

# 1) PODACI U RAČUNALU

	CJELOBROJNI				
	BEZ PREDZNAKA		S PREDZNAKOM		
	NBC	BCD	BIT 7+ PREDZNAK	1K	2K
OPSEG	$0 \rightarrow 2^N - 1$	- ovisi o pakiranosti i broju brojaka → nepotpuni $0 \rightarrow 10^N - 1$	$-(2^{N-1} - 1) \rightarrow +(2^{N-1} - 1)$	$-(2^{N-1} - 1) \rightarrow +(2^{N-1} - 1)$	$-(2^{N-1} - 1) \rightarrow +(2^{N-1} - 1)$
NEKOSTRAK			$+0, -0$	$\pm 0$	
PRIMER	$25 = 16 + 8 + 1$ $= 00011001$ $-25 \rightarrow$ ne može	$25 = \frac{2}{0010} \frac{5}{0101}$ $-25 \rightarrow$ ne može	$25 = 00011001$ $-25 = 10011001$	$25 = 00011001$ $-25 \Rightarrow 25 = 00011001$ $\rightarrow K = 11100110$	$25 = 00011001$ $-25 \Rightarrow 25 = 00011001$ $\rightarrow K(25) = 11100110$ $+1 = 11100111$

IEEE FORMAT: Broj X se zapisuje u obliku  $X = \text{Mantisa}_2 \cdot 2^{\text{Eksponent}}$

(1. Frakcija)

u 32 bit IEEE:

$$X = (-1)^P \cdot 1.F \cdot 2^{K-127}$$

Predznak	Funkcijsko	
P	K	F
31	30-23	22-0
0 = +	15K5254	FUNKCIJA MANTISE
1 = -	K = EKSPONENT + 127	
	→ EKSPONENT = K - 127	

1)  $K=0, F=0 \rightarrow$  broj = 0

2) Podjevu > predalek broj za prikaz, Eksponent > 127

$K=255, F=0$

3) Podjevu > premalen broj za prikaz,  $E < -126$

$K=0, F \neq 0$

4) Neispravan ili NaN  $K=255, F \neq 0$

BF2000016  
= 0111111101000000...0

P=1  
K=11610 → E=K-127=-11

F=010...0 → MANTISA=1,01

= 1,25<sub>10</sub>

$X = (-1)^1 \cdot 1,25 \cdot 2^{-1} = -0,625$

7245,125 → 32 IEEE

7245,125(10) = 1110001001101,001(2)

→ Prikaz (E=0) = 1,11...1001(2)  $\cdot 2^{\frac{12}{2}}$

E=12 → K=E+127=139 = 10001011

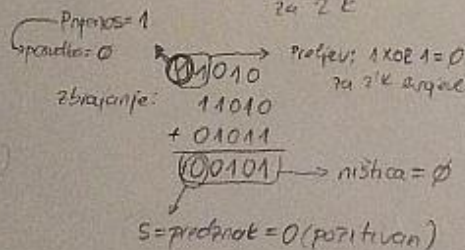
P K F = 0100101111...11001

\* Posirenje nula i jedinica → kod FRISA, proširenje (predznačno)

→ NBC čulo iznos

## ZASTAVICE

- Prijenos: C
- Prejevu: OV
- Predznak: S
- ništa: 0
- posudba: b ili not c



OPERACIJA	C	
	+	-
NBC	C=1 C=0	
2K	OV=1 OV=1	

ROTACIJE LSL - logički pomak u lijevo 0110101 → punimo desno 0 → 1010100

- odgovara množenju s 2 za NBC lijeve odnosno za N mjesta do 2<sup>N</sup>

LSR - -11- u desno → punimo s lijeve strane s 0.

- dijeljenje s 2 odnosno 2<sup>N</sup>

ASR - aritmetički pomak u desno → s lijeve strane punimo redovnim vrijednostima najvišeg bita, npr 10110 → 110110

→ dijeljenje za 2<sup>K</sup> lijeve

→ rotacije → ulazni bit uzme vrijednost izlaza li. Rotacije kroz zastavice → na ulazni bit prišemo stanje zastavice  
a izlazni bit odmah u zastavicu (zastavica prijenosa c)



# RAZBOJ IČK 1.1.1

Procesor izvrši programe. Program je zaron u formi naredaba koje su zapisane u memoriji.

3 faze izvršenja: - Pohvat, dekodiranje, izvršenje

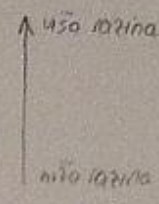
razne opštice - sistemska (ne trebamo mnogo znati o arhitekturi)

→ skupa naredaba - su podaci o procesoru i vanjskim jedinicama

→ mikroarhitektura - opću načine povezivanja i signale

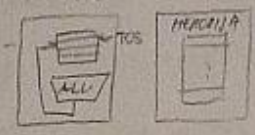
- gate level - + kašnjenje sklopa

- layout level - fizički postupak kod proizvodnje



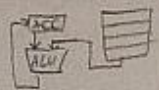
- podaci u memoriji: - 2 strukture - Von Neumann i Harvardski

- ovisno o saopštenju operandi: - stogovne arhitekture



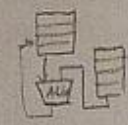
- 2 operanda na vrhu stoga  
- put do čitanja iz memi, put do pisanja

- akumulatorska arhitektura



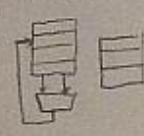
→ uvijek jedan operand u posebnom registru  
- drugi iz memorije, rezultat u → ACC

- reg-mem



→ jedan operand iz registra, drugi iz memorije, rezultat u registru

- reg-reg



→ oba operanda iz registra, rezultat u registru, LOAD → STORE

- CISC, RISC

→ sama sadržine: 32b. Adresa u memoriji iznosi 1 bajt. Program counter (PC)

→ skup naredaba: → aritmetičko-logičke - ADD, SUB, MUL, OR, XOR

naredba OP1, OP2, OP3 → OP1, OP2 → OP3

OP kod	---	dst1	src1	src2	---
31-27	16	20-23	12-20	19-17	18-0

→ memorijste: LOAD, STORE

LOAD/STORE registar (naredba)

→ konstante: KONST DW BROJ

oprac. kod	---	dst1	---	dst2	---
31-23	16	15-23	22-10	19-0	

→ upravljačke: JP adresa → bezuslovni skok  
JP usjet adresa

↳ 2 instance → 127 slojeva

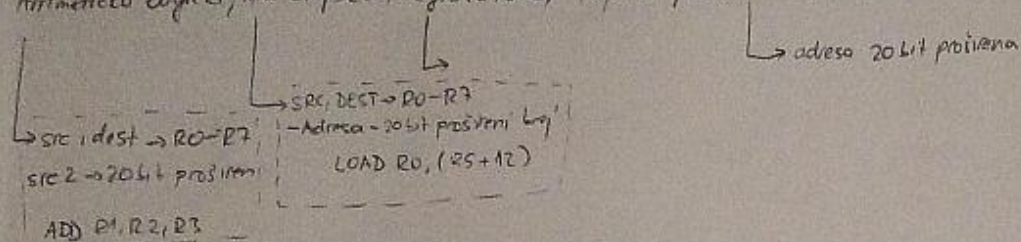
OP kod	---	cond	---	dst1	dst2
31-27	16	15-23	24-10	19-0	

→ strojni kodovi naredbi

→ 18b, podjedini

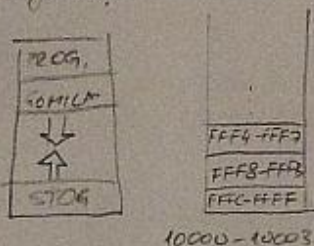


Aritmetičko logičke, memorijske, registarske, upravljačke (službeni šalabahter)



- potprogrami → pozivaju se sa `CALL`, povratat sa `RET`
- pri svakom pozivu potprograma: spremi se PC na stog i skoči na mjesto labela  
PC+4 → sljedeća adresa na stog
- `HALT` zaustavlja procesor
- ⇒ `REGISTARSKO` - podatak ili rezultat se nalazi u jednom od registara
- ⇒ `NEPOSREDNO` - podatak je zadan kao broj (ili labela)
- ⇒ `APSOLOTNO` - broj predstavlja adresu podatka, u memoriji, ili adresu određene skok

Potprogrami:



`CALL R2, -1, R7` → u funkciji

Prijenos registrima: → nema spremanja na memorijske lokacije, već se obavlja u registrima.  
→ problem ograničenih registara i pregaženja njihovih vrijednosti.  
→ brzo i jednostavno, ali se često kao međurezultati, nema rekurzije (zbog SP-a)

Prijenos fiksnim lokacijama: → nema ograničenja na broj parametara

→ spor, rade, duži programi, nema rekurzije

Prijenos stogom - omogućuje rekurziju

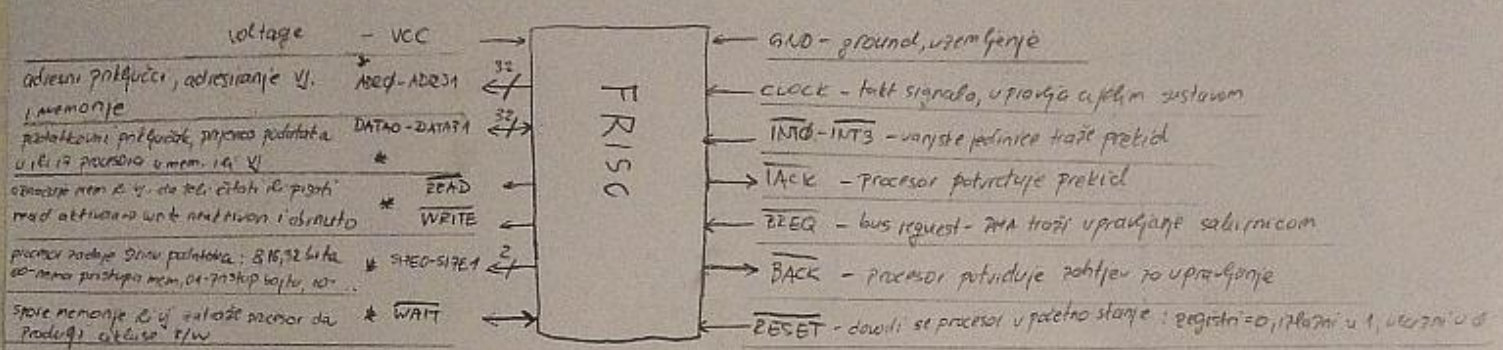
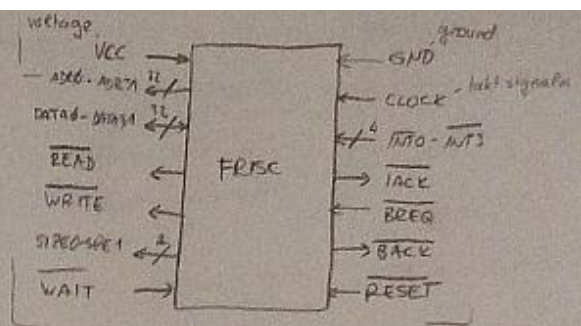


Sadržaj: - adretna sabirnica  
- priključci: - podatkovna  
- upravljačka

- aktivni visoki nivo  
READ  
TREQ

- jednostruki ili grupirani

- ulazni  
- izlazni  
- dvosmjerni



\* povezan s memorijom

Protokoli: čitanje - 1 takt clocka  
Pisanje - 1 takt clocka

čitanje sponh memorija - dodatni ciklus čitanja: TW  
Pisanje u sponh memorije

→ nastupi budi aktivira procesor. Radi se o čitanju ili pisanju, procesor u tom trenutku mora postaviti adresu gdje će biti pisati ili čitati (odnosno na adresnu sabirnicu poslati adresu). Memorija dekodira adresu na tom budiu, te dobiva podatke s te adrese. Mora postaviti podatak prije padajućeg brida, a ako se to ne dogodi, ona mora zatvoriti WAIT odnosno poslati dodatni ciklus clocka. Na padajućem briedu se preuzima podatak.

→ Proširenja naredaba: AL-naredbe (231. str.) → 02-FRISC-arhitektura

REG-naredbe (256. str.)

MEM-naredbe (293. str.)

(295. str.) - LOAD i STORE s BROJEM - LOAD R1, (1000) - LOAD R1, (1000+X)  
(298) - PUSH i POP STORE R1, (1000) - STORE R1, (1000+X)

UPB-naredbe - (334) - JP, JP

(336) - CALL, RET

→ 341. do kraja !!!

→ su podaci se proširuju !!!

→ načini adresiranja: REGISTARSKO - podatak ili rezultat se nalazi u jednom od registara:

1.6.06/07
3. 07/08
2. '09

NEPOSREDNO - podatak je neposredno kao broj

ABSOLUTNO - adresirajte je broj (ili adresirajte skoku), kopirajte: može pisati 'balela'!

RELATIVNO - samo kod JP

REGISTARSKO INDIJEKTNO → u upravljačkim naredbama, registri u zagradama

REGISTARSKO INDIJEKTNO S ODRATOM → u memorijskim naredbama, registri u zagradama ± odmah  
→ ! može biti 0

IMPLICITNO → samo kod PUSH i POP



Rad s bitovima ponoviti! → str. 66. 03. FRISC - programiranje

→ postavljnje bitova: OR s maskom

0 u maski → ti se bitovi ne mijenjaju

1 u maski → ti se bitovi mijenjaju

$$\begin{bmatrix} 0 \text{ OR } X = X \\ 1 \text{ OR } X = 1 \end{bmatrix}$$

→ brisanje bitova: AND s maskom

0 u maski → ti se bitovi mijenjaju

1 u maski → ti se bitovi ne mijenjaju

$$\begin{bmatrix} 0 \text{ AND } X = 0 \\ 1 \text{ AND } X = X \end{bmatrix}$$

→ komplementiranje bitova: XOR s maskom

0 u maski → bitovi se ne mijenjaju

1 u maski → bitovi se komplementiraju

→ pozor! na proširanje s 20 na 32 bita! → str. 70 03. FRISC