

U MODEL 8-INSTRUKCIJSKOG PROCESORA DODAJTE NOVU DEVEU INSTRUK.

LDA #KONST(A)

ZADANO: LDA UČITAVA PODATAK IZ MEMORIJE SA ADRESA $A + KONST$ TE ONDA TO SPREMA U A. NACRTATI KOMBINACIJSKI DFO CU ZA TU NOVU INSTRUKCIJU.

FETCH : ISTI ZA SVE

IZIR:

NIJE SE DEKODIR IZ 3/8 U 4/16, U 12 - PISANOM 4 BITA

① MDR + A

ENDR $C_5 = I_9 \cdot (\phi_8 + \phi_9)$

ADD $C_6 = I_9 \cdot (\phi_8 + \phi_9)$

LALU $C_{15} = I_9 \cdot \phi_9$

PROPUŠTANO SADRŽAJ MDR-a NA INTERNU SABIRNICU

ZBRAJANO SADRŽAJ MDR + AKUMULATOR DVA CIKLA

PROPUŠTANO REZULTAT U INT REZ ALU

② INT REZ ALU → NAR

EALU $C_6 = I_9 \cdot (\phi_{10} + \phi_{11})$

LNAR $C_{13} = I_9 \cdot \phi_{11}$

PROPUŠTANO REZULTAT DO MEMORIJE ADRESI REG.

NA INTERNU SABIRNICU PUTAMO REZ. DVA CIKLA

PUNIMO NAR

③ M(NAR) → MDR

READ $C_3 = I_9 \cdot (\phi_{12} + \phi_{13})$

LNDR $C_{14} = I_9 \cdot \phi_{13}$

ENDR $C_5 = I_9 \cdot (\phi_{14} + \phi_{15})$

LA $C_{10} = I_9 \cdot \phi_{15}$

ČITAMO SA MEM (NAR+A) → MDR

ČITAMO MEMORIJU DVA CIKLA

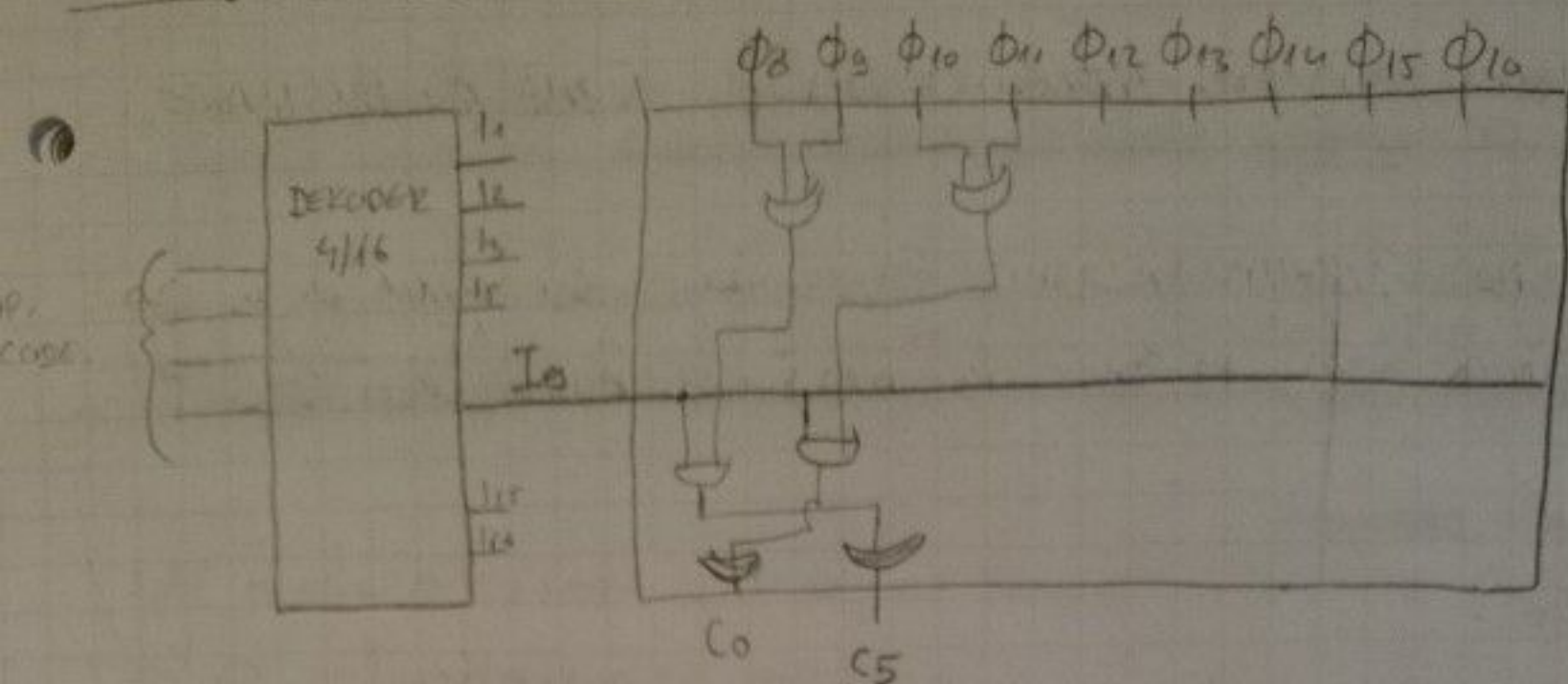
PUNIMO MDR SA PROČITANIM

PUŠTAMO MDR → INT BUS

PUNIMO AKUMULATOR

OD KUDA SMO KRENULI? PA PRETPOSTAVKA JE DA SE OBAVIO PRIBAVI
OD LDA #KONST(A) A ON NARUMI MDR, IZ NJEGA 12, DEKODIRA SE
INSTRUKCIJA I INAMU SPREMA MDR
- NARUMI DA SA SPREMANIM MDR-OM OČITO U MEMU INAMU KONSTANTU!!

CRTANJE KONVILACIJSKOG DJEŁA


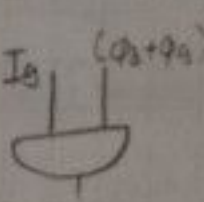


INSTRUKCIJA I_0 PREDSTAVLJA "ČVRSTU" JEDINICU, GENERIRANU OP. CODEM -> DOKAZ


• SIKALNI VOĐENJA (SU KJE SMO KORISTILI $\phi_8 - \phi_{10}$) DOKAZE OBRNUTU, UVIJEK.

POSTUPAK: ČITAMO JEDNAKOST I VIDIHO PRU

$$C5 = I_0 \cdot (\phi_8 + \phi_9)$$

DAKLE $(\phi_8 + \phi_9)$ JE , OUDA TO • JE 

ITERATIVNO IZ JEDNAKOSTI POSLOŽIMO SVE SLOVKE, KA SAMO

KRAJU DODAMO JOŠ JEDNU  JER NIJE I_0 JEDINA INSTRUKCIJA KOJA

• POBUĐUJE Cx

Enon
L non

IAKO JE MODEL TADA BIO PRVOM DRUGAČIJI, MI MIJE MO ZADATK RJEŠAVATI PO NAŠEM MODELU. (AKTUALNO)

PREMAČITE MODEL 8-INSTRUKCIJSKOG PROCESORA SA SLIKE NA KAKU DA DODATE NOVO 9-TU INSTRUKCIJU INDIRKTNOS SKOKA JMPI, KOJA SVOJU ADRESU SKOKANJA DOBJA IZ SVOG OPERANDA. ODREĐITE SEKVENCIJU.

FAZA FETCH: OBAVLJENA

FAZA EXECUTE:

$$\text{ENDR: } C_5 = I_9 \cdot (\phi_8 + \phi_9)$$

$$\text{LADR: } C_{13} = I_9 \cdot (\phi_9)$$

$$\text{READ MEMORY: } C_3 = I_9 \cdot (\phi_{10} + \phi_{11})$$

$$\text{LADR: } C_{14} = I_9 \cdot (\phi_{11})$$

$$\text{ENDR: } C_5 = I_9 \cdot (\phi_{12} + \phi_{13})$$

$$\text{LPC: } C_{11} = I_9 \cdot \phi_{13}$$

BRANJE TRAJE 2 CIKLUSI
LADR TRAJE 1 CIKLUS
} $\text{ADR (AD)} \rightarrow \text{ADR}$

ČITAN MEMORIJE
PUNJEN ADR
} $\text{R (ADR)} \rightarrow \text{MDR}$

PUNJEN NA INTRBUZ
PUNJEN PC
} $\text{MDR} \rightarrow \text{PC}$

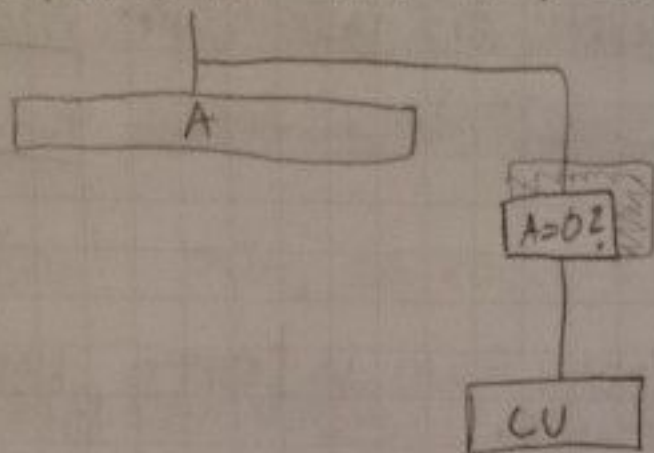
PREINACITE MODEL 8-INSTRUKCIJSKOG PROCESORA SA SLIKE NA NAČIN DA DODATE INSTRUKCIJU UVJETNOG INDIREKTNOG SKOKA JMPNI.

INSTRUKCIJA PRIHVACA GRANANJE AKO JE REZULTAT POSLEDNJE ARITMETIČKE OPERACIJE NEGATIVAN, A ADRESA GRANANJA SE DOBIVA IZ OPERANDA.

ODREDITE JEDNADŽBE SIGNALA KOJI UPRAVLJAJU RADOM JMPNI.

NAŠ MODEL OČITO NEMA SKLOP KOJI ĆE PROJEKTI PREDZNAK PRETHODNE OP. MI SA MOŽEMO PROIZVOLJNO DODATI:

VIDIMO DA NAŠ MODEL IMA VEĆ $A = ?$ SKLOP ZA ISPITIVANJE UVJETA.



← MOŽEMO JOJ DODATI JOŠ FUNKCIONALNOST $Z=1$ AKO REZULTAT NEGATIVAN.

FAZA IZVRŠI

ENDR: $C_5 = I_9 \cdot (\phi_8 + \phi_9)$

LHAR: $C_{13} = I_9 \cdot \phi_9$

READ: $C_3 = I_9 \cdot (\phi_{10} + \phi_{11})$

LHDE: $C_{14} = I_9 \cdot \phi_{11}$

ENDR: $C_5 = I_9 \cdot (\phi_{12} + \phi_{13})$

LPC: $C_{11} = I_9 \cdot \phi_{13} \cdot (N=1)$

EKVIVALENT UVJETU I
 $N=1$ ZADOVOLJEN.

- ODREĐITO PUTA UVJET JE NAČIN KAKO SE DODAJU UVJETI.

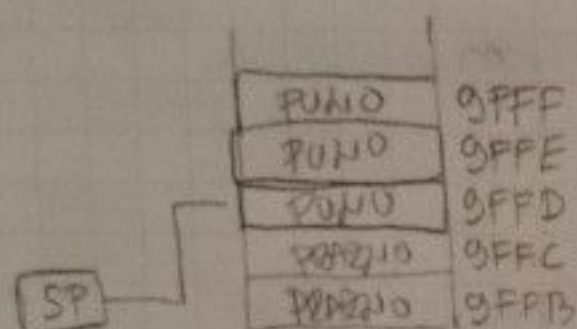
1. PREDISPIT 2012/2013 ZADATAK 3 (10 BODOVA) (27.11.2012)

PREDLOŽITE IZMJENE PUTA PODATAKA I UPRAVLJAČKE JEDINICE 8-BITNOG MODELA PROCESORA KAKO BI MODEL PODRŽAO OSNOVNE RADNJE SA STOGOM.

- PODRŠKA ZA PUSH NAREDBU KOJA SPREMA AKUMULATOR U STOG
- PODRŠKA ZA POP NAREDBU KOJA SA VRHA STOGA SPREMA U AKUMULATOR

NAVEDITE POTPUNE JEDNADŽBE.

ODGOVORNO ZNATI KAKO STOG RADI \Rightarrow DEKREMENT.



UOBICAJENO: STOG RADI PREMA NIŽNIM ADRESAMA,
TE SP SADRŽI ADRESU ZADNJE PUNE

TREBAMO REČINO DA POKAZUJE NA ZADNJU PUNU IZ 9FFD, KAD PUSH-AMO NA STOG ON ĆE POKAZIVATI NA 9FFC, TI NIŽE.

FAZA IZVRŠI OD POP (POVRATKU AKUMULATOR U SP)

PRVO MORAMO SNAJITI SP. NE ZNAMO GDE JE SP. MORAMO OPERACIJU DEC SP.

\Rightarrow HARDVERSKI MORAMO DODATI

1° REGISTAR SP

2° SIGNALI INC SP C20
DEC SP C21

* VEĆ IMAMO 17 REGISTARSKIH
SIGNALA

3° SIGNALI ENABLE SP (ESP) C18
LOAD SP (LSP) C19

INKREMENT/DEKREMENT NE MOŽEMO VADITI PREKO DRUGIH SKLOPOVA JER
SU SVE SLOŽENIJI OD DIREKTOG INC SP/DEC SP REGISTRA.

SADA MOŽEHO PRISTUPITI REALIZACIJI FAZE IZVRŠI ZA POP/PUSH:

● POP (DEC SP, DOHVAĆI SA ADRESA LA KOJU POCAZUJE, POHVAĆI U A)

DEC_SP : $C_{21} = I_9 \cdot \phi_8$ SNAHÍ SADRŽAJ REGISTRA SP

ESP : $C_{18} = I_9 \cdot (\phi_9 + \phi_{10})$ PROPUŠTI SP NA INT BUS

LMAR : $C_{13} = I_9 \cdot \phi_{10}$ STAVÍ TO U MAR

READ M : $C_3 = I_9 \cdot (\phi_{11} + \phi_{12})$ ČTAJ MEMORIJU M(MAR)

LMAR : $C_{14} = I_9 \cdot \phi_{12}$ NARUHI MDR

EMDR : $C_5 = I_9 \cdot (\phi_{13} + \phi_{14})$ PROPUŠTI MDR → INT BUS

● LA : $C_{10} = I_9 \cdot (\phi_{14})$ PUNÍ A

NEBÍMO

PITAT ZA GREŠKU! PITAJ ZA POREDAK KOD PUSH (A, SP ILI SP, A)!

PUSH (IZ A STAVÍ U MDR, IZ SP STAVÍ U MAR, PÍŠÍ UNEN, INC SP)

EA : $C_7 = I_{10} \cdot (\phi_8 + \phi_9)$

LMAR : $C_{14} = I_{10} \cdot \phi_9$

ESP : $C_{18} = I_{10} \cdot (\phi_{10} + \phi_{11})$

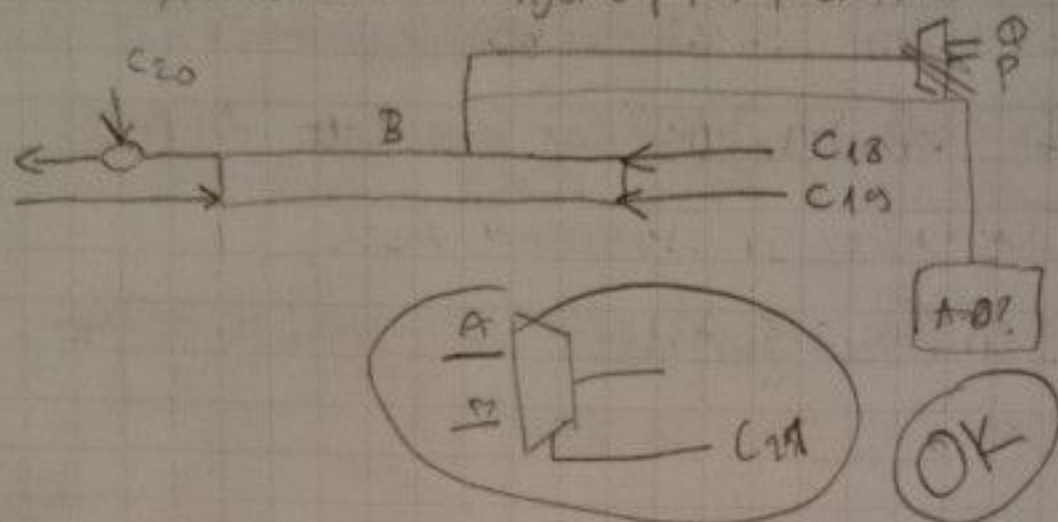
● LMAR : $C_{13} = I_{10} \cdot \phi_{11}$

WRITE : $C_4 = I_{10} \cdot (\phi_{12} + \phi_{13})$

INC SP : $C_{20} = I_{10} \cdot \phi_{13}$

PRETPOSTAVIMO DA MODELU 8-INSTRUKCIJSKOG PROCESORA ŽELIMO DODATI JEDAN REGISTAR OPĆE NAMJENE. (RB). REGISTAR B JE RAVNO PRAVAN SA A KOJE JE PROMJENE NUŽNO IAPRAVITI U INSTRUKCIJSKOJ ARHITEKTURI, PUTU PODATAKA, TE U UPRAVLJAČKOJ JEDINICI!

1° FIZIČKE/HARDVERSKIE RIZJENE / PUT PODATAKA



$$C_{18} = LB$$

$$C_{19} = SHB$$

$$C_{20} = EB$$

PITANJE: ZAŠTO PC NIJE SPOLJN LA SKUP ZA UYET?

PITANJE: ZAR NEKUNA DENUX (Q,P) ISTO INAT NEKI SIBLAC!

2° PROMJENE U INSTRUKCIJSKOJ ARHITEKTURI

TREBA DODATI NOVE INSTRUKCIJE KOJE RADE SA B (LDB X, STB X ...)

3° PROMJENE U UPRAVLJAČKOJ JEDINICI

POTREBNO JE DODATI NOVI DEKODER 4/16 UMJESTO 3/8

POTREBNO JE U KOMBINACIJSKOM DIELU REALIZIRATI DODATNE INSTRUKCIJE

SADA ĆE VJEROJATNO NEKE INSTRUKCIJE TRAJATI DILJE OD ϕ_{16} PA TREBA

I GENERATOR SJEKDOVA MODIFICIRATI

- PRELAŽITE MODEL 8-INSTRUKCIJSKOG PROCESORA NA NAČIN DA ONOGUČITE RELATIVNI ADRRESNI NAČIN ZA NEKORISNE INSTRUKCIJE LOAD/STORE. POTREBNO NAPISATI SVE JEDNAKOSTE KOJE UPRAVLJAJU SA RADOM LDR, STR.

VRSTE ADRRESIRANJA:

IZRAVNO: UZ INSTRUKCIJU IDE ADRESA U OPERANDU

NEIZRAVNO (NEPOSREDNO): UZ INSTRUKCIJU IDE DIREKTNO NUMERIČKI ADRESA

RELATIVNO: INSTRUKCIJA DOBIJE ODMAK U OPERANDU, KOJI OMDA ZBRAJA SA PC, TE TOČINI ODREĐIVANU ADRESU.

KLASIČAN STA RADI (U FAZI IZVRŠI)

STA ADRESA: POHDANJUJE SADRŽAJ AKUMULATORA U ADRESA-U

ENDR	$C_5 = I_x \cdot (\phi_8 + \phi_9)$	}	DOBIVAJA ADRESA U MR
LVAR	$C_{13} = I_x \cdot \phi_9$		
EA	$C_7 = I_x \cdot (\phi_{10} + \phi_{11})$	}	AKUMULATOR U NDR
LADR	$C_{14} = I_x \cdot \phi_{11}$		
WRITE	$C_4 = I_y \cdot (\phi_{12} + \phi_{13})$	}	PIŠI U MEM

RELATIVAN STA STR ODMAK RADI (U FAZI IZVRŠI)

STR ODMAK: PRVO NORA POZBROJAJ PC + NDR (U KUPET JE ODMAK), A ZATIM ODRADITI STORE!

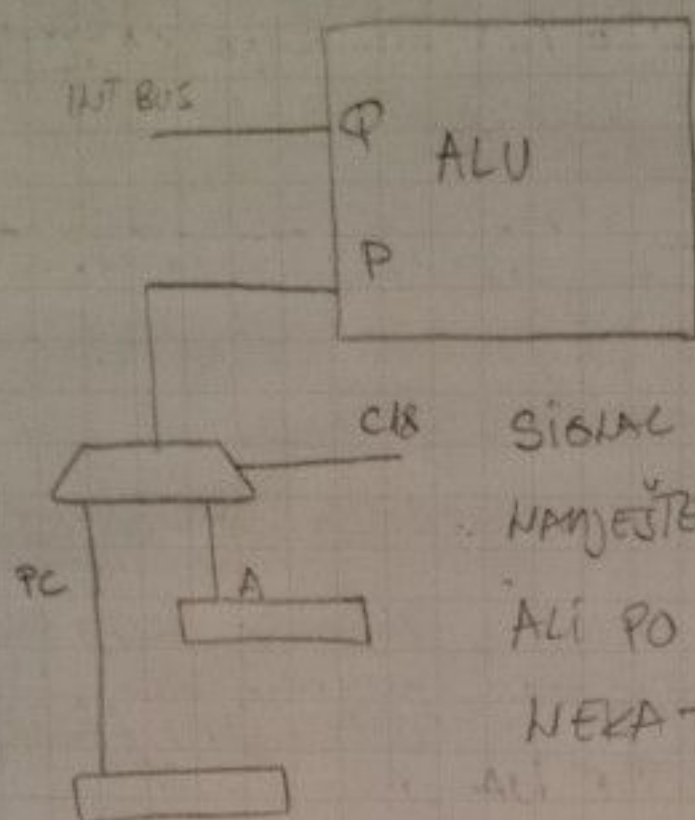
RELATIVNE STORE (U FAZI IZVRŠI) [STR]

ENDR: $C_5 = I_g \cdot (\phi_8 + \phi_9)$

TREBAMO DVEST PC NA ALU, TE OPERAND TAKODER!

- OČITO JE DA PC NIJE SPOJEN SA ALU!!! TREBA PRVO ONOGUČITI DA PC DOĐE PO POTREBI NA ALU UVESTU [A] !!!

MODIFIKOVANÝ MODEL



SIGNAL KOJÍ JE STALNO "PO DEFAULTU"
 NAMĚŘENO DA PROPUŠTĚA A NA P ,
 ALI PO POTŘEBĚ MOŽE PROPUŠTITÍ I PC
 NEKATO BUDE SIGNAL $C18$ (PCP)

ENDR : $C_5 = I_9 \cdot (\phi_8 + \phi_9)$

PCP : $C_{18} = I_9 \cdot \phi_9$

ADD : $C_0 = I_9 \cdot (\phi_{10} + \phi_{11})$

LALU : $C_{15} = I_9 \cdot \phi_{11}$

EALU : $C_6 = I_9 \cdot (\phi_{12} + \phi_{13})$

LNAR : $C_{13} = I_9 \cdot \phi_{13}$

EA : $C_7 = I_9 \cdot (\phi_{14} + \phi_{15})$

LMOR : $C_{14} = I_9 \cdot \phi_{15}$

WRITE : $C_4 = I_9 \cdot (\phi_{16} + \phi_{17})$

PROPUŠTÍ ODNAK NA INT BUS Q

PROPUŠTÍ PC NA P

ZEROJI

PROPUŠTÍ REZULTÁT NA INTERLU REG. ALU

PROPUŠTÍ SA NA INT BUS

UČITAJ SA U NAR

POŠTÍ SADRŽÍAJ IŽ A NA INT BUS

UČITAJ TO U MOR

PÍŠÍ UNEN.

OBČASN LOAD A (POVRATN U A SADRŽÍAJ)

ENDR
 LNAR
 READ
 ENDR
 LA

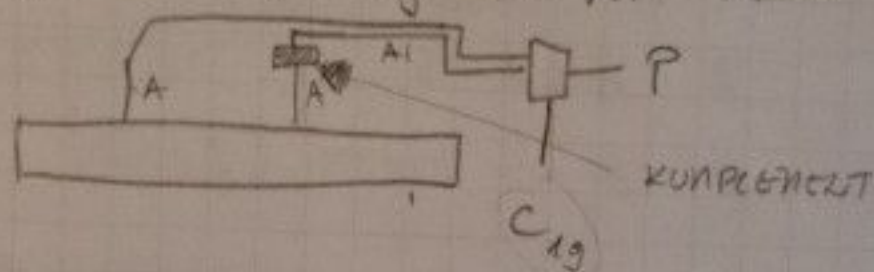
LOAD RELATIVE A (PRVO ZEROJI AX)

ENDR
 PCP
 ADD
 LALU
 EALU
 LNAR
 READ
 ENDR
 LA

MODELU 8-INSTRUKCIJSKOG PROCESORA ŽELIMO DODATI INSTRUKCIJU SUBRA
KOJA OD SADRŽAJA MEMORIJSKE LOKACIJE (X), ODUZIMA SADRŽAJ AKUMULATORA
TE ODA REZULTAT VRAĆA U AKUMULATOR.

PREPOSTAVITI DA JE ALU VEĆ MODIFICIRAN - PRIKLADNO (IMA ALU)
NABROJATI DODATNE MODIFIKACIJE KOJE JE POTREBNO NAČINITI? I
NAPISATI JEDNADŽBE ZA NAVEDENU INSTRUKCIJU.

S OBZIROM DA ODUZIMAMO A, TO ZNAČI DA DODAJEMO KOMPLEMENT.
POTREBAN NAM JE ŠKLOP KOJI ĆE NA ALU SLATI ILI A, ILI



$$E_{HOR} \quad C_5 = I_g \cdot (\phi_{12} + \phi_{13})$$

$$C_{13} = I_g \cdot (\phi_{13})$$

$$C_{12} = I_g \cdot (\phi_{14} + \phi_{15})$$

$$C_{15} = I_g \cdot (\phi_{15})$$

$$C_6 = I_g \cdot (\phi_{10} + \phi_{17})$$

$$C_{10} = I_g \cdot \phi_{17}$$

2. DIO

$$C_9 = I_g \cdot \phi_8 + \phi_3$$

$$C_{13} = I_g \cdot \phi_3$$

$$C_2 = I_g \cdot (\phi_{10} + \phi_{11})$$

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

1. DIO

PRIKAZATI MODIFIKACIJE 8-INSTRUKCIJSKOG PROCESORA U SLUČAJU DA JE POTREBNO ONOGOCITI USPOJNO ADRESIRANJE (OPERAND JE KONSTANTA U TIJEZU INSTRUKCIJE)

KORISTEĆI OVE MOĆNOŠTI NAPIŠATI PROGBANSKI ODSEČAK KOJIM SE SADRŽAJ PODATKA SA \$1000 UMANYJE ZA 5.

$$\boxed{\text{SUBI } X} \Rightarrow \boxed{A = A - X}$$

$$\underline{\text{SUBI } \#5} \Rightarrow A = A - 5$$

PROGRAM :

```

LOAD $1000      // MEM ($1000) -> A
SUBI #5          // A = A - 5
STORE $1000     // A -> MEM ($1000)

```

INSTRUKCIJA :

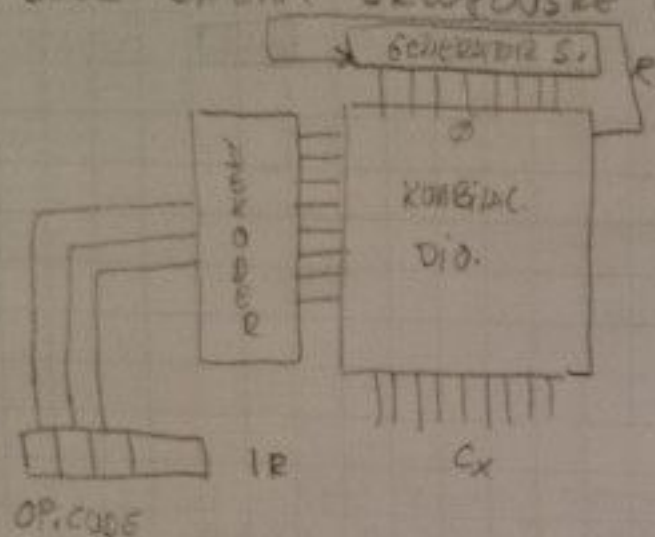
- 1) $E_{IR} \quad C_9 = I_9 \cdot (\Phi_8 + \Phi_9)$
- 2) $SUB \quad C_{10} = I_9 \cdot (\Phi_{10} + \Phi_{11})$
- 3) $L_{ALU} \quad C_{15} = I_9 \cdot \Phi_{11}$
- 4) $E_{ALU} \quad C_6 = I_9 \cdot (\Phi_{12} + \Phi_{13})$
5. $LA : C_{10} = I_9 \cdot \Phi_{13}$

← // #5 IDE NA INTRINS
| PREKO NJE NA Q
// NOVI SIGNAL C₁₀ -

- IAKO JE TADA MODEL 8-INSTRUKCIJSKOG CPU-A BIO NEMICU
 • DRUGAČIJI NI MOŽEMO RIJEŠITI ZADATAK KORISTEĆI NAŠ CPU.

ZA MODEL PROCESORA (POJEDNOSTAVLJENI CISC) NACRTATI BLOK
 SHEMU SKLOPOVNE IZVEDBE UPRAVLAČKE JEDINICE, LOGIČKE
 JEDNADŽBE ZA $LD\ X$ ($LOAD\ X$), NACRTATI KOMBINACIJSKE SKLOPOVE
 ZA REALIZACIJU TIH

1^o BLOK SHEMA SKLOPOVNE UPRAVLAČKE JEDINICE



2^o JEDNADŽBE ZA INSTRUKCIJU

$LOAD\ X$ - POHRANI KONSTANTU X U \boxed{A} ?
 - POHRANI SA ADRESE X U \boxed{A} ?

- TREBA NA ISPITU PITATI ASISTENTA NA ŠTO SE NIJE. PRETPOSTAVLJAM
 DA JE $LOAD$ SA KONSTANTE X U AKUMULATOR A

FAZA PRIJAVI

1. $PC \rightarrow MAR$
2. $n(MAR) \rightarrow NDR$
3. $PC = PC + 1$
4. $NDR \rightarrow IR$

$$EPC : C_8 = I_x \cdot (\phi_1 + \phi_2) \quad PC \rightarrow INTBUS$$

$$LMAR : C_{13} = I_x \cdot \phi_2 \quad INTBUS \rightarrow MAR$$

$$READ : C_3 = I_x \cdot (\phi_3 + \phi_4) \quad READ$$

$$LNDR : C_{14} = I_x \cdot \phi_4 \quad RUM\ NDR$$

$$INCPC : C_{16} = I_x \cdot \phi_5 \quad INC\ PC$$

$$ENDR : C_5 = I_x \cdot (\phi_6 + \phi_7) \quad NDR \rightarrow INTBUS$$

$$LIR : C_{12} = I_x \cdot \phi_7 \quad INTBUS \rightarrow IR$$

$NDR =$ SADRŽI OPERAND!

FAZA IZVRŠI

ENDR :

LMAR :

READ :

LNDR :

ENDR :

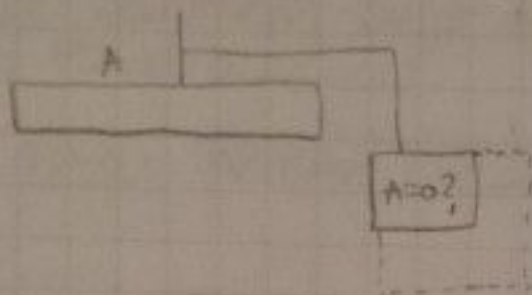
LA :

- Iako je model 8-instrukcijskog CPU-a bio malo drugačiji, možemo riješiti zadatak u skladu sa našim modelom.

Modelu procesora sa skupom od 8 instrukcija dodati devetu instrukciju JHPPN X (IF $AC < 0$ THEN $X \rightarrow PC$).

Sukladno tome modifikirati model. Nacrtati kombilacijski dio za fazu pribavi i izvrši za tu instrukciju.

Postojeći sklop za ispitivanje uvjeta, možemo proširiti da u slučaju da $AC < 0$ da tada postavlja zastavicu



IF $AC < 0$ THEN $N=1$
ELSE $N=0$

Sada kada imamo dodatni sklop možemo izvršiti JHPPN X

1° EDR : $C_5 = I_9 \cdot (\phi_8 + \phi_9)$

Preuzeti adresu X iz EDR-a \rightarrow INT BUS

2° LPC : $C_{11} = I_9 \cdot \phi_9 \cdot (N=1)$

Ako $N=1$, napuni PC sa INT BUS

Faza pribavi za JHPPN X

1. $PC \rightarrow PAR$
2. $n(PAR) \rightarrow NDR$
3. $PC = PC + 1$
4. $NDR \rightarrow IR$

1. EPC : $C_8 = I_9 \cdot (\phi_1 + \phi_2)$
LPAR : $C_{13} = I_9 \cdot \phi_2$

2. READ : $C_3 = I_9 \cdot (\phi_3 + \phi_4)$
LNDP : $C_{14} = I_9 \cdot \phi_4$

3. HPC : $C_6 = I_9 \cdot \phi_5$

4. EDR : $C_5 = I_9 \cdot (\phi_6 + \phi_7)$
LIR : $C_{12} = I_9 \cdot \phi_7$

\Leftarrow možemo iskratiti i

- ! IAKO JE MODEL TADA BIO PONEŠTO DRUGAČIJI, NI MOŽENO ZADATI
 RJEŠITI SUKLADNO JASEN TRIZUTION MODELU.

MODEL PROCESORA SA 8-INSTRUKCIJA PROŠIREN JE TAKO DA SADRŽAVA DVA
 AKUMULATORA, POVEZANA POSREDSNOM DVA SIGNALA:

$$\begin{array}{l} C_{18} - \text{PRIJENOS IZ AKUMULATORA A U B} \\ C_{19} - \text{PRIJENOS IZ AKUMULATORA B U A} \end{array} \quad \left(\begin{array}{l} \text{NAŠ MODEL IMA DO} \\ C_{17} \end{array} \right)$$

POTREBNO JE OSTVARITI DVIJE NOVE INSTRUKCIJE TAB, TBA, KOJE
 OBTARBUJU TU FUNKCIONALNOST.

PRIKAŽITE FAZU PRIBAVI I FAZU IZVRŠI ZA OBE INSTRUKCIJE!!

FAZA PRIBAVI (ISTA ZA OBE INSTRUKCIJE)

1. $PC \rightarrow NAR$
2. $N(NAR) \rightarrow NDR$
3. $PC++$
4. $NDR \rightarrow IR$

$$\Rightarrow \begin{array}{l} EPC : C_8 = I_T \cdot (\phi_1 + \phi_2) \\ LAR : C_{13} = I_T \cdot \phi_2 \\ READ : C_3 = I_T \cdot (\phi_3 + \phi_4) \\ LND : C_{14} = I_T \cdot \phi_4 \\ INC : C_{16} = I_T \cdot \phi_5 \\ END : C_5 = I_T \cdot (\phi_6 + \phi_7) \\ LIR : C_{12} = I_T \cdot \phi_7 \end{array}$$

FAZA IZVRŠI TAB

$$1. C_{18} = I_9 \cdot (\phi_8 + \phi_9)$$

? KOJIKO TRAJE TRANSFER



~~TRAJE~~ TRAJE

SIGORNE BI BILLO 2 TAKTA

FAZA IZVRŠI TBA

$$1. C_{19} = I_9 \cdot (\phi_8 + \phi_9)$$

- Iako se model 8-instrukcijskog procesora tada bio ponovito drugačiji
 • možemo zadatak riješiti pomoću našeg modela.

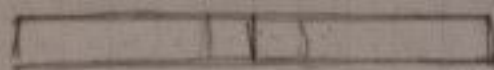
U 8-instrukcijski model procesora čija je upravljačka jedinica ostvarena sklopovskom
 uvodimo dodatnu devetu instrukciju INC n koja inkrementira sadržaj memorije
 lokacije n.

Procesor je izveden tako da je nužno kroz dva ciklusa dohvatiti INC n

- A) SPECIFICIRATI SVE PROMJENE U STRUKTURI SKLOPOVSKE UPRAVLJAČKE JEDINICE
 (+ LACRTATI)
 B) REALIZIRATI KOMBINACIJSKI PRIBAVI ← OK!

OPASKA: INKREMENTIRANJE SE IZVODI U AKUMULATORU A DODANAJEM NOVOG
 DODATNOG SIGNALA C₁₈

- A) 1.) - IR → VOĐIMO 4 LINIJE IZ IR U DEKODER
 2.) - DEC 3/8 → DEC 4/16
 - UVESTI NOVI SIGNAL ZA ALU C₁₈ - INC



- DODATI NOVI RESET AKO JE NOVA
 INSTRUKCIJA KRAĆA OD NAJDUŽE "STARE"

II IAKO JE MODEZ TADA BIO DRUGAČIJ DOLJEKE NI NUŽNO ZADATAK
RIJEŠITI PO IASEN MODELU.

PREPOSTAVITE DA JE MODEZ 8-INSTRUKCIJSKOG PROCESORA PRETVAČEN TAKO DA MU JE
DODAN I AKUMULATOR B. SKUP INSTRUKCIJA OBOGAĆEN JE ZA TAB, TBA (TBA
JE COM B (KOMPONENT B))

DODATNI SIGNALI : C_{18} : TAB

C_{19} : TBA

C_{20} : AKTIVIRANJE SKLOPA ZA KOMPONENT B

NAPRAVITI PRIBAVI, IZVRŠI ZA SVE TRI INSTRUKCIJE, TE NACRTAJTE KOMB. SKLOP

AKO PROGNAITE DA SE RADI O BRZIN" INSTRUKCIJAMA, IZVEDITE I POTREBNE
PRELASE KOJE SKRAĆUJU "INSTRUKCIJSKI CIKLUS.

PRIBAVI

1. PC \rightarrow NDR
2. N(NDR) \rightarrow NDR
3. PC++
4. NDR \rightarrow IR

$$EPC: C_8 = I_x \cdot (\phi_1 + \phi_2)$$

$$LVAR: C_{13} = I_x \cdot \phi_2$$

$$READ: C_3 = I_x \cdot (\phi_3 + \phi_4)$$

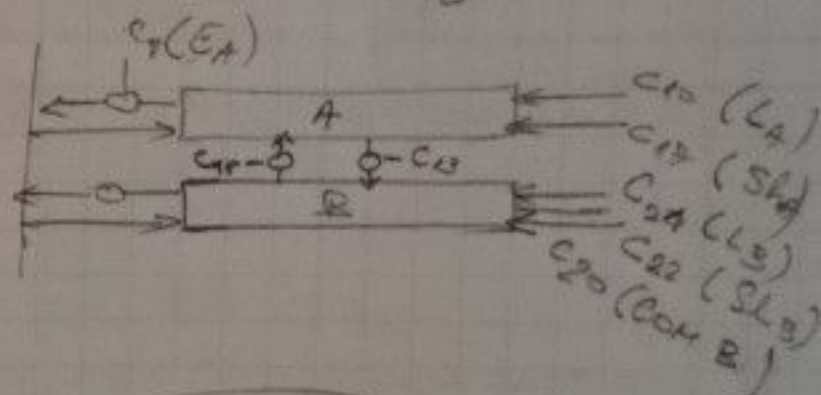
$$LHDR: C_{14} = I_x \cdot \phi_4$$

$$HLC: C_{15} = I_x \cdot \phi_5$$

$$ENDR: C_5 = I_x \cdot (\phi_6 + \phi_7)$$

$$LIR: C_{12} = I_x \cdot \phi_2$$

ISTA ZA SVE TRI INSTRUKCIJE



IZVRŠI TAB $\Rightarrow I_9$

TBA $\Rightarrow I_{10}$

COM B $\Rightarrow I_{11}$

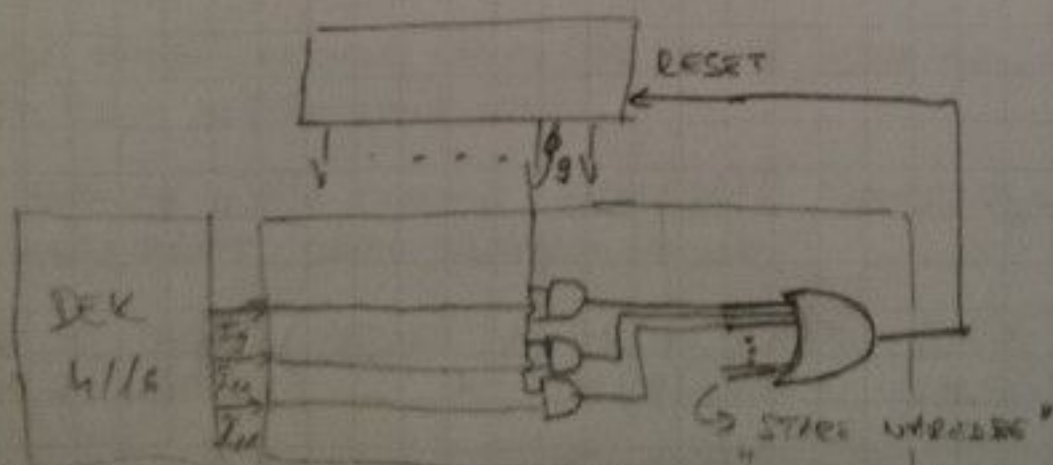
$$TAB: C_{18} = I_9 \cdot \phi_3$$

$$TBA: C_{19} = I_{10} \cdot \phi_3$$

$$COM B: C_{20} = I_{11} \cdot \phi_3$$

SVE TRAJU 8ϕ

u $9. \phi$ RADIMO RESET BROJILA



STARE JARENJE - ONE OD KIH
OKAN, KJE SU BILE LAJENJE
(KRAJE OD NJOJJE)
NOVE JARENJE - ONE TRI SUKRO
TAKOJE DOŠE LA TOJ ILI TOJ