

Основи рачунарске технике 2

Испит – Л4

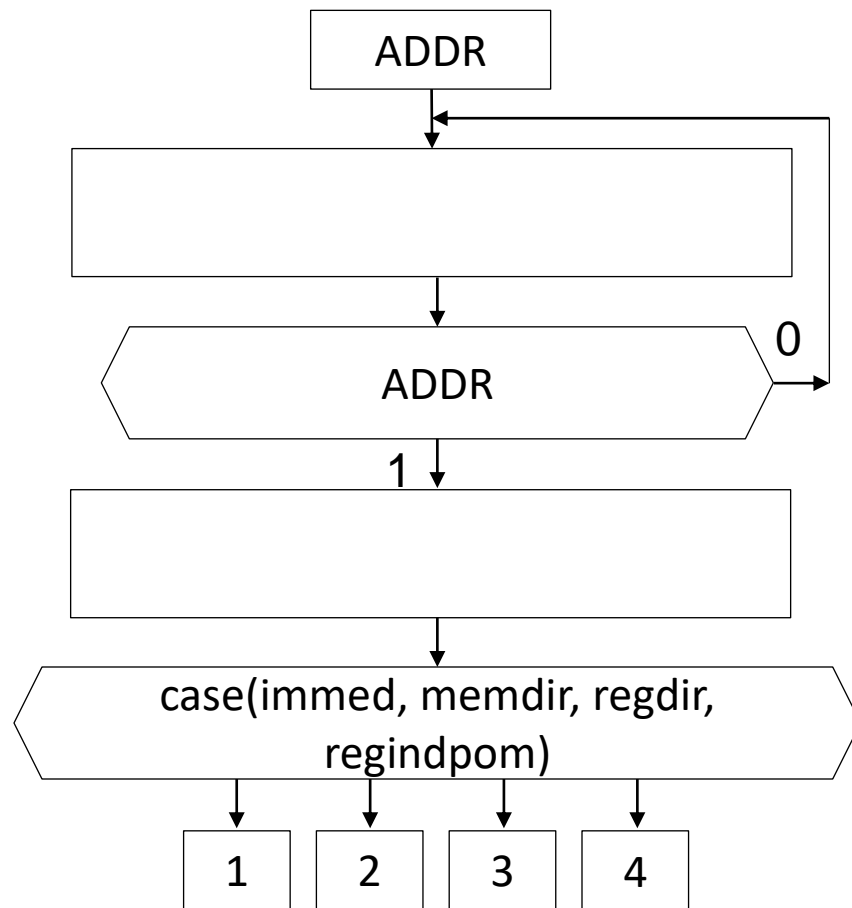
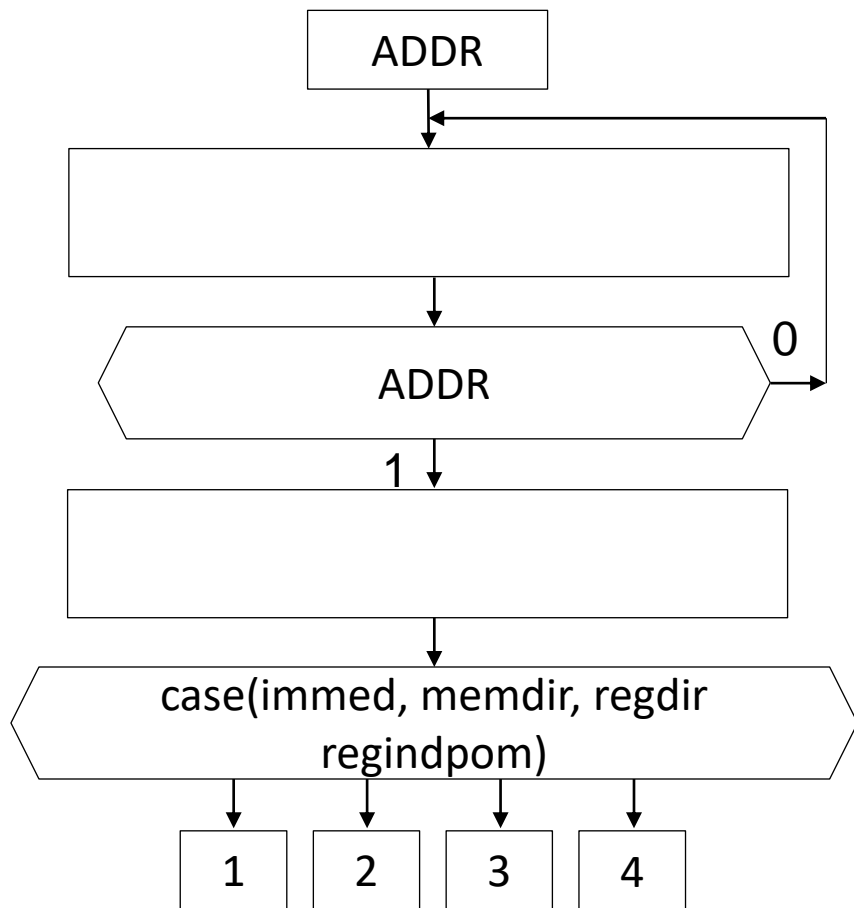
Основи рачунарске технике 2

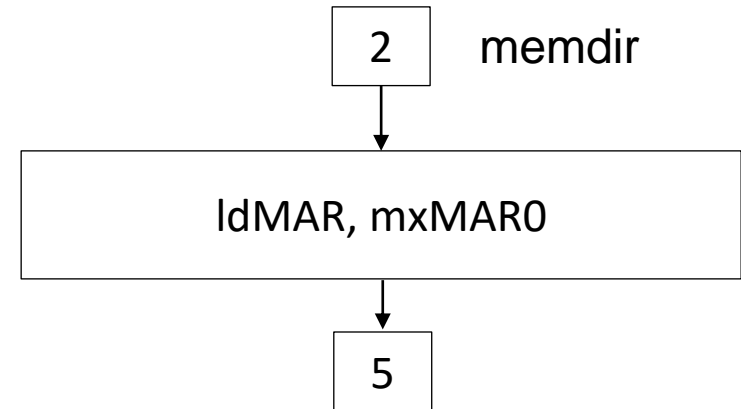
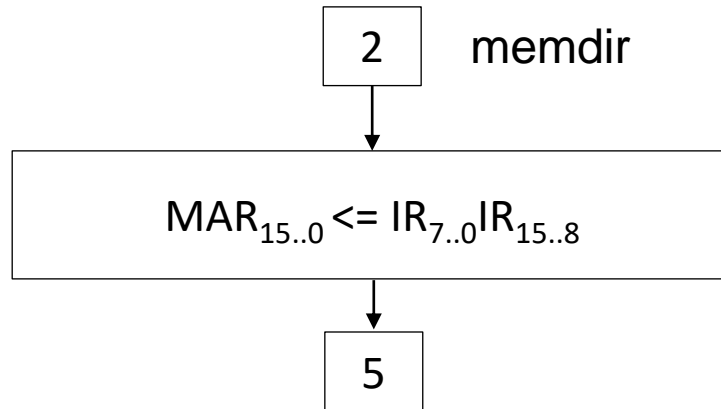
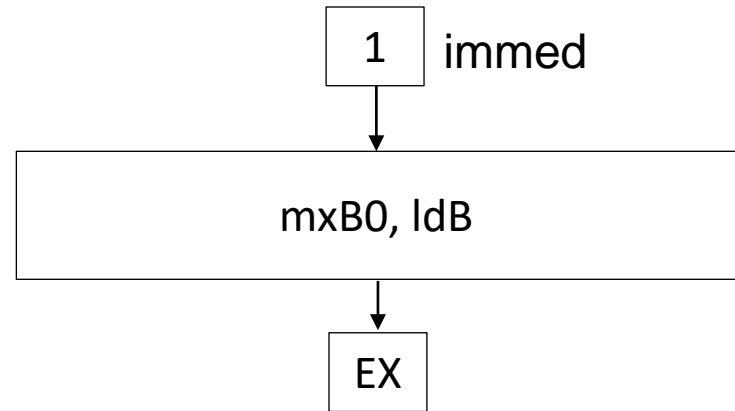
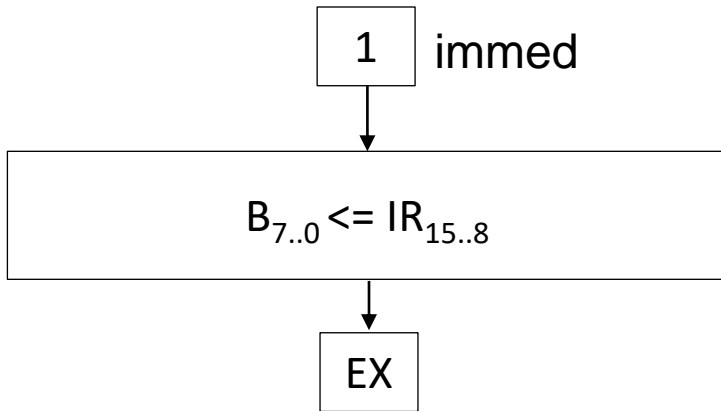
ADDR блок

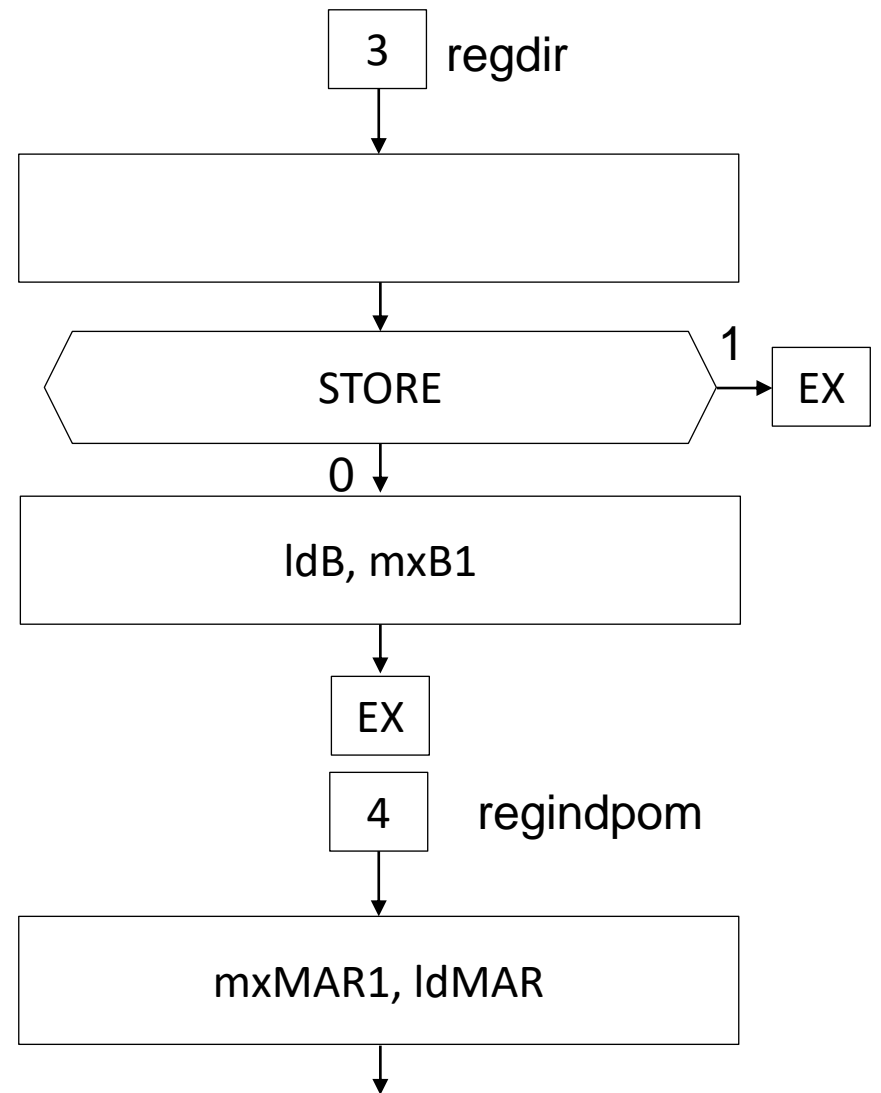
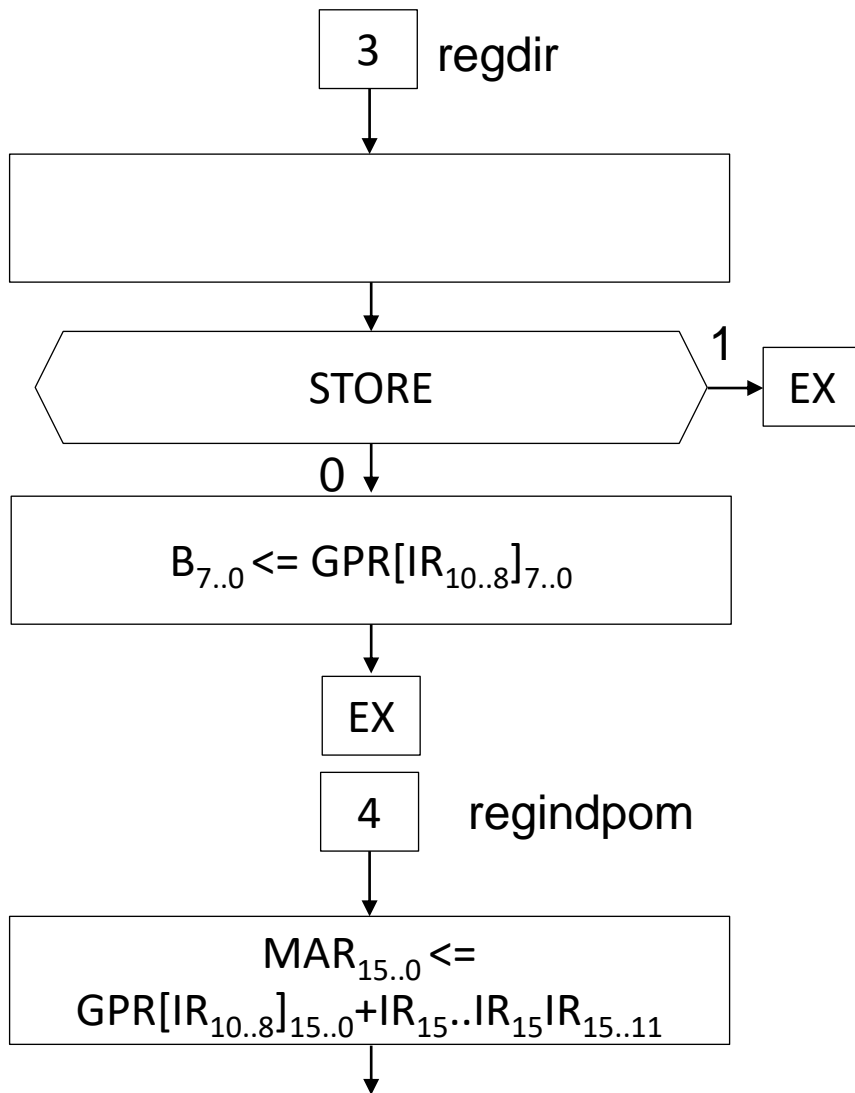
Инструкција	IR _{23..19}	Инструкција	IR _{23..16}	IR _{15..8}	IR _{7..0}	Дужина
LD	0011 0b	JSR	0010 0001b	млађи бајт	старији бајт	3B
ST	0011 1b	JMP	0010 0000b	млађи бајт	старији бајт	3B
ADD	0100 0b	JLEQ	0001 0010b	млађи бајт	старији бајт	3B
SUB	0100 1b	BGREU	0001 0001b	померај	/	2B
		BLSS	0001 0000b	померај	/	2B
		POPGPR	0000 0100b	/	/	1B
		PUSHGPR	0000 0011b	/	/	1B
		RTI	0000 0010b	/	/	1B
		RTS	0000 0001b	/	/	1B
		HALT	0000 0000b	/	/	1B

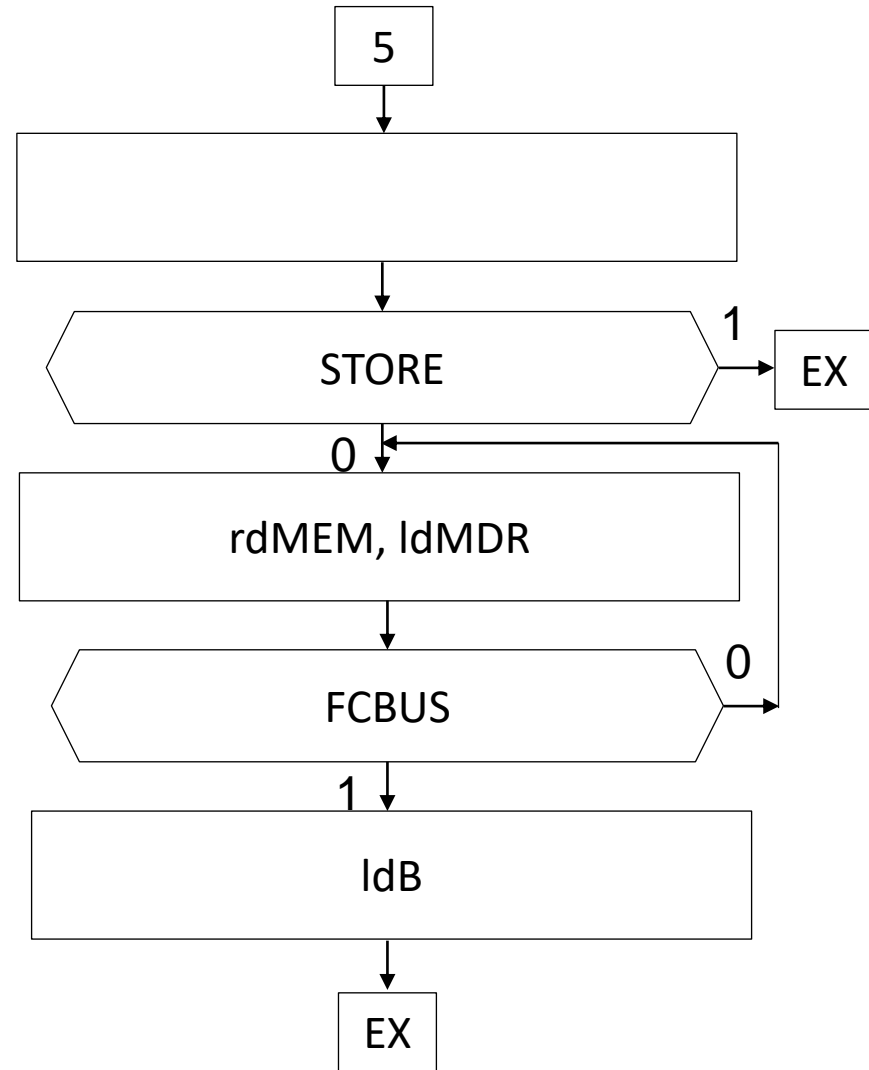
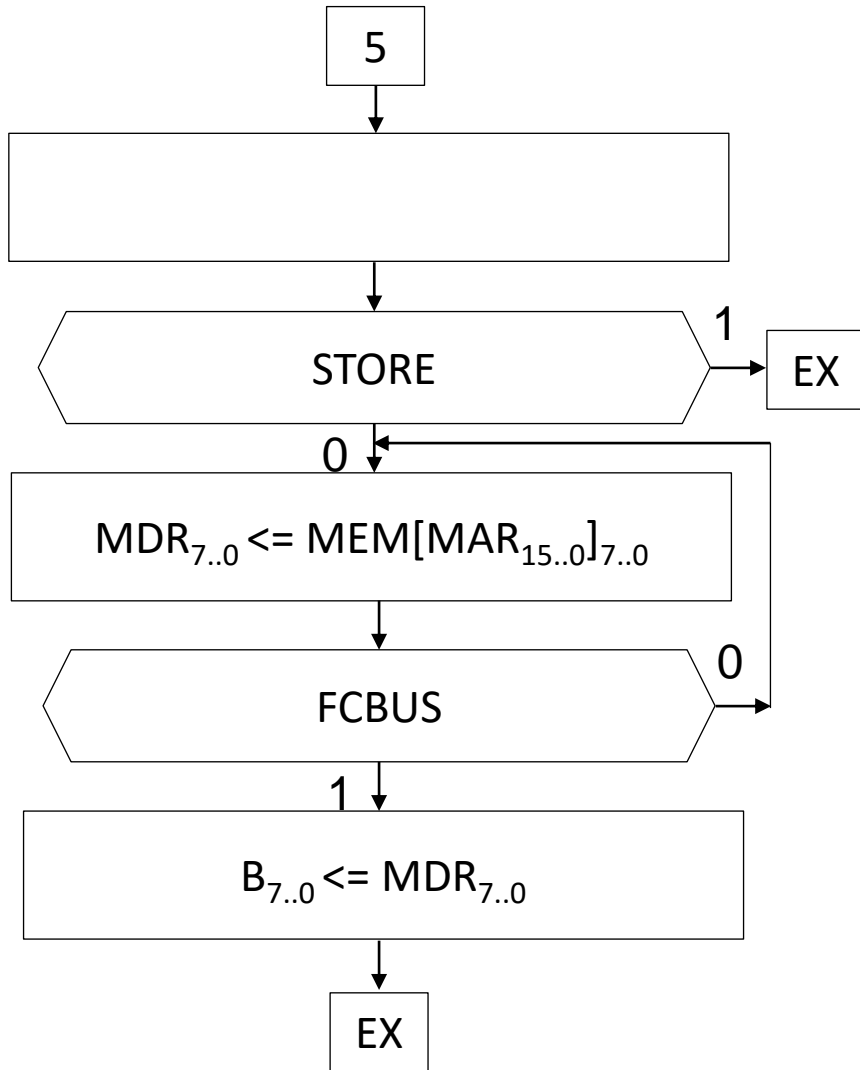
Адресирања	IR _{18..16}	IR _{15..8}	IR _{7..0}	Дужина
regindpom	011b	PPPP PRRRb	/	2B
regdir	010b	XXXX XRRRb	/	2B
memdir	001b	млађи бајт	старији бајт	3B
immed	000b	податак	/	2B

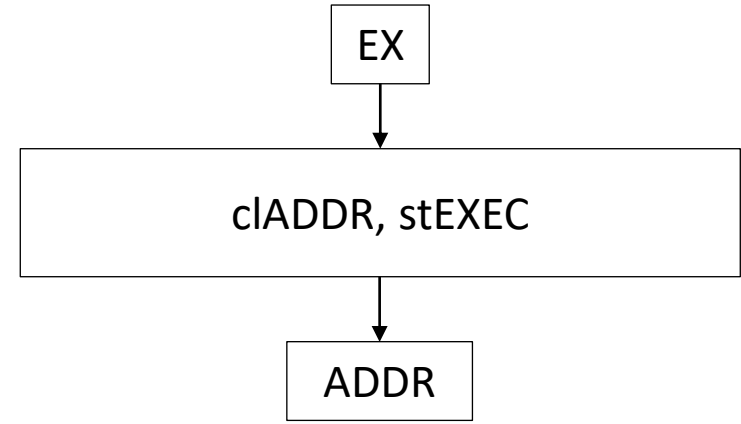
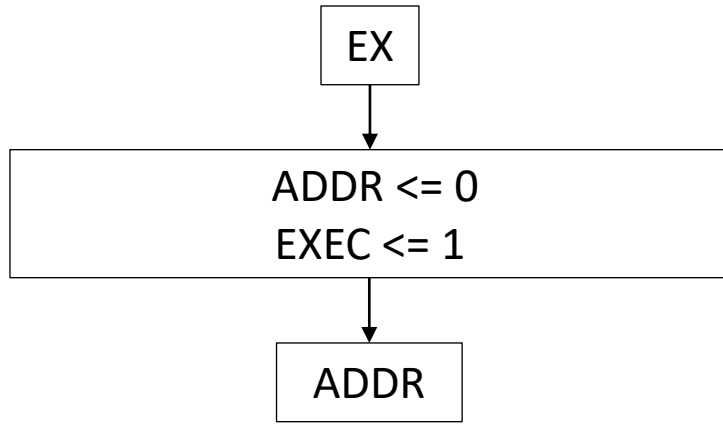
	Нижа адреса	Виша адреса
Адреса	млађи	старији



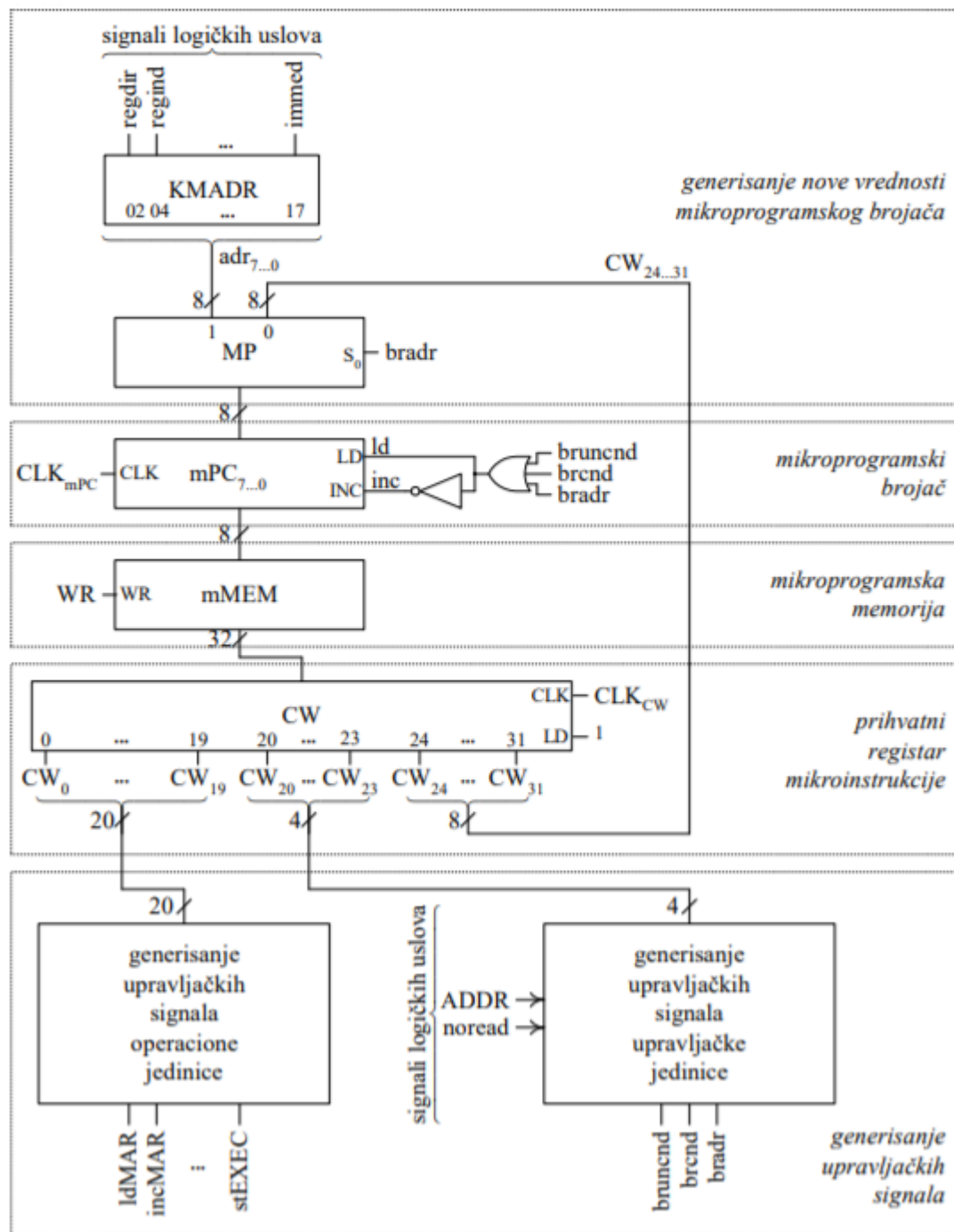






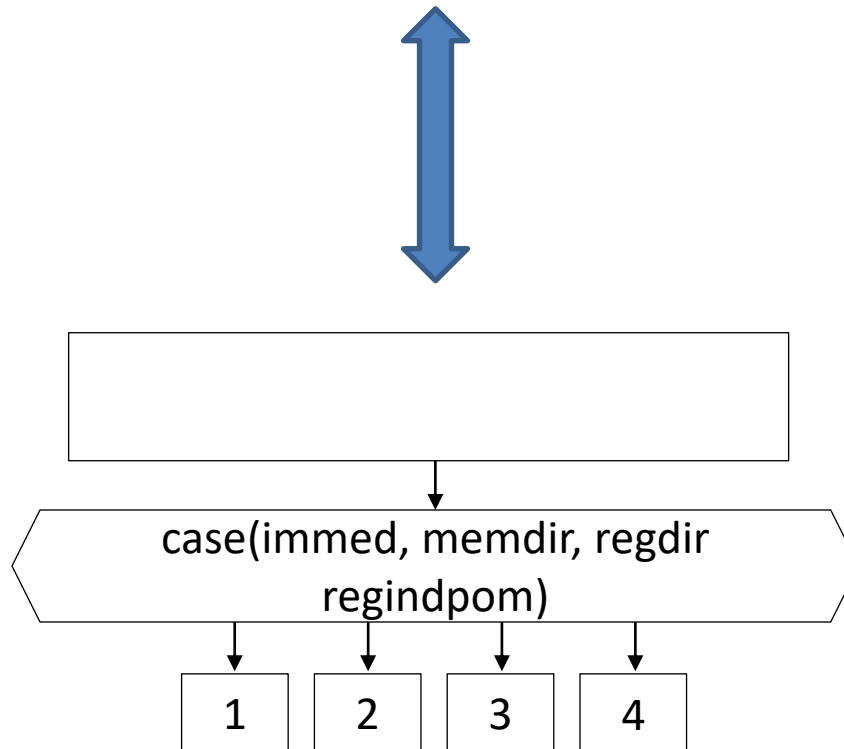


Управљачка јединица



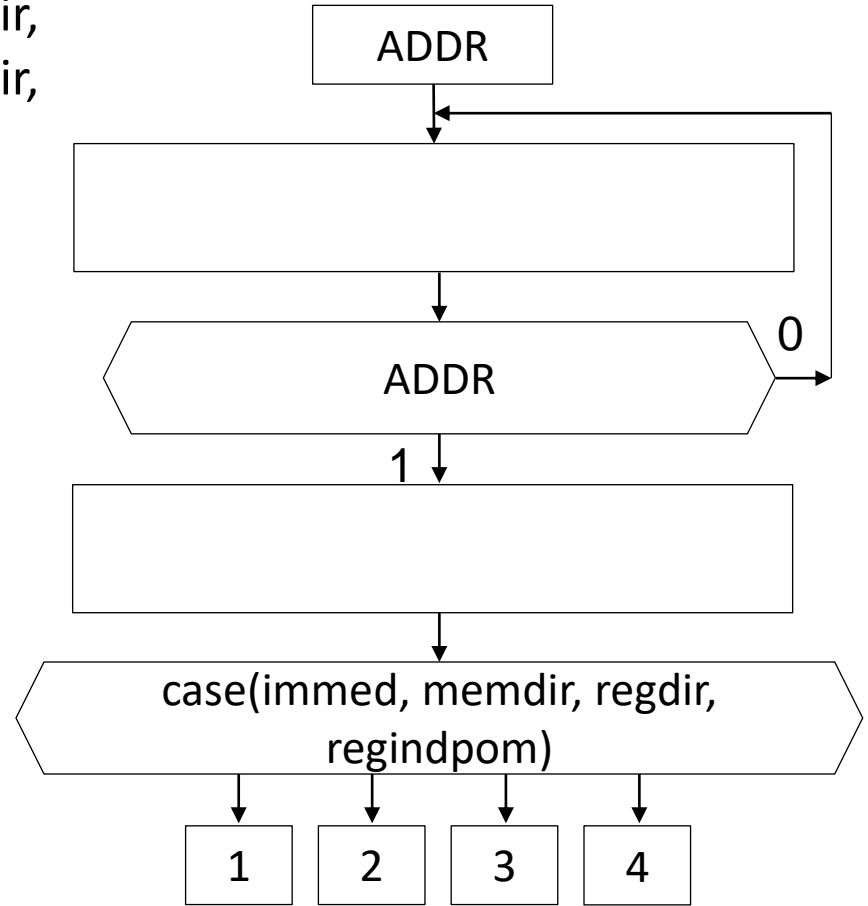
Секвенца управљачких сигнала

$\text{step}_{xx} \text{br}(\text{case}(\text{immed}, \text{memdir}, \text{regdir}, \text{regindpom}) \text{then}(\text{immed}, \mathbf{step}),$
 $(\text{memdir}, \mathbf{step}), (\text{regdir}, \mathbf{step}), (\text{regindpom}, \mathbf{step}))$



step00 br(if notADDR **step00**)

step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step**), (memdir,
step),(regdir, **step**),(regindpom, **step**))



step00 br(if notADDR **step00**)

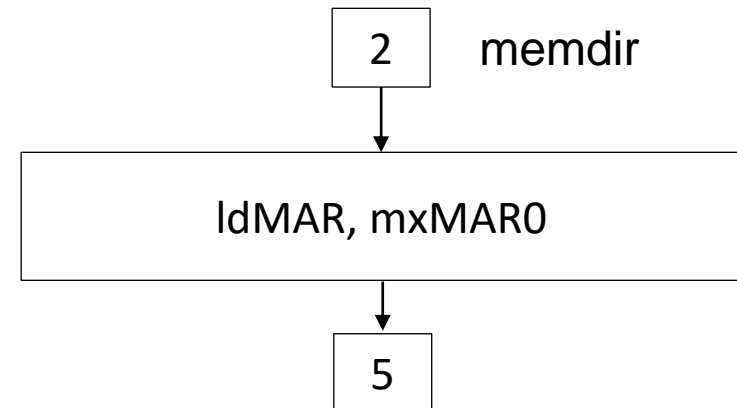
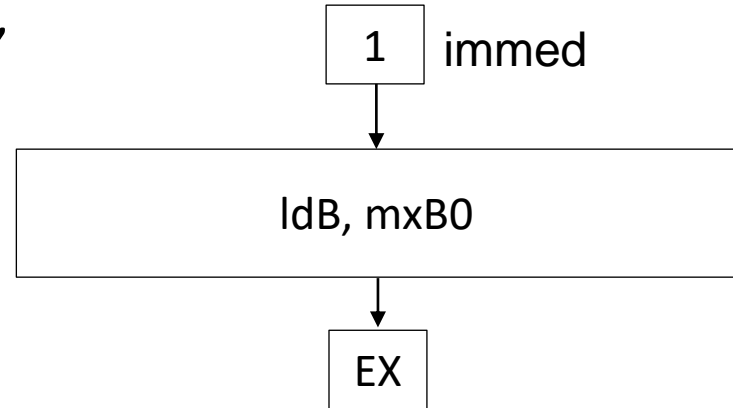
step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step02**),
(memdir, **step03**),(regdir,
step),(regindpom, **step**))

//-----IMMED-----

step02 ldB, mxB0, br **step**

//-----MEMDIR-----

step03 ldMAR, mxMAR0, br **step**



step00 br(if notADDR **step00**)

step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step02**),
(memdir, **step03**),(regdir,
step04),(regindpom, **step06**))

//-----IMMED-----

step02 ldB, mxB0, br **step**

//-----MEMDIR-----

step03 ldMAR, mxMAR0, br **step**

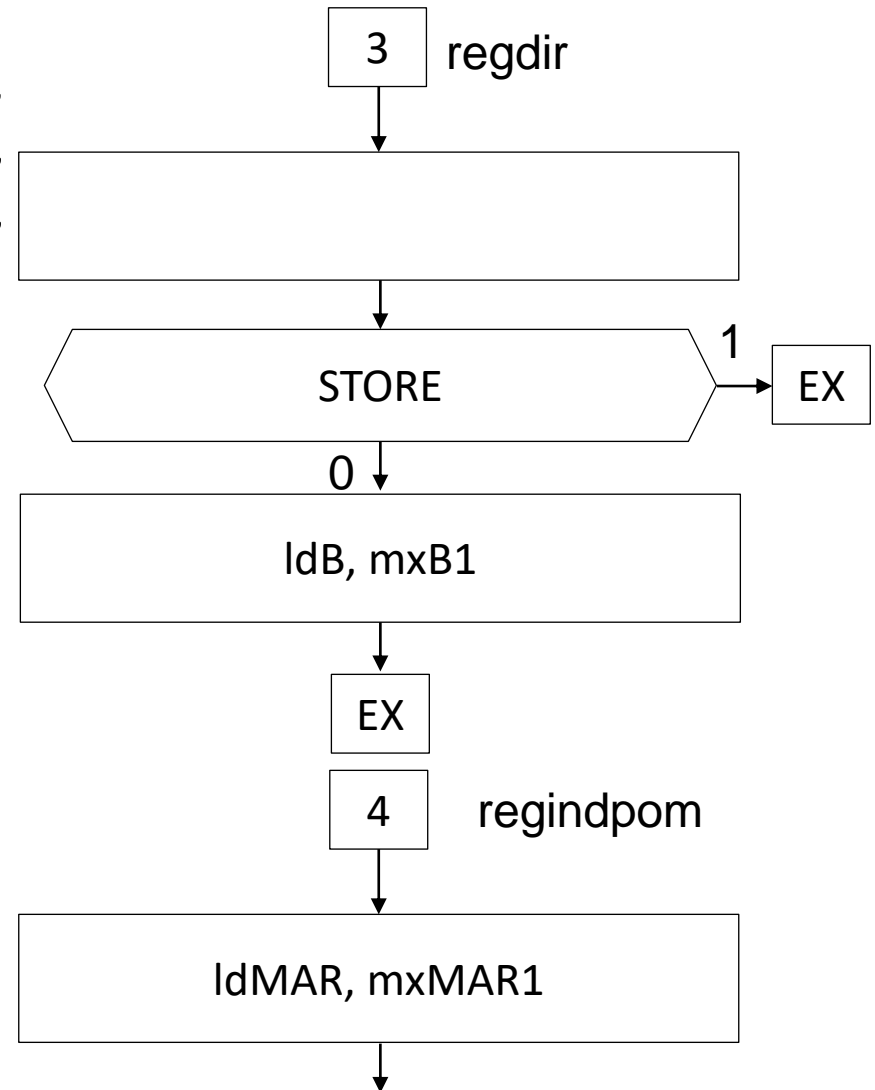
//-----REGDIR-----

step04 br(if STORE then **step**)

step05 ldB, mxB1, br **step**

//-----REGINDPOM-----

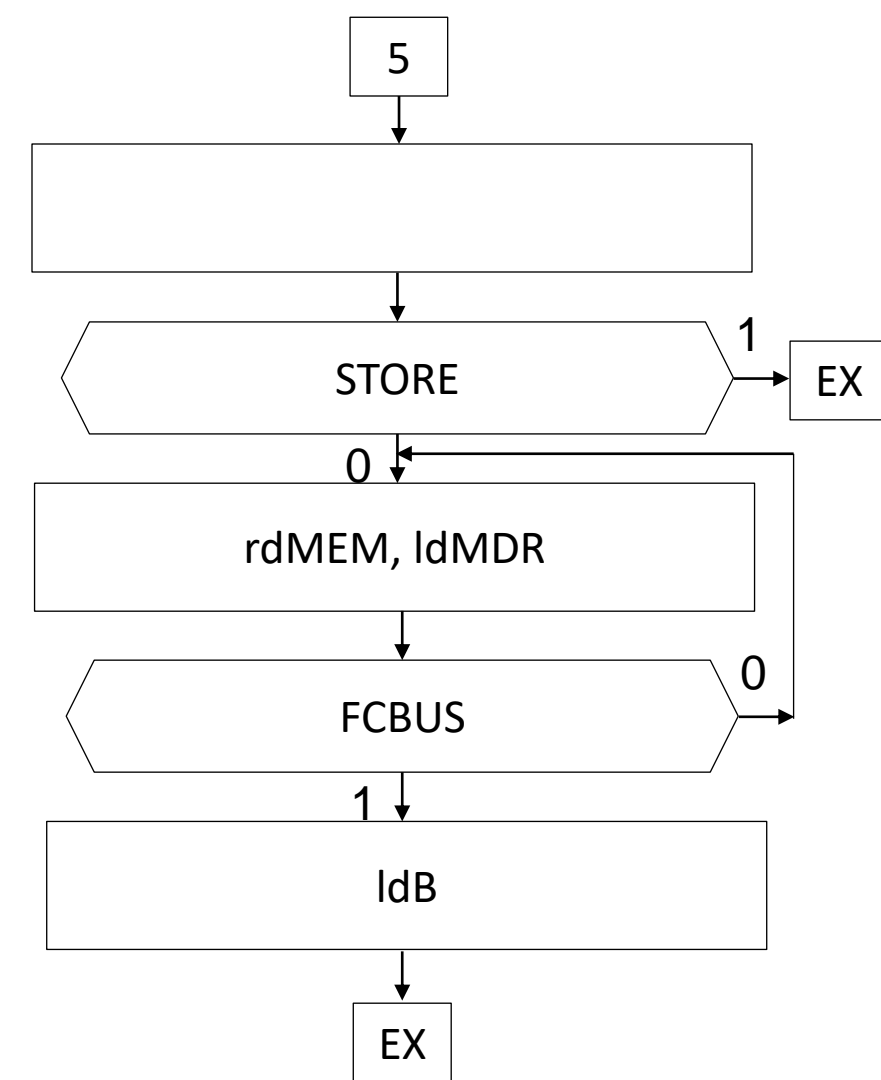
step06 ldMAR, mxMAR1



```

step00 br(if notADDR step00)
step01 br(case(immed, memdir, regdir,
regindpom)then(immed, step02),
(memdir, step03),(regdir,
step04),(regindpom, step06))
//-----IMMED-----
step02 ldB, mxB0, br step
//-----MEMDIR-----
step03 ldMAR, mxMAR0, br step07
//-----REGDIR-----
step04 br(if STORE then step)
step05 ldB, mxB1, br step
//-----REGINDPOM-----
step06 ldMAR, mxMAR1
//-----
step07 br(if STORE then step)
step08 rdMEM, ldMDR,
br(if notFCBUS then step08)

```



step09 ldB

step00 br(if notADDR **step00**)

step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step02**),
(memdir, **step03**),(regdir,
step04),(regindpom, **step06**))

//-----IMMED-----

step02 ldB, mxB0, br **step0A**

//-----MEMDIR-----

step03 ldMAR, mxMAR0, br **step07**

//-----REGDIR-----

step04 br(if STORE then **step0A**)

step05 ldB, mxB1, br **step0A**

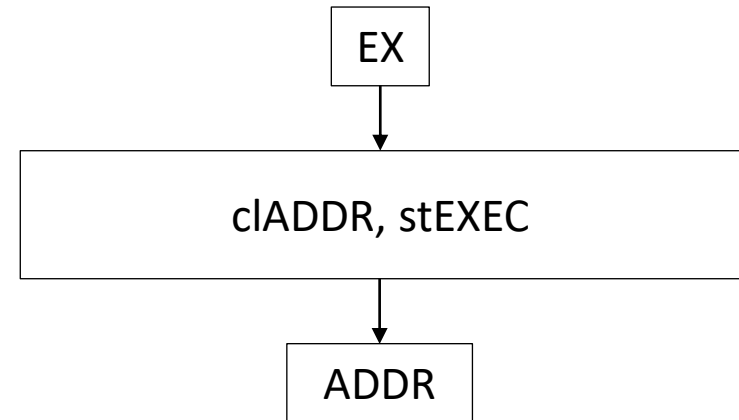
//-----REGINDPOM-----

step06 ldMAR, mxMAR1

//-----

step07 br(if STORE then **step0A**)

step08 rdMEM, ldMDR, br(if notFCBUS
then **step08**)



step09 ldB

step0A clADDR, stEXEC, br **step00**

step00 br(if notADDR **step00**)

step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step02**),
(memdir, **step03**),(regdir,
step04),(regindpom, **step06**))

//-----IMMED-----

step02 ldB, mxB0, br **step0A**

//-----MEMDIR-----

step03 ldMAR, mxMAR0, br **step07**

//-----REGDIR-----

step04 br(if STORE then **step0A**)

step05 ldB, mxB1, br **step0A**

//-----REGINDPOM-----

step06 ldMAR, mxMAR1

//-----

step07 br(if STORE then **step0A**)

step08 rdMEM, ldMDR, br(if notFCBUS
then **step08**)

0	1	2	3
ldB	mxB0	ldMAR	mxMAR0

4	5	6	7
mxB1	mxMAR1	rdMEM	ldMDR

8	9	10	11

12	13	14	15

16	17	18	19

20	21	22	23

step09 ldB
step0A clADDR, stEXEC, br **step00**

0	1	2	3
ldB	mxB0	ldMAR	mxMAR0

4	5	6	7
mxB1	mxMAR1	rdMEM	ldMDR

8	9	10	11
clADDR	stEXEC	/	/

12	13	14	15
/	/	/	/

16	17	18	19

20	21	22	23

Сигнал безусловног скока	CC
bruncnd	1

Сигнал условног скока	CC	Сигнал услова
brnotADDR	2	$\overline{\text{ADDR}}$
brSTORE	3	STORE
brnotFCBUS	4	$\overline{\text{FCBUS}}$

Сигнал вишеструког условног скока	CC
bradr	5

Услов	Корак
immed	2
memdir	3
regdir	4
regindpom	6

0	1	2	3
ldB	mxB0	ldMAR	mxMAR0

4	5	6	7
mxB1	mxMAR1	rdMEM	ldMDR

8	9	10	11
clADDR	stEXEC	/	/

12	13	14	15
/	/	/	/

16	17	18	19
cc			

20	21...	...26	27
ba			

Микропрограм

step00 br(if notADDR **step00**)

step01 br(case(immed, memdir, regdir,
regindpom)then(immed, **step02**),
(memdir, **step03**),(regdir,
step04),(regindpom, **step06**))

//-----IMMED-----

step02 ldB, mxB0, br **step0A**

//-----MEMDIR-----

step03 ldMAR, mxMAR0, br **step07**

//-----REGDIR-----

step04 br(if STORE then **step0A**)

step05 ldB, mxB1, br **step0A**

//-----REGINDPOM-----

step06 ldMAR, mxMAR1

//-----

step07 br(if STORE then **step0A**)

step08 rdMEM, ldMDR, br(if
notFCBUS then **step08**)

step09 ldB

step0A clADDR, stEXEC, br **step00**