

## TIPOVI ZADATAKA SORTIRANO PO PREDUHAJIMA

- ① VON NEUMANN - STAVANJE RAZLIČITIH MODULA, NA  
16 BIT I 8 BIT SABIRNICU, UZ TEŽENJE  
NET. PROSTOR
- ② PREDPOSTAVLJENI CISC - SKICIRATI STANJE MEMORIJE, STANJE  
NA SABIRNICAMA, STANJE U REG.  
NAKON SVAKE OPERACIJE
- ③ CU-SKLOPOVSKI - INstrukcijski PROCESOR, TREBA  
PREINACITI, OPISATI PREINAKE, NAPISATI SVE  
JEDNAŽBE, SKICIRATI KOMBINAC. SKLOP.
- ④ CU-MIKROPROGRANSKI - ZA ZADANI NUDZ, TE FURNAT RJEČI  
SKICIRATI ICH, ZA ZADANU INSTRUKCIJU  
ZA NEKI KOD SKICIRAJ USER STACK ICI  
ZA ZADANI SLUČAJ KOD NC 6300 I  
PREKID SKICIRAJ USER I SYSTEM STACK
- ⑤ VEZA SA PROG. (STOGOVI) - ZA NEKI KOD SKICIRAJ USER STACK ICI  
ZA ZADANI SLUČAJ KOD NC 6300 I  
PREKID SKICIRAJ USER I SYSTEM STACK
- ⑥ PRIRUČNE MEMORJE (CACHE) - ZADANA MEMORIJA (a, b, z, m) TE ODREDI  
STRUKTURU ADRESE, ODREDI POLOŽAJ  
NEKE ADRESE (INDEXS)

# ① TIPI - ZADACI SA SPRAJANJEI ELEMENATA NA 102. CAR,

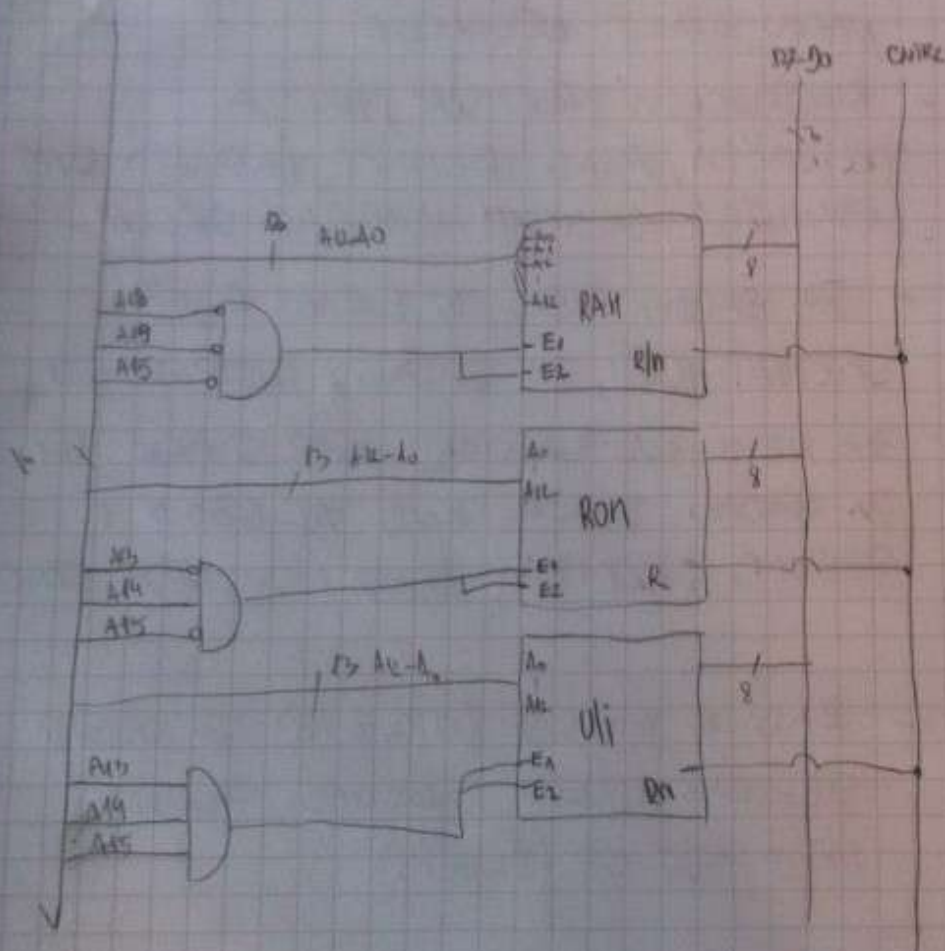
ZADATOK

11.11.2012/2013

ZADANA SU DVA MODULA VELICINE 3182 BAJTA. ZENATOST JE BAJNA. (RAM, ROM)

TAKODER IMAMO I JEDAN ULI MODUL. SA 4 REGISTRA ZA R.W. Svi imaju 2E  
 EN OD ADRESA 0. Uli OD SRADINE, TE ROM PR. KODU.

- 1<sup>o</sup> UNIKZ PRVO HACERAI KULICE TEDIKA; CONKRE - DENO; DATA → DATA ULAZE, ULE → EN
  - 2<sup>o</sup> VELICINA MODULA =  $2^m \rightarrow m$  JE GLAVNA NISOBANU ADRESNIH BAJTA (ODAB) → HANSTRA ADE
  - 3<sup>o</sup> VIDJETI KJE SU BAJTNI ADRESSE, NVE RAM OD 0 DO? PA OD SVEJ VELICINA!
- POKSIRANI BITONI MO NA ⑤, DUK GUSANU BITONI MO NA ADZ



-imamo 3182 B  $\rightarrow 2^{15}$

RAM - mi zelimo da  
 on bude EAGLES kada  
 000 ako bude bilo koji  
 jedinica  $\rightarrow$  gori

dalle zelimo 1D

ROM - zelimo 010

dalle  $\rightarrow$   $\begin{matrix} 24 \\ 9 \\ 1 \\ 0 \end{matrix}$

ULAZ	ELAZ
X	1
X	1
X	1

RAM: od adrese 0000 0000 0000 0000  
 do adrese 000,1 1111 1111 1111 } 2<sup>15</sup> lokacija

ULI: od adrese 0100 0000 0000 0000  
 do adrese 010,1 1111 1111 1111 }

ROM: od adrese 1110 0000 0000 0000  
 do adrese 111,1 1111 1111 1111 }

Ovo nam govori da se zalupih 13  
 bita "službeno" mišljenja a prva tri bita  
 treba "fiksirati" za svaki modul

FIKSIRANI  $\Rightarrow$  NA ELAZKE  
 SLOBODNI  $\Rightarrow$  NA ADZ

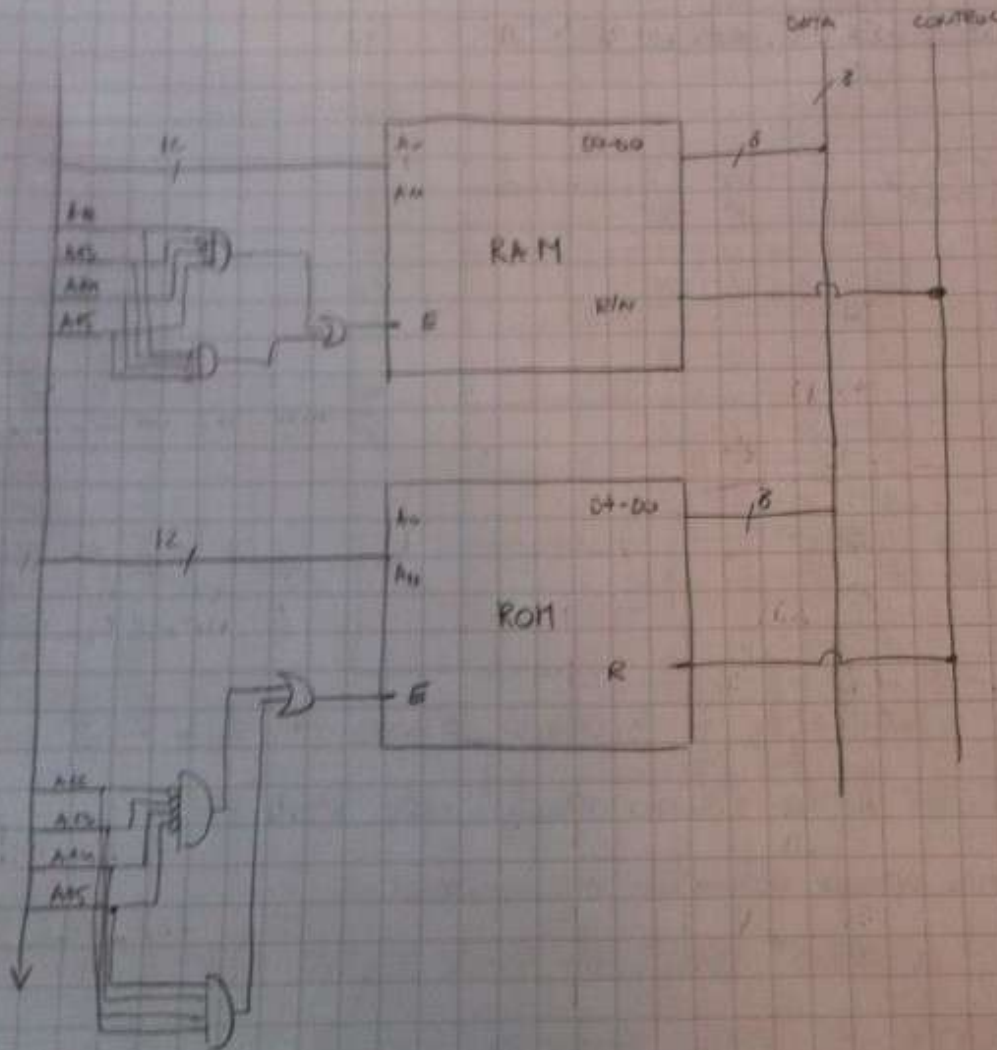


ZADATK 2

ZADATI ROM (8192B), TE RAM (65536B). OBA IMAJU JEDNOJE E.  
 ROM: 11 AB 16BITA; 3 BITA.

TRAŽI SE: ROM  $\Rightarrow$  0000 - 0FFF te 0000 - FFFF  
 RAM  $\Rightarrow$  1000 - EFFF

- (1) NACRTAJ KIRKED; DOKAZI ČITANJE I PISANJE ROM, DATA I A D+DU
- (2) ROM = 8192  $\times$  16 BIT = 131072, RAM = 65536  $\times$  16 BIT = 1048576



ROM od 0000 0000 0000 0000 (0000)  
 do 0000 1111 1111 1111 (0FFF)  
 + od 1111 0000 0000 0000 (F000)  
 do 1111 1111 1111 1111 (FFFF)

želim A15-A12 0 ili 1

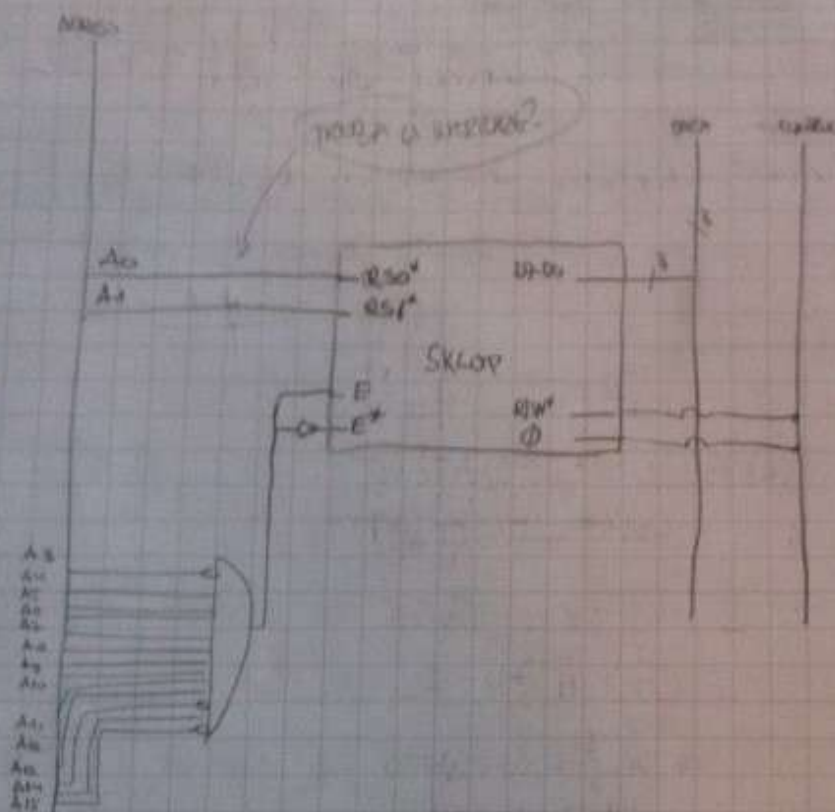
ULAZ	IZLAZ
0	1
0	1
0	1
0	1

RAM od 0001 0000 0000 0000  
 do 1110 1111 1111 1111

## 12. PREDAVANJE

### ZADANJE

IZRAČUNATI, KAKO JE VEČA INDEKSI NA ŽOLU OBRATITI R0-R3  
TRAJER IMA R0, R1, E, R\*, D3-D4, R1W, R\*, D. SKUP SE JOKA AN 1000  
KORISTI TUDI ADRESA DEKODIRANJE TE SA DOKUMENTI R0-R3 V 3 DEKODIRAN  
NAROKO DEKODIRAN.



- veličina sklopa je?  $R0, R1, R2, R3 \Rightarrow 4 \text{ LOKACIJE, 4 BAJTA} \Rightarrow A_0, A_1! 2^2$
  - ako sklop ima REGISTRE kajima želimo upravljati te R0, R1...
  - tada se tim REGISTRI upravljajo putem teh register-select-a
  - R0 pali se na 1, a R0\* znači da pali na 0
  - zelena adresa je \$8008 \rightarrow
- |      |      |      |      |
|------|------|------|------|
| 0100 | 0000 | 0000 | 0100 |
| +    | 0100 | 0000 | 0000 |
| +    | 0100 | 0000 | 0000 |
| +    | 0100 | 0000 | 0000 |
- A15-A3 "FIKSNI"



ZADATCI

KAKO STAJI BC-CIJE?!

1.11.2009-12.00.8

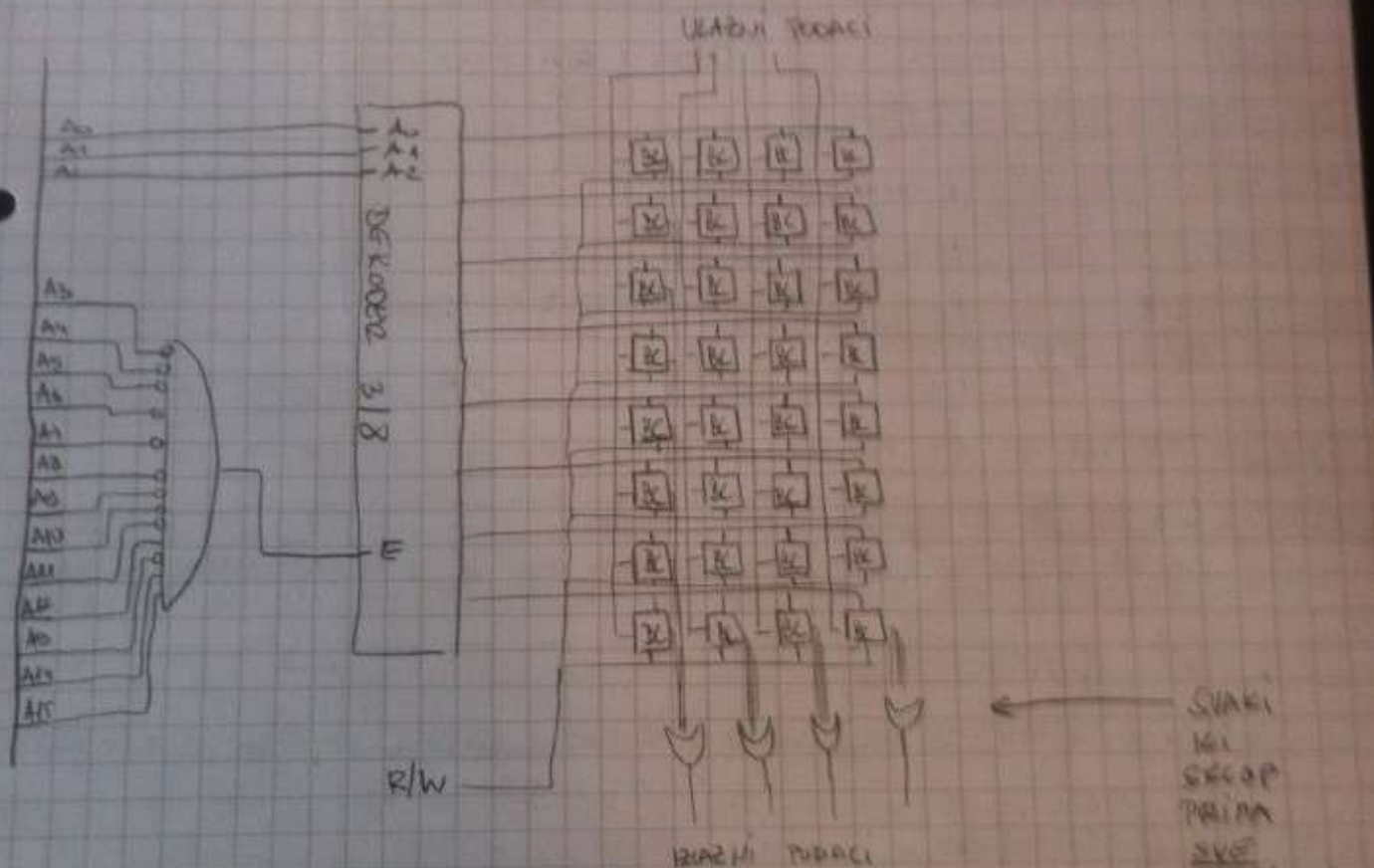
NACRTATI IZVEDBU MEMORIJE 8x4.

8 = broj bajti

4 = broj grupa bajti

PRISLUŠATI POKRETNOST ADRESU SADRŽANU TAKO DA POČINJE SA \$8000.

- 1) NACRTATI ZADNU BOJU BC-CIJE (8x4) = 32 BC-CIJE
- 2) ULAZNI PODACI ULAZE ODREĐENI, TE IZU LJU U BC
- 3) NACRTATI ODRŽE POKRETNOSTI (8) → DEKODER 3/8 → IZLAZI SVAKA BC
- 4) NACRTATI IZLAZ OZNAČIO DA IMAJU (KONTROLNE) IZLAZI SVAKA BC
- 5) "PRAVO" IZLAZE U KI SKUP
- 6) ISTOM METODOM KAO I BIVJE ODRŽE SVAKA BC ADRESA DO (8)



SVAKI KI SKUP PRAVO SVJE BC-CIJE A11 RADI TEČNO ČETIRI ALIČANI SVJE NACRTANO

- SKUP SE JAVIA NA \$8000

- S OBZIROM DA IMA 8x4 BAJTA ⇒ MUŽE ZAVIJAT SLOKACIJA!!

- STOGA ⇒

3000  
800 1  
800 2  
800 3  
:  
800 7

OVO JE ADRESA  
PROSTOR KOJ ZAVIJAT

⇒ VIDIŠU FISIČNI: A15-A3  
SLOKACI: A2, A1, A0

TE I U PRIKLADU NACRTANO

800 ⇒ A15 = 1011 0000 0000

## ② TIP II - ZADACI SA POJEDNOSTAVLJENIM CISC RAČUNOM

- PRIKAZ SADRŽAJA MEMORIJE ZA ZADANE INSTRUKCIJE, TE PRIKAZ STANJA NA TABLICAMA

ZADATAK 1

1.11.2011/2012

PRIKAZITE STANJE REZUALSTNOG DIJELA MEMORIJE, TE STANJE NA TABLICAMA, ZA POJEDNOSTAVLJENI MODEL, TIJEKOM IZVOĐENJA SLJEDEĆEG KODA:

ZADANO:

A - učitan 8bit konst u A, op.kod \$95  
 DD - op.kod \$B1 dodaje operand++  
 C - op.kod \$12 umanjuje operand--  
 E - op.kod \$46 grana na zadani adres  
 ako rezultat ≠ 0

\$0100	:	LDA	#00
		ADD	\$0200
		DEC	\$0200
		BNE	\$0100
\$0200			\$03

RIJEŠENJE:

MEMORIJA → KREĆEMO OD ZADANOG POČETKA \$0100  
 → NA \$0100 DOLAZIMO DO INSTRUKCIJE LDA, KOJA JE U MEMORIJI POVRATNA KAO ŠTO?  
 KAO SVOJ OPERACIJSKI KOD \$95

⇒ \$0100 : \$95

→ NO NARAVNO UZ TAJ KOD MORAMO SE NAHAZITI I OPERAND, TJ OVDJE JE TU KONST.

⇒ \$0100 \$95  
 \$0101 \$00

→ TINE SMO GOTIVI SA INSTR LDA, NO DA JE PISNO LDA \$0010 ONDA BI MORALI U SVOJ SEDMO MEMORIJSKU LOKACIJU UČITATI TAJ DODATNI 01

→ SAD DOLAZIMO DO ADD \$0200, INSTRUKCIJA JE U MEMORIJI POVRATNA KAO SVOJ OP. KOD, TE PRIMA \$0200 KAO OPERAND

⇒ \$0100 \$95  
 \$0101 \$00  
 \$0102 \$B1  
 \$0103 \$02  
 \$0104 \$00

→ SADA NAHAZIMO NA DEC \$0200 ⇒ ISTO

\$0100 \$95  
 \$0101 \$00  
 \$0102 \$B1  
 \$0103 \$02  
 \$0104 \$00  
 \$0105 \$12  
 \$0106 \$02  
 \$0107 \$00



→ INŠTANJE

PRINJETI DA VAŽIJI TI TEŽI BIT SVAKE ADRESE, KPE IZ \$0200 LEŽI NA NIŽOJ ADRESI, ODNOSNO AKO STAVJAMO ADRESE OVIM REDOSLJEDOM KOJIM IH PIŠEMO ⇒ BIG-ENDIAN

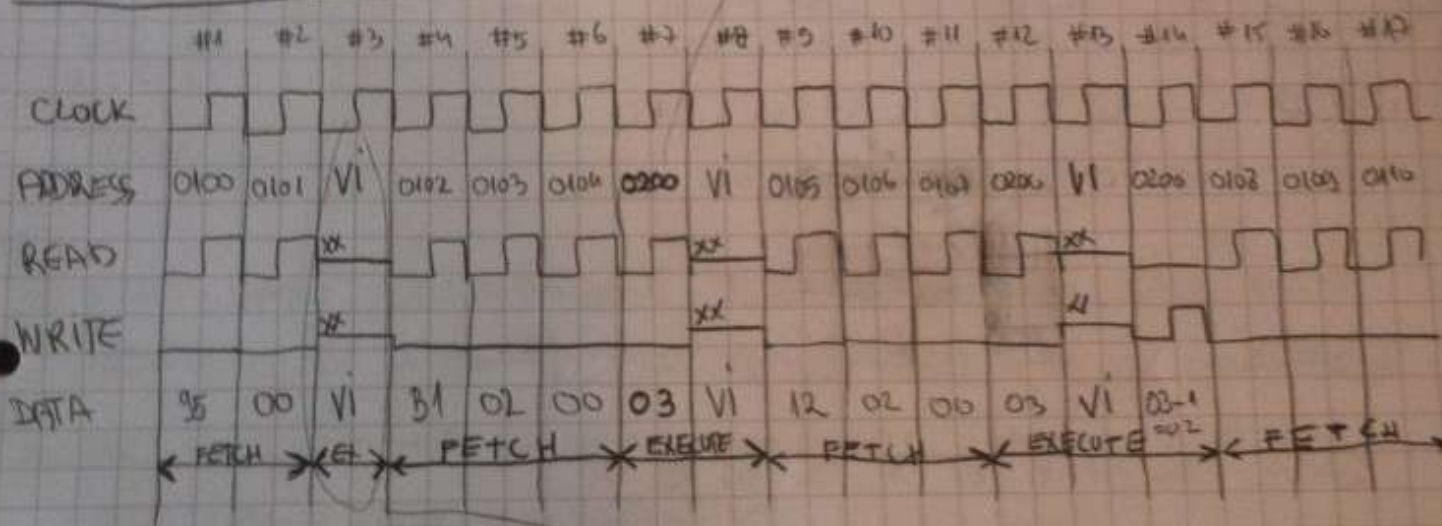
→ NA KRAJU OXAZINU DO BNE \$0100 DOKLE

KRAJINU:

\$0100	:	\$95
\$0101	:	\$00
\$0102	:	\$B1
\$0103	:	\$02
\$0104	:	\$00
\$0105	:	\$12
\$0106	:	\$02
\$0107	:	\$00
\$0108	:	\$46
\$0109	:	\$01
\$0110	:	\$00

ADD \$0200 PRIO NABAVNO  
ULTA SVJ. OP. CODE, PA OPEKANO, TE  
OVDA STACE VINTU ADRESU \$0200, TE  
TAMU NABAVNA PODATAK I OVDA  
IDE U EXECUTE I VRAĆA SE GDJE JE  
STAO  
- VIŠKA INFORMACIJA JE SAMO OBAVJES  
TE OPEKANE, NAKON KEGA DOKLE IDE NRIPE

## STANJE NA SABIENICANA



→ PC RASTE ČIM RADE NA SABIENICU, U FETCH TAZI

→ MUAMO TOČNO ZNATI ŠTO I KAKO RADE ZADANE INSTRUKCIJE!

LDA - dohvata konstantu te ju sprema u A → NIJE 100% JASNO ZADANO U ZADATKU DALI KORISTI ITACT ZA STORE ILI NE ⇒ PITATI ASISTENTA.

ADD \$0200  $A \leftarrow A + M[\$0200]$  → dohvati operand → EXECUTE AD

DEC/INC \$0200 dohvati iz  $M[\$0200]$ , snaji za 1 U EXECUTE TAZI

BNE \$0100

WRITE/READ - SAMO ZA ČITAJ/PIŠI U MEM!  
NE U AKUMULATUR!

## ZADATAK 2

21. SKI ROK 2012

ZA POJEDINOSTAVLJENI MODEL PROCESORA NACRTAJI STANJE IIA SPOLJNOSTI

TJERKON IZVODENJA

LDA # 1
OPET: DEC A
BNE OPET

KOD POČINJE NA \$0780

LDA op. kod \$A3

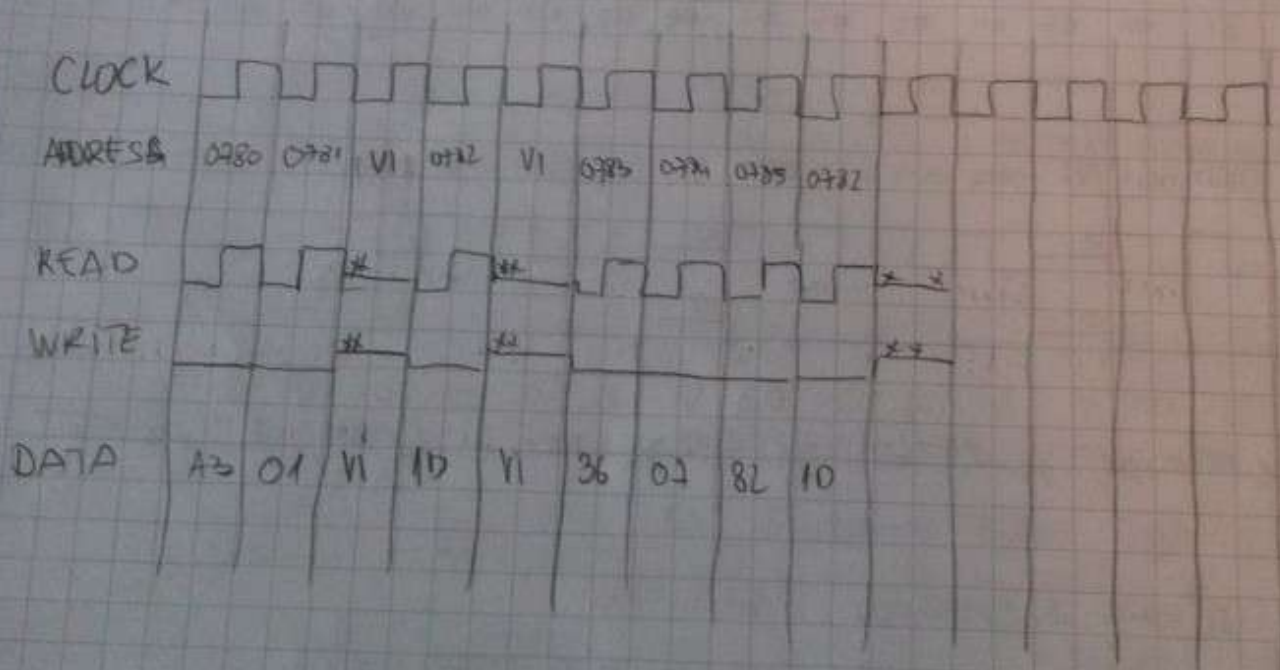
DEC op. kod \$1D

BNE op. kod \$36

DODATNO: ODREĐITE STANJE REGISTARA NAKON IZVODENJA SVAKE NAREDBE! (PC, A, D, R)

MEMORIJA:

\$0780	\$A3	op. kod instrukcije LDA	OPET = \$0782
\$0781	\$01	međunarodni broj instrukcije	
\$0782	\$1D		
\$0783	\$36		
\$0784	\$04		
\$0785	\$82		



AKO IMAMO LDA # 1, DAKLE UČITAVAMO KONSTANTU U REGISTAR A (AKUMULATOR) => NEMA SKAKAJA PO MEMORIJI, VEĆ INSTRUKCIJA "INJEKCIJA" U CPU ODREĐUJE EXECUTE



### ZADATAK 3

NACRTAJI STANJE NA SADRŽANJE (M=10) NA RAČUNALU 4185 JV  
ODREĐIVANJE NA TEMELJU POJEDINOSTAVLJENOG CRTEŽA. ZADANI SU GEDNATSKI  
ODSJEČAK

0000 B6 LDA \$A000

0001 A0

0002 00

0003 CE INC \$A00B

0004 A0

0005 0B

0006 FF

A000 A1

A001 22

A002 33

A003 44

A004 55

A00A B1

A00B 29

A00C 23

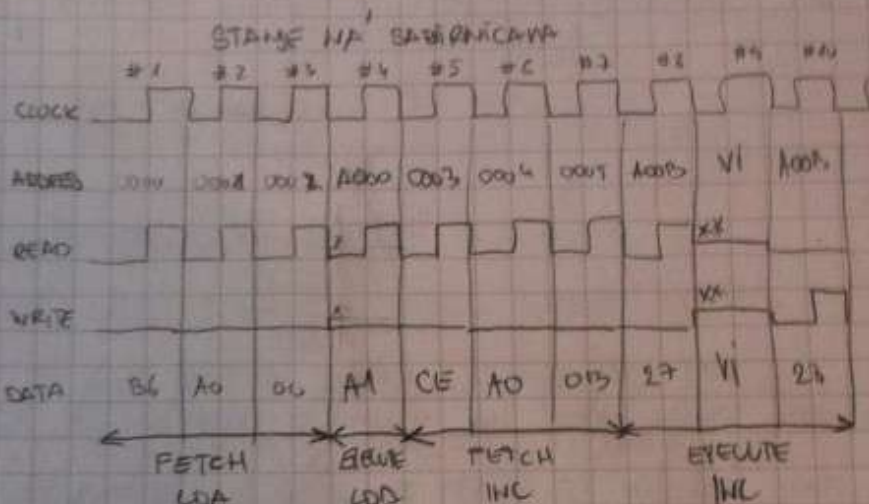
A0A0 B1

A0A1 C1

B6B4 B0

B6B5 0A

B6B6 B1



⇒ KADA IMAMO LDA \$A000 DA BI SE OBAVILA LDA

MORA OTIĆI NA A0000 TE TAMO OBAVITI READ I TO JE NJEN EXECUTE

⇒ INC EXECUTE IMA DOHVAĆ SA ADRESE, ZATIM "INTERU" OBRADU, TE NA KRAJU  
UPIS, IATO JE EXECUTE FAZA DUGAJKA 3T

### STANJE U REGISTRIMA

- |             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| ① PC = 0000 | ② PC = 0001 | ③ PC = 0002 | ④ PC = 0003 | ⑤ PC = A000 |
| DC = /      | DC = /      | DC = A0     | DC = A000   | DC = A000   |
| A = /       | A = /       | A = /       | A = /       | A = A1      |
| IR = /      | IR = B6     | IR = B6     | IR = B6     | IR = B6     |

### ③ TP III

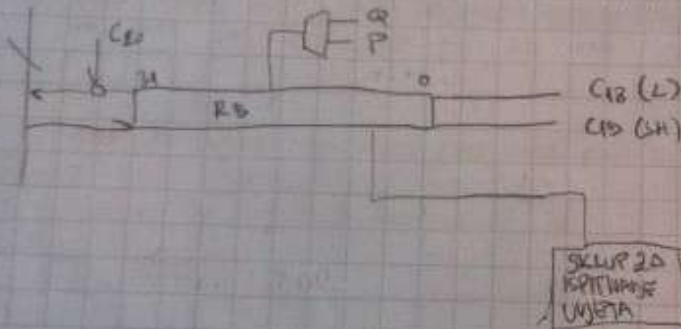
## MODIFICIRANJE / ANALIZA 8-INSTRUKCIJSKOG RAČUNALA

2011/2012 ni 1

### ZADATAK 1

PREPOSTAVIMO DA 8-INSTRUKCIJSKOM PROCESORU ŽELIMO DODATI JOŠ JEDAN REGISTAR OPĆE NAMJENE. RA KOSI JE ZAMISLJIVANO KOJE PROMJENE TREBA OBAVITI U ARHITEKTURI, PUTU PODATAKA, I E U UPRAVLJAČKOJ JEDINICI?

1<sup>o</sup> FIZIČKA PROMJENA - DODAVANJE REGISTRA. - moramo izmijeniti / dodati



- registar  $R_8$  mora imati jednu liniju jednolikih upravljačkih signala ( $C_{13}, C_{12}, C_0$ )
- registar  $R_8$  mora također biti spojen na "skup za ispijanje uvjeta"
- registar  $R_8$  mora moći moći biti na  $Q$ , moći na  $P$ , da li se mogao izvoditi sa  $R_8$  ili sa drugim operacijama

2<sup>o</sup> PROMJENE U INSTRUKCIJSKOJ ARHITEKTURI -

- treba projektirati nove instrukcije npr.  $LD B, X$ ,  $ADD R_8, R_8$  itd.
- sada ćemo imati ukupno 16 instrukcija, a ne 8 ??

PROMJENE U UPRAVLJAČKOJ JEDINICI -

- pošto imamo više od 8 instrukcija moramo dekoder iz  $3/8 \rightarrow 4/16$



ZADATOK 2 PROMENITI NODZ SVAKOJ INstrukciji TAKO DA SE OPOSREDI  
RELATIVNU ADRESIRANJE, ZA LOAD I STORE.  
NAPISATI JEDNADŽBE SUH SIGNALA ZA NOVE INstrukcije LOAD I STORE.

ABSOLUTNO ADRESIRANJE: INstrukcija UZIMA ABSOLUTNU ADRESU IZ NODZ

RELATIVNU ADRESIRANJE: INstrukcija RELATIVNU ADRESU (NAK+ODNAK) PRETVARA U  
PRAVU ABSOLUTNU ADRESU TAKO DA ZBRAJAJE ODNAK+PC

<u>UVIJEK:</u>	PRIBAVI:	PC → NAR	PC STAVI U NAR	} UVIJEK ISTO
		NDR ← M[NAR]	DOHVAĆI IZ NODZ I STAVI U NDR	
		PC ← PC + 1; IR ← 100	POVRATI PC, PUNJI INstruk. REG.	} OVISI O INstrukciji
	IZVRŠI (STORE):	NDR[23:0] → NAR	DOJE POHVAĆIVATI	
		NDR ← A	PUNJI NDR	
		M[NAR] ← NDR	NDR BACI U NCH	

STORE STA pohranjuje sadržaj registra A

- 1° gdje se pohraniti mora zapamti u NAR
- 2° iz A uzima sadržaj i stavlja na ndr
- 3° taj sadržaj sprema na M[NAR]

OBICAN STA = 1

RELATIVNI STA = 1

1°  $NAR \leftarrow NDR[23:0]$

$C_5 = I_1 \cdot (\phi_8 + \phi_9)$

$C_5$  - aktivira ndr → Int Bus  
kaj trije 2T

$C_{13} = I_1 \cdot \phi_9$

2°  $ndr \leftarrow A$

$C_7 = I_1 (\phi_{10} + \phi_{11})$

$C_{14} = I_1 \cdot \phi_{11}$

- iz slide se vidi kaj signal

3°  $M[NAR] \leftarrow ndr$

$C_4 = I_1 (\phi_{12} + \phi_{13})$

↑ iz CU ↓ nad sa PEN  
2T

1° NDR ZBRAJAJE ODNAK I PC →  $ALU \leftarrow NDR[23:00]$   
 $ALU \leftarrow PC$

$C_5 = I_1 \cdot (\phi_8 + \phi_9)$

$C_{13} = I_1 \phi_9$

2° ALU ZBRAJA P+Q, TE ZATIM STAVJA NA Int Bus

$C_6 = ADD = I_1 (\phi_{10} + \phi_{11})$

$C_5 = I_1 \phi_{12}$

$C_6 = I_1 (\phi_{12} + \phi_{13})$

DALJE ISTO

3°  $NAR \leftarrow P + Q$  (ne jema Int Bus)  
 $C_{13} = I_1 \cdot \phi_{15}$

4°  $ndr \leftarrow A$

$C_7 = I_1 (\phi_{16} + \phi_{17})$

$C_{14} = I_1 \phi_{17}$

5°  $M[NAR] \leftarrow ndr$

$C_4 = I_1 (\phi_{13} + \phi_{15})$

ZADATAK 3

MODELU BINAJE PROCESORA ŽELIMO DODATI NOVU INSTRUKCIJU SUBA X ( $A \leftarrow M[X] - A$ )  
NABAVITI POTREBNE IZMENJE. NAPOISATI FAZU IZVRŠI.

- POŠTO ODDZIVAMO

$M[X] - A$

TRAJEBA NDI INVERTIRANJE  $\bar{A}$ , PRJE SAGA U ALU



SUBA X

- 1° odnosi se na X kraj MAR
- 2° odnosi se na NDR
- 3° odnosi se na ALU, zbog

→ DODAJEMO NOVU INSTRUKCIJU → NIJEJA SE DEKODIR, ŠIRIJA OP KODA, DODATI SIGNALI C

FAZA PRIBAVI  
(INJEKCIJA)

- ①  $MAR \leftarrow PC$
- ②  $NDR \leftarrow M[MAR]$
- ③  $PC \leftarrow PC + 1$
- ④  $IR \leftarrow NDR$

FAZA IZVRŠI  
(SUBA X)

$$C5 = I_5 \cdot (\phi_8 + \phi_9)$$

$$C13 = I_9 \cdot \phi_9$$

$$IntBus \leftarrow NDR$$

$$MAR \leftarrow IntBus$$

} KLASIČAN  $MAR \leftarrow NDR$

$$C3 = I_5 \cdot (\phi_{10} + \phi_{11})$$

$$C14 = I_9 \cdot \phi_{12}$$

} READ from CU 2T mesmije zadržati  
UPISU NDR

$$C5 = I_9 \cdot (\phi_{13} + \phi_{14})$$

$$NDR \leftarrow M[MAR]$$

$$C18 = I_9 \cdot (\phi_{14})$$

$$P \leftarrow A \quad (M[X] \text{ je većina } P)$$

$$C10 = I_9 \cdot \phi_{15}$$

ALU ZBIRAJA

$$C15 = I_9 \cdot \phi_{16}$$

$$IntRegALU \leftarrow ALU$$

$$C6 = I_9 \cdot (\phi_{17} + \phi_{18})$$

$$IntBus \leftarrow IntRegALU$$

$$C10 = I_9 \cdot \phi_{18}$$

$$A \leftarrow IntBus$$

POŠTO OPERACIJA TRAJE  $\phi_{18}$  NUŽNE IZMENJE I NA  
GENERATORU PO NOVOU N (IMPULSA)



# CRTANJE ZOMBILACIJSKOG SKLOPA

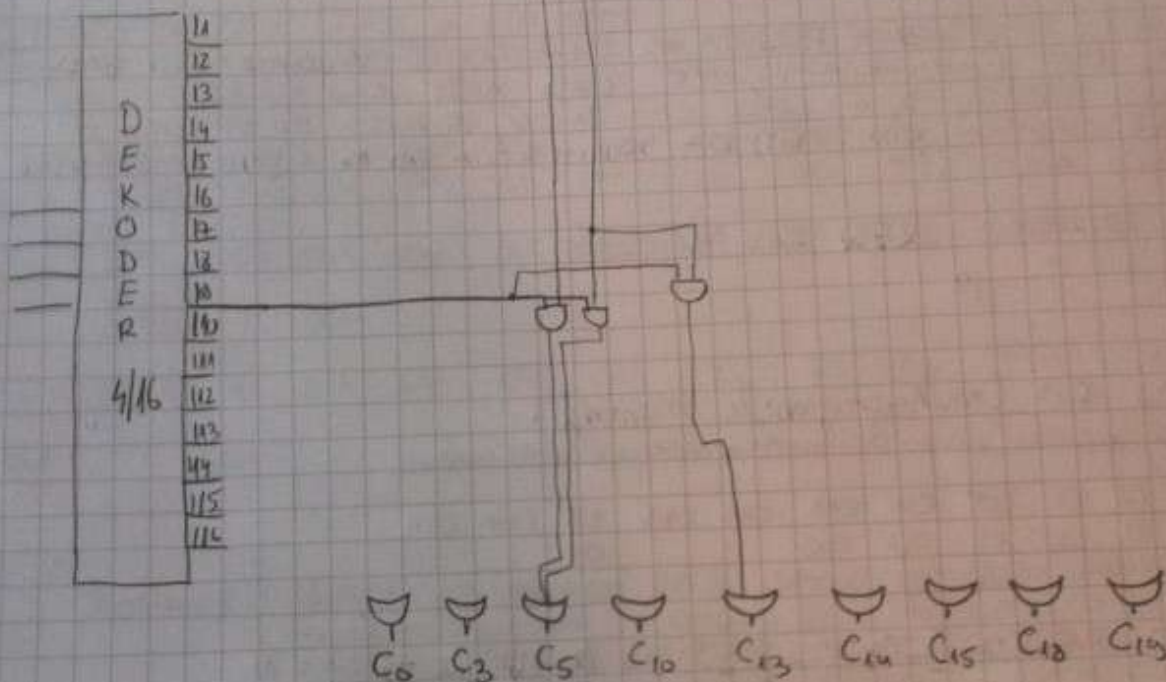
NASTAVAK NA ZADATAK SA DEKANSKOG ROKA

KAKO SE ZADANI NZ JEDNAOŽBI CRTA?

1° IMAO VIŠE OD 8 IMEN → DEKODER 4/8

2° GORE  $\phi$ , DOLE C (KJE SMO KRESTILI)

$\phi_1 \phi_2 \phi_3 \phi_4 \phi_5 \phi_6 \phi_7 \phi_8 \phi_9 \phi_{10} \phi_{11} \phi_{12} \phi_{13} \phi_{14} \phi_{15} \phi_{16} \phi_{17} \phi_{18} \phi_{19}$



• NZR INARO  $C_5 = \lg \cdot (\phi_8 + \phi_9) = \lg \cdot \phi_8 + \lg \cdot \phi_9$

$C_5$  je ZBROJ DIA UNAKŠKA (1)  
(121)

$C_{13} = \lg \cdot \phi_9$  ITD.

# 4) TPR4 MIKROPROJEKCIJE

21.11.2012

## ZADANJE

ZA ROL MIKROPROJEKCIJE ZADAN SU, TE SA ZADANIM FUNKCIJAMA MIKROPROJEKCIJE, NAPIŠI MIKROPROGRAM ZA JREQ

JREQ JE POKRETAČ ZA 1, AKA A=B (A I B AKUMULIRAJE), A U SUPROTNOJ PC OSTAJE NEIZMJENJEN.

OPREDAJ JREQ JE \$40, A POČETNA ADRESA FAZE PRIBAVI \$F0

- IZ SUKE TE IZ TABLICE FUNKCIJA => DIREKTNO OČITAVANJE ŠTO RADI KOJI DIO MIKRO-RJEŠI, NPR CAB NOJE AKA NU UPISANO 00 BACITI SADEJAJ PC -> LIJEVO SADRŽAJU, A AKO NU UPISANO 01 BACITI SADRŽAJ CEN (381) NA DRUGIH ZETA OD 16 LIJEVO SADRŽAJU I TAKO DAGE !!
- NAKA NAKREDA JREQ RADI SJEDEĆE:

10

## SPOROZAS CN (mikroprogramske memorije)

POČETAK MIKROPROGRAM = OPERACIJSKI KOD MIKROPROJEKCIJE

	CA	CB	COP	CSH	CNB	CAB	CBB	CST	CNA	CEN
\$40	11	010	01	00	000			01		

ŽELIMO UPUREDITI A I B => DONOSIMO A LIJEVO => CA = 11  
 DONOSIMO B DESNO => CB = 010  
 ZBRAGAMO UZ C=1 => COP = 01  
 NE ŽIFTANO => CSH = 00  
 NE NOSIMO => CNB = 000  
 REZULTAT ZERO TESTA SE => CST = 01

PITAMO SE:

SJEDEĆE DA BI ODREDITI CNA CBB I CAB, : JEL IDE FABA FETCH SADA ILI MIKROPROJEKCIJE? PA AKO A=B IDE, DAKLE POSTOJI MOGUĆNOST => CNA KOJI GUVORI (UJELJEUAH) NUKA BITI PRVIH 6 BITOVA OD \$F0 TI 1111 0000 => CNA 1111 00

=> DAKLE SADA CE CAB I CBB "ODLUČITI" IDENO LI MA FETCH ILI MA SJEDEĆE MIKRO OPERACIJA! => IDENO ILI NA 1111 0000 ILI NA 1111 0000

=> AKO REZULTAT BUDE \$F0 => mikroprojekcija DAKLE NI CAB I CBB 10 00 => \$F2

=> DAKLE PILENO SJEDEĆE

	CA	CB	COP	CSH	CNB	CAB	CBB	CST	CNA	CEN
\$40	11	010	01	00	000	10	00	01	111100	00000000
\$F2	01	011	00	00	011	00	00	00	111100	00000000

DONOSIMO A, DONOSIMO B, ZBRAGAMO, ZET=SE, IF A≠B ODI NA \$F0, AKO A=B ODI \$F2

AKO A=B DAKLE PC=PC+1, ODI NA F0



ZA MODEL MIKROPROGRAMIRANOG PROCESORA KOJI JE ZADAN SLIKOM, TE FUNKCIJOM MIKRO FETČI ZA DJNZ NAPIŠI MIKROPROGRAM (ODGOVOR: SADEŠNJ MIKRO PROGRAM ŠICE MIKROFETČI)

DJNZ PRVO B--, ZADIM  $IF(B--) \neq 0$  SKAČE NA PC+A  
 $IF(B--) = 0$  IDE NA MIKRO FETČI

OPERACIJSKI KOD INSTRUKCIJE \$AB = 1010 1011

MIKRO PRIBAVI \$FO. = 1111 0000

	CA	CB	COP	CSH	CMB	CAB	CBB	CST	CJP	CEN
\$AB	01	001	00	00	010	00	00	00	101011	1111 1111
\$AC	01	001	00	00	010	M	00	01	11100	0000 0000
\$FO	NITEXH									
\$PZ	11	011	00	00	011	00	00	00	111100	0000 0000

KORAK 1: ŽELIMO B SNAJITI ZA (A) DAKLE NA LIJEVU CABI RNICU DOVOJIMO 1111 1111, A DESNO B I POZBRUJIMO TO.

DAKLE  $\Rightarrow CEN = 1111 1111$ ,  $CA = 01$ ,  $CB = 001$ ,  $COP = 00$ ,  $CSH = 00$

$\Rightarrow CAB = 00$  JER  $B \in NB$  DA SNAJIMO B,  $CST = 00$

MISNO JOŠ GOTOVI SA INSTR, TI DEFINITIVNO NEJEDNO JOŠ I A NFETČI KOJI JE I A 1111 0000, PA POŽELJO IČ I A \$AC!  $\Rightarrow CA = 101011$   
 $\Rightarrow CAB = 00$   $CBB = 00$

KORAK 2: SADA U DRUGOJ MIKRO INSTRUKCIJI ŽELIMO B USPOREDITI SA NULOM, TE OČIJA ICI SKOČITI NA NFETČI ICI I A MIKRO INSTR KOJA CE PRVO PC=PC+1 PA OČIJA SKOČITI NA NFETČI.

BRŽ USPOREDIMO SA NULOM TAK DA ZBROJIMO  $0+B$  I A KO ŠNO DOBILI I A A  $B=0$ , I AČE NIJE

DAKLE LIJEVO DOVOJIMO LIJEVO 0000 0000 0000 0000, A DESNO DOVOJIMO B, ZBROJIMO, A REZULTAT ŽT SPREMIMO U  $GR(0) \leftarrow ZT$   
 DAKLE  $CA = 01$ ,  $CB = 001$ ,  $COP = 00$ ,  $CSH = 00$ ,  $CMB = 010$ ,  $CST = 01$

SADA JE MOGUĆE DA  $(B--) = 0 \Rightarrow$  MIKRO FETČI I A ADRESI \$FO = 1111 0000  
 DA  $(B--) \neq 0 \Rightarrow$  DODATNA INSTR I A ADRESI \$PZ = 1111 0010

\$PZ ODOBRAO RADI LAKŠEG RJEŠAVANJA TAKO DA  $H(1)$  KONTROLINA SNAJMI I DA NI ODGOVORA ČNA

## ZADATAK 3

11.11.2012/2013

ZA MODEL MIKRO PROGRAMIRANOG PROCESORA IAPITEFAZU IZURSI ZA  
BAZB (AKO  $A=0$  ODI NA ADRESU 12B)

BAZB OP. KOD \$5A = 0101 1010

IFETCH \$00 0000 0000

→ AKO  $A=0 \rightarrow$  skoci mi na [B]  
INACE IFETCH

## MIKRO MEMORIJA

	CA	CB	COP	CSH	CNB	CAB	CBB	CST	CNA	CEN
\$5A	11	000	00	00	001	10	00	01	000000	0000 0000
\$00	IFETCH									
\$02										

## KORAK 1

PRVO JE LINO USPOREDITI A SA NULOM. MOŽEMO U B DOVEST NULA  
NULU, POUZBOJITI SA A, SPRENITI ZERO-TEST U SR  
⇒  $CA=11$  LEA,  $CB=000$  R=0,  $COP=00$  A=0,  $CSH=00$  IBC=5  
 $CNB=001$  AG=NB, i 2T → SR(0) DAKLE CST=01  
⇒ AKO  $A=0$  TADA NEH[B] NOVA INSTRUKCIJA PA NAKON TUG IFETCH  
AKO  $A \neq 0$  TADA IFETCH ODMAH!

⇒ DAKLE CNA=0000 00, TE CBB=00 ⇒ DAKLE CAB  
JE "ODLUŽITI" HOĆE U SKOK IĆI NA IFETCH ILI \$0002

## KORAK 2

INSTRUKCIJA NA \$02 JE HIJEĆI STAVITI B U PC



## ⑤ TIP 5

## CRTANJE KORISNIČKIH I UPRAVLJAČKIH STOGOVA

- ZADATAK - PRIMJER ZA KORISNIČKI STOG: DOBIJEŠ KOD I TRAŽI TE DA SKICIRAŠ STOG.

```

VOID *ALOCIRAJ (INT N) {
    VOID *BUF = MALLOC(N);
    memset (BUF, 0xFF, N); //B
    RETURN BUF;
}

```

```

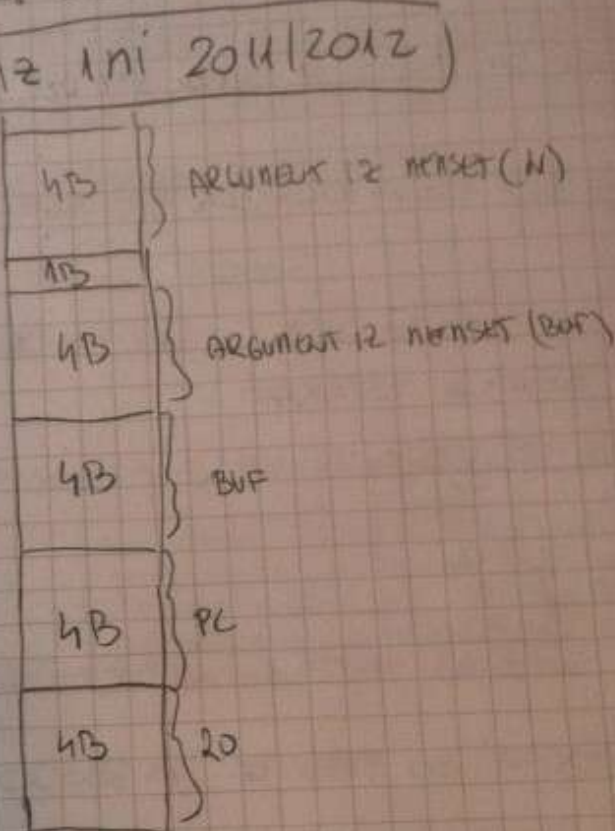
INT MAIN () {
    //A
    ALOCIRAJ (20);
    //C
}

```

A)



B)



1° ARGUMENT FUNKCIJE

2° PC

3° LOKALNE VARIJABLE



KOD POZIVA I ULASKA U FUNKCIJU UVIJEK OVIM REDOSLEDOM OVO STAVJAMO NA STOG.

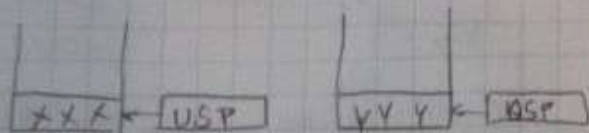
## ZADATAK 2

11. 2011/2012

PROCESOR IC6800 POZIVA KONTROLNI PROGRAM \$2300: JSR \$3200, OP. KOD  
ZAUZIMA 6 BAJTA. PARAMETRI SE NE PRENOSE PREKO STOGA, A KONTROLNI KORISTI  
DUGE LOKALNE VARIJABLE OD 18.

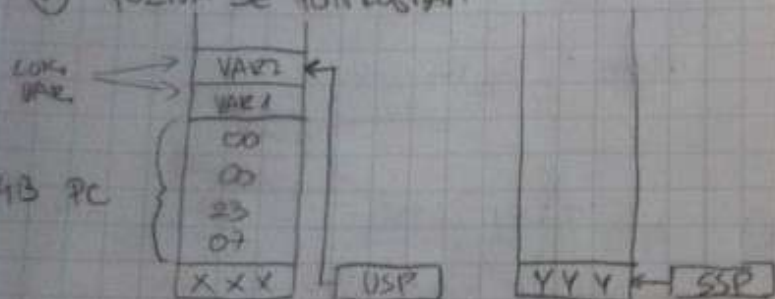
NAKON ZAVRŠAVANJA \$3300 PROCESOR ZAPRIMA ZAHTEV ZA PREKIDOM, POZIVA SE  
PREKIDNI PROGRAM NA ADRESI \$FE080

① PRIJE POZIVA KONTROLNOG PROGRAMA.



STOGU SU PRAZNI

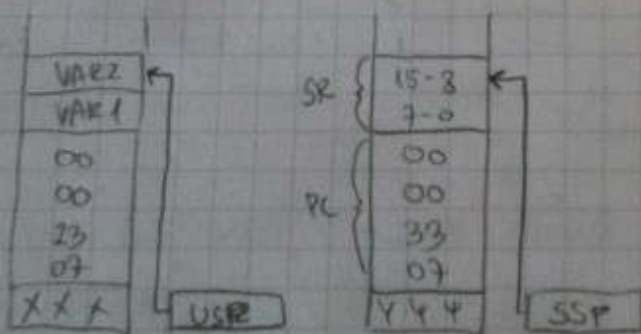
② POZIVA SE KONTROLNI PROGRAM



na adresi \$2300 čitamo JSR  
druge SADR SE NAČITAMO  
NA \$2300

- 1° ARGUMENT NE PREKIDAMO JSR  
TAKO ZADANO
- 2° PREKIDAMO PC
- 3° PREKIDAMO 2-LOKALNE VAR

③ DOLAZI DO PREKIDA NA \$3300



KOD IRUINKI SE KORISTI  
UPRAVLJACKI STOG, A KONTROLNI  
SE NE MIJENJA!!

KOD PREKIDA

- 1° PRVO STAVIMO PC NA STOG
- 2° STAVIMO SR 2B

KAD SE VRATIMO IZ PREKIDA STOG NE BIRAMO VEĆ SAMO NIŠTAKO SSP

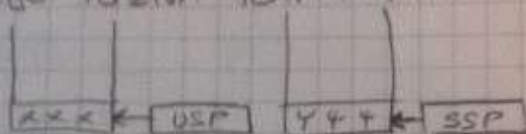


ISTO SE STOG NE MIJENJA VEĆ SAMO USP

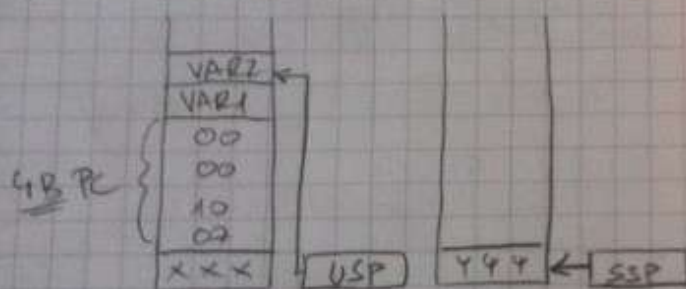


- PROCESOR MC 6800 POZIVA POTPROGRAM INstrukcijom \$1000:JSR \$2000  
 OP. KOD ZAOZINA 6 BAJTA. PARAMETRI SE NE PREKIDUJE PREKO SDGA  
 TE NEKA POTPROGRAM KORISTI DUJE LOKALNE VARIABLE ĆIRINE 1B.  
 → NEPOSREDNO NAKON IZVRŠAVANJA \$2100:MOVE D0,DA PROCESOR PRIMA  
 ZAHTEJ ZA PREKID, TE SE POKREĆE PREKIDNI POTPROGRAM NA  
 ADRESI \$FF0A0

① PRVE POZIVA POTPROGRAMA → STOGU PRAZNI!

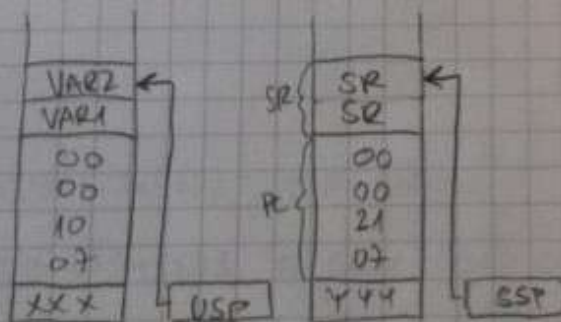


② NAKON POZIVA KORIŠNIČKOG POTPROGRAMA → PUNIO KORIŠNIČKI STOG.



MO OCHENI \$1000 ĆHOMO OP. KOD  
 00 6 BAJTA ⇒ SADA SMO NA  
 ADRESI 12A TOGA. TU SE POZIVA  
 POTPROGRAM ⇒ STAVJAMO PC,  
 TE LOKALNE VAR IIA STOG

③ DOLAZI PREKID ⇒ PUNIO NADGLEDI/SUSTAVNI STOG



④ ISTO ISTO SANO SE KOD KRAJA PREKIDA SSP VRATI

⑤ ISTO ISTO SANO SE KOD ZAVRŠETKA POTPROGRAMA USP VRATI

# 6. TIP VI

## PRIRUČNE MEMORIJE

SAVREMENA

21. 2003/2009

RAZLIKA JE PRIRUČNA MEMORIJA SPOSAN-ELEMENTARNI ASOCIJATIVNOSTI  
 NA KOLIKO SA 32 BITNIM ADRESAM. ZNAČIJE JE 1B, KARAKTER  
 REKURSIJE 32B, A ŠIRINA LINIJE 64B. IMAMO I VIO SEKULNE BITOVE.

### 1. STRUKTURA ADRESE = ?

$$32 = w(p) + w(o) + w(i)$$

$$w(i) = \log_2 \left( \frac{n}{s} \right)$$

$$w(p) = \log_2 (b)$$

b = ŠIRINA LINIJE  
 a = ASOCIJATIVNOST  
 n = BROJ LINIJA PN

z = ZNAČIJE

$$w(i) = \log_2 \left( \frac{128}{8} \right) = \log_2 16 = \log_2 2^4 = 4$$

$$w(p) = \log_2 (64) = \log_2 2^6 = 6$$

$$w(o) = 32 - 6 - 4 = 22$$

PN ima 8192B =>

$$m = \frac{8192B}{64B} = 128$$

$$z = 1 \quad b = 64$$

STRUKTURA

ADRESA = 0000 0000 0000 0000 0000 0011 11PP PPPP  
 22 4 6

VIO TU IOU DODATKU

### 2. 0X12345678 IOE KANO?

$$0X12345678 = 0001 0010 0011 0100 0101 0110 0111 1000$$

i -> KAN GOVORI KANO IOE BLOK  
 A TINE I ODREĐENA ADRESA!! => 1001

FORMULE

$$w(p) = \log_2 \left( \frac{b}{z} \right)$$

$$w(i) = \log_2 \left( \frac{n}{a} \right)$$

$$w(o) = \text{ŠIRINA} - w(p) - w(i)$$

z = ZNAČIJE

n = BROJ LINIJA PN =  $\frac{\text{KARAKTER}}{\text{ŠIRINA}}$

a = ASOCIJATIVNOST

b = ŠIRINA



## ZADATAK 2

Zilski ROK 2012

ZADANA PM SA CENEROSTRUKOM ASOCIJATIVNOŠĆU, NA 32 BIT. ENKLAVU,  $k=1$ .  
KAPACITET (S) MEMORIJE JE 16kB, A ŠIRINA LINIJE 32B. (V.I.D)

STRUKTURA ADRESE ?

$$w(p) = \log_2 \left( \frac{b}{2} \right) = \log_2 \left( \frac{32}{2} \right) = \log_2 16 = \log_2 2^4 = 4$$

$$w(i) = \log_2 \left( \frac{n}{a} \right) = \log_2 \left( \frac{2^{10}}{8} \right) = \log_2 \left( \frac{2^{10}}{2^3} \right) = \log_2 2^7 = 7$$

$$\Rightarrow w(o) = 32 - 4 - 7 = 21$$

$$n = \frac{S}{b} = \frac{16kB}{32B} = \frac{2^{14}B}{2^5B} = 2^9$$

STRUKTURA ADRESE = V, D,

21 BIT      6 BITA      5 BITA  
OZNAKA      INDEKS      PPM

GDJE JE ADRESA 0x21006543?

- RASPIŠENO ADRESU, VIDIŠO GDJE SU "6" BITOVI

KOLIKI BI BIO GNJEZ POGODAKA/PROMAŠNJA KADA BI SLEDJEDNU  
ČITALI POGE int 7 [128], POČEVŠI OD 0x21006540?