица са директним везама, једном или више интерних магистрала, реализације управљачких јединица неком од ожичених техника, микропрограмских техника или техником преклапања различитих фаза већег броја инструкција, реализације магистрала одређених ширина, коришћење или не коришћење кеш меморија, различите реализације кеш меморија итд.