

Aquaman_Edition_ARH2

Par napomena vezano za ovaj dokument:

- (1.) dokument je sken mojih postupaka prilikom rješavanja zadataka koji su se pojavljivali na ispitima,
- (2.) dokument je sistematično napravljen da slijedi sljedeći princip (poredani po godinama i po redoslijedu ispita u svakoj godini),
- (3.) uložen je trud da se riješe zadaci koju su rješavani u već postojećim fajlovima, ali su dodani i novi zadaci koji još nisu rješavani,
- (4.) neki zadaci su skenirani iako nisu rješavani, ali u oznakama je napisano kojem zadatku su slični pa da se može pogledati,
- (5.) u postupcima sigurno ima pogrešaka pa sadržaj materijala nemojte uzimati "zdravo za gotovo".

Uglavnom, svrha materijala je da pomogne barem jednoj osobi, fajl sadrži dosta korisnih stvari i poprilično je uredan i sistematičan pa se nadam će ispuniti svoju svrhu.

Sretno s pripremom za ispite! :)

VI, 2009./2010.

①

$$C = 8192 \quad B = 2^{13} \quad B$$

16-bitna adresna $\rightarrow A_0 - A_{15}$

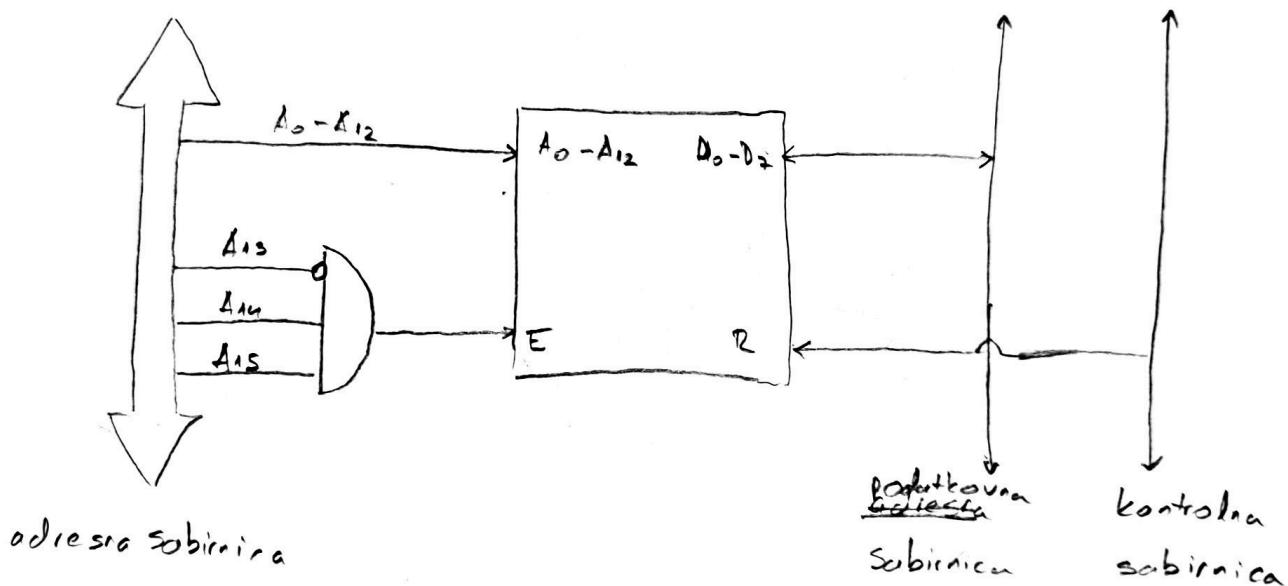
8-bitna podatkovna $\rightarrow D_0 - D_7$

- Pocetna adresa: \$c000 \equiv

| | |
|------|----------------|
| 1100 | 0000 0000 0000 |
|------|----------------|
- Završna adresa: \$dff f \equiv

| | |
|------|----------------|
| 1101 | 1111 1111 1111 |
|------|----------------|

A_{15}, A_{14} i A_{13} su fiksni za ovaj raspon adresa pa ćemo njih dovest u E kako bismo osigurali da E radi samo za ovaj raspon



kod bi procesor izveo instrukciju sta \$cdet došlo bi do greske jer se u ROM memoriju ne može upisivati

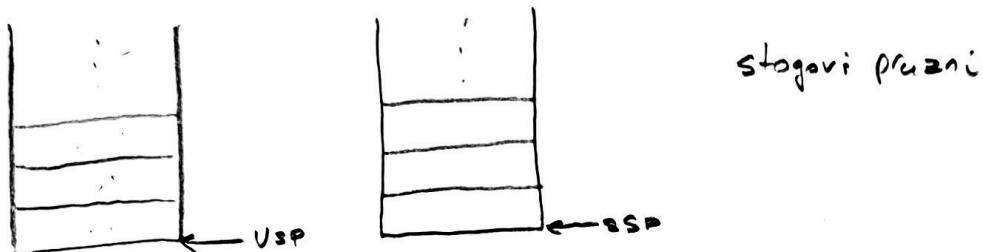
①

MI, 2009./2010.

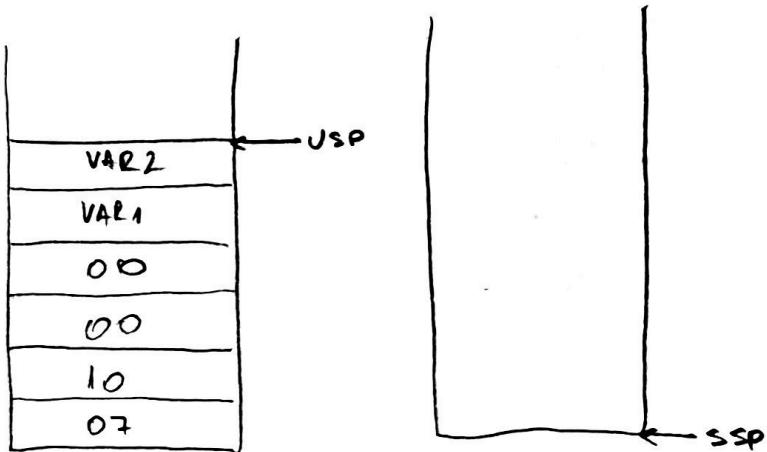
② MC68000

- \$1000: jsr \$2000
- pretpostavka: $PC + \text{operand} = 6 \text{ bajtu}$
- potprogram koristi dvoje memoriske lokalne varijable širine po 1B
- nakon instr. potp. \$2100: move d0,d1, procesor prima zahtjev za prekid te se poziva prekidni potpr. na adresi \$ff0a0

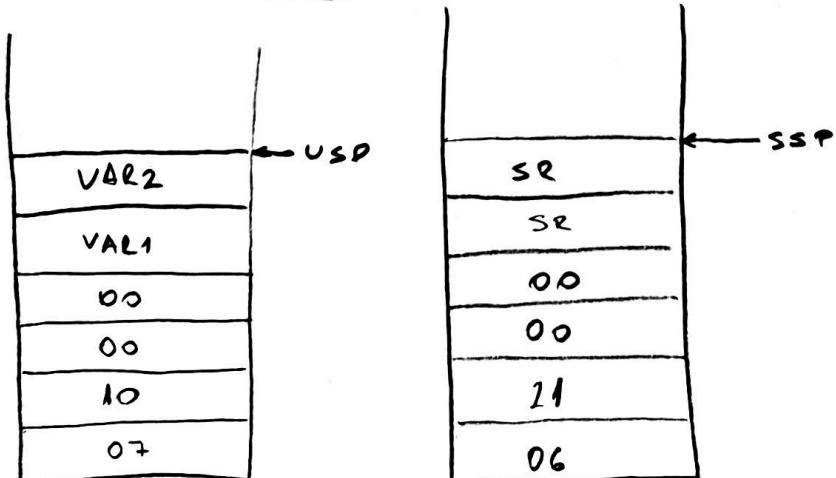
(a) Stanje stogova prije poziva potprograma



(b) neposredno prije primanja prekida



(c) tijekom obrade prekida



U1, 2009./2010.

- ③ program:
\$0010: inc \$13
\$0012: lda \$80

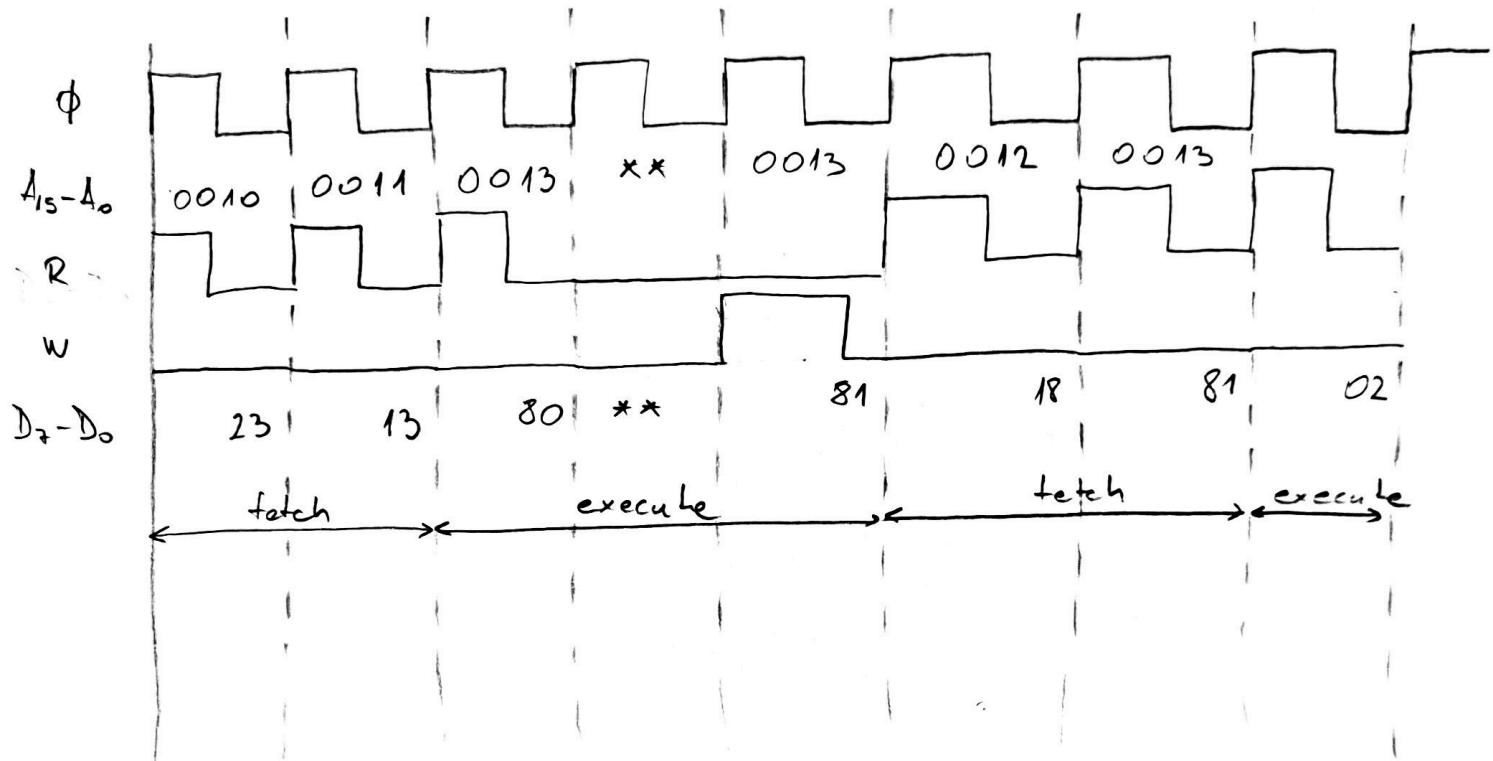
zadano: inc(\$23)
lda(\$18)

sadržaj radne memorije:

\$007f: \$00

\$0080: \$01

\$0081: \$02



$\Rightarrow \$13$ se u DC zapisuje kao 0013, isto tako i 0080 jer se može izravno adresirati nulta stranica.

MI, 2009./2010.

④ - dodati instrukciju uvjetnog indirektnog skoka jump

- odrediti jednadžbe signala koji upravljaju pribavljanjem i izvođenjem instrukcije jump

PRIJAVA:

① $MAR \leftarrow PC$

$$\rightarrow EPC: C_8 = \phi_1 + \phi_2$$

$$\rightarrow LMAR: C_{13} = \phi_2$$

② $MDR \leftarrow M[MAE]$

$$\rightarrow READ: C_8 = d_3 + d_4$$

$$\rightarrow LMDR: C_{14} = \phi_4$$

③ $PC \leftarrow PC + 1, IR \leftarrow MDR[31:24]$

$$\rightarrow INC: C_{16} = \phi_5$$

$$\rightarrow EMDR: C_5 = \phi_5 + \phi_6$$

$$\rightarrow LIR: C_{12} = \phi_6$$

④ dekodiranje operacijskog koda

$$\rightarrow \phi_7$$

Izvrši:

① $PC \leftarrow MDR[23:0]$, ali samo ako $A < 0$

$$\rightarrow EMDR: C_5 = I_8 \cdot (\phi_2 + \phi_6)$$

$$\rightarrow LPC: C_{11} = I_8 \cdot \phi_9 \cdot (A < 0)$$

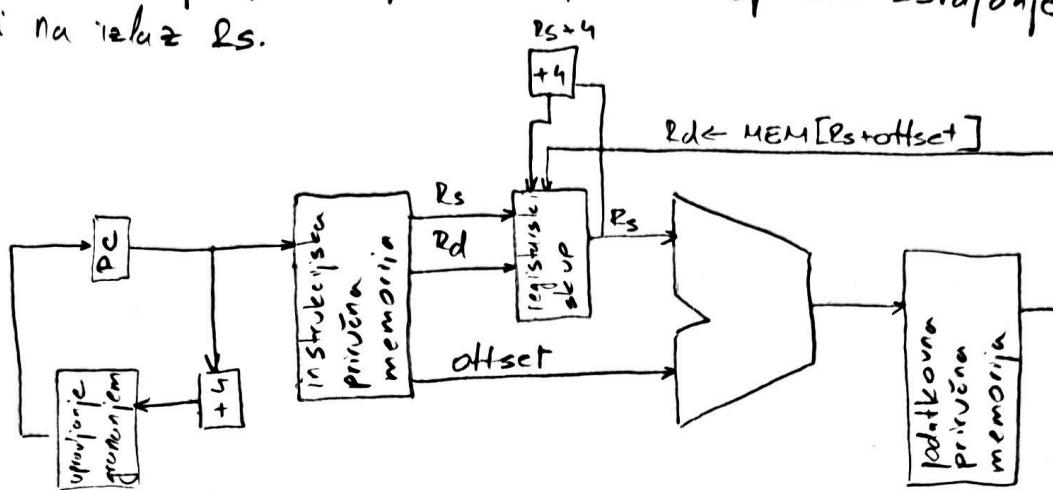
21.2009./2010.

2. arhitektura MIPS

- lwinc Rd, offset, Rs # $Rd \leftarrow \text{MEM}[R_s + \text{offset}]; R_s = R_s + 4$
- addmem Rd, offset, Rs # $Rd \leftarrow R_d + \text{MEM}[R_s + \text{offset}]$

Za prvu instrukciju potrebno je dodati jedan sklop za zbrajanje koji će moći spojiti na izlaz Rs.

(a)



(b)

21.2009./2010.; slično LjIR, 2011./2012. (4. zadatak)
i slično 21.2011./2012. (2. zadatak)

③ arhitektura MIPS

- $\$t_1, \$t_3, \$t_0$ su registri opće namjene

LOOP:

| | |
|-------------------------------------|-----------------------|
| <code>lw \$t_0, 0(\$s_1)</code> | # load word |
| <code>addi \$t_0, \$t_0, 25</code> | # add immediate |
| <code>sw \$t_0, 0(\$s_1)</code> | # store word |
| <code>addi \$s_1, \$s_1, 4</code> | # add immediate |
| <code>bne \$s_1, \$s_3, Loop</code> | # branch if not equal |

(a) skicirati kod u C-u:

```
int *s1 = &start;
int *s3 = &end;
while (*s1 < *s3) {
    int t = *s1;
    t += 25;
    *s1 = t;
    s1++;
}
```

(b) kako kod bi bio generiran za procesor s dvostrukim statickim izdavanjem kojeg istovremeno može izraditi: • jednu instrukciju tipa ALU/branch, • jednu instrukciju tipa load/store

| ciklus | alu/branch | lw/sw |
|--------|-------------------------------------|---------------------------------|
| 1. | nop | <code>lw \$t_0, 0(\$s_1)</code> |
| 2. | <code>addi \$t_0, \$t_0, 25</code> | nop |
| 3. | <code>addi \$s_1, \$s_1, 4</code> | nop |
| 4. | <code>bne \$s_1, \$s_3, Loop</code> | <code>sw \$t_0, 0(\$s_1)</code> |

$$CPL = \frac{h}{5}$$

21.2009./2010.

④ 32-bitna adresna sabirnica

$$a=4$$

$$n=256$$

$$b=32 \text{ B}$$

$$\rightarrow \text{netto-veličina-PH} = s = n \cdot b = 256 \cdot 32 \text{ B} = 8192 \text{ B}$$

$$w(i) = \log_2(n/a) = 6$$

$$w(p) = \log_2(b) = 5$$

$$w(o) = 32 - 5 - 6 = 21$$

* minimalni broj servisnih bitova = 2 (bitovi v i D; Valid i Dirty)

$$\Rightarrow \text{ukupan-broj-bitova-za-realizaciju-PH} = [w(o \neq nula) + \text{servisni-bitovi} + 32 \cdot 8] \cdot 256 \\ = [21 + 2 + 32 \cdot 8] \cdot 256 = (71424 \text{ bitova})$$

$$\Rightarrow \text{traženi omjer} = \frac{8192 \cdot 8}{71424} = 0.91756$$

21.2009./2010.

(5) 8-bitna adresna i podatkovna sabirница

$$DH: a=2, n=8, b=8 \text{ B}$$

[0,1,0,2,0,3,0,2]

pristupi:

| | | |
|------|-----------|---------|
| \$a2 | 1010 0010 | promaći |
| \$65 | 0110 0101 | promaći |
| \$89 | 1000 1001 | promaći |
| \$ca | 1100 1010 | promaći |
| \$54 | 0101 0100 | promaći |
| \$30 | 0011 0000 | promaći |
| \$bf | 1011 1111 | promaći |
| \$18 | 0001 1000 | logodak |

strukturna adrese:

$$w(i) = \log_2(n/a) = 2$$

$$w(p) = \log_2(b) = 3$$

$$w(o) = 8 - 2 - 3 = 3$$

FORMAT ADRESE:

oooiiippp

stanje DM prije pristupanja adresama:

| indeks | oznaka1 | oznaka2 |
|--------|---------|---------|
| 00 | 000 | 001 |
| 01 | 000 | 010 |
| 10 | 000 | 011 |
| 11 | 000 | 010 |

MI, 20.11. / 2012.

①

(a)

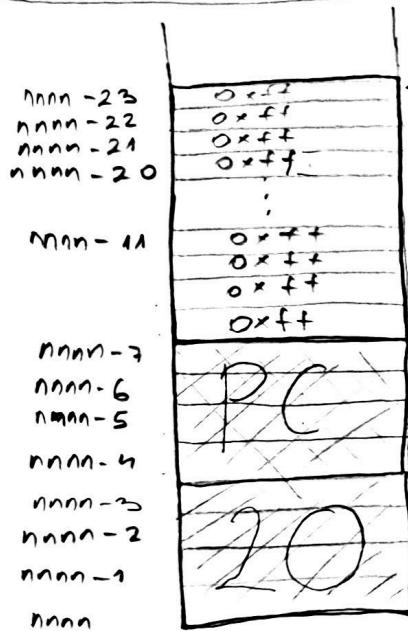
učebíček A

adrese

nnnn-4
nnnn-3
nnnn-2
nnnn-1
nnnn

Užívateľské kazety a logy
nnnn
32-bitový register A7

bajt



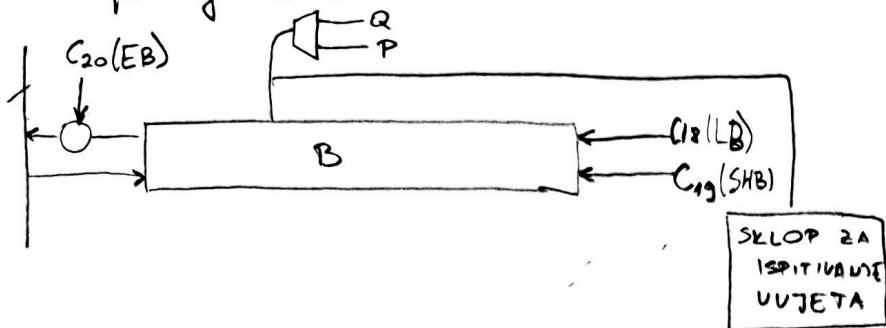
(b) učebíček B

④

② Potrebne promjene:

(a) na putu podataka; fizička promjena:

→ dodavanje registra B



(b) promjene u instrukcijskoj arhitekturi:

→ treba projektirati nove instrukcije; npr. add RA, RB, ...

(c) promjene u upravljačkoj jedinici:

→ budući da imamo više od 8 instrukcija, trebat ćemo umjesto dekodera 3/8 uvesti dekoder 4/16

MI, 2011./2012.

6. • HC68000

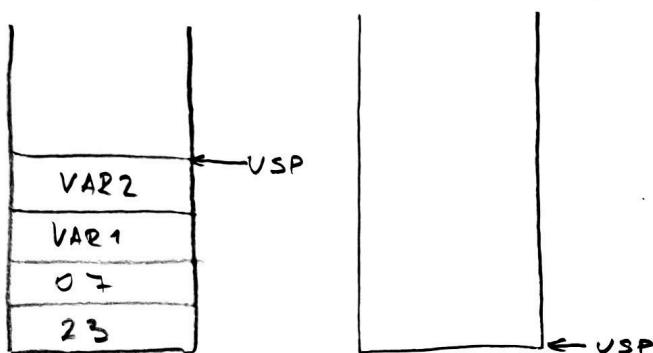
• poziv potprograma; \$2300: jsr \$3200

...

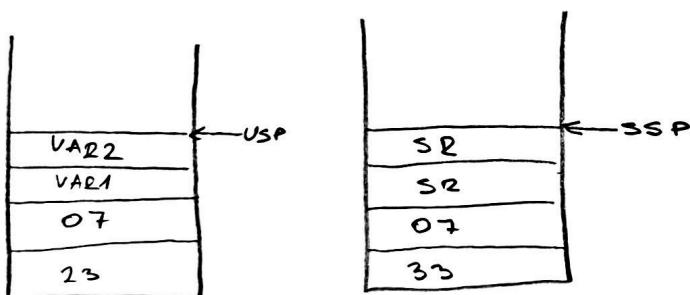
(a) stanje stoga prije poziva potprograma



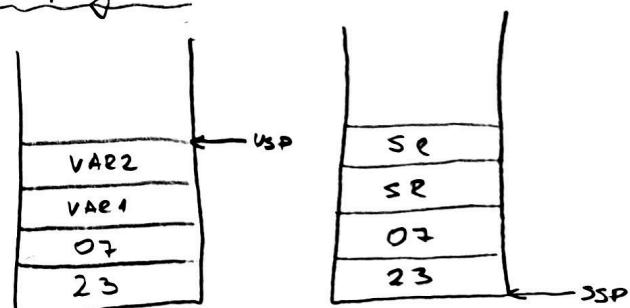
(b) stanje stoga neposredno prije primanja prekida



(c) tijekom obrade prekida



(d) nakon povratka iz prekidnog
potprograma



21.2011./2012.

① $I_{du} \neq \text{konst}(A)$

① $MAR \leftarrow A + MDR[23:0]$

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

$$\rightarrow ADD: C_0 = I_g \cdot (\phi_8 + \phi_9)$$

$$\rightarrow LALU: C_{15} = I_g \cdot \phi_9$$

$$\rightarrow EALU: C_6 = I_g \cdot (\phi_{10} + \phi_{11})$$

$$\rightarrow LMAR: C_{13} = I_g \cdot \phi_{11}$$

② $MDR \leftarrow M[MAE]$

$$\rightarrow READ: C_3 = I_g \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow LHDR: C_{14} = I_g \cdot \phi_{13}$$

③ $A \leftarrow MDR$

$$\rightarrow EMDR: C_5 = I_g \cdot (\phi_{14} + \phi_{15})$$

$$\rightarrow LA: C_{10} = I_g \cdot \phi_{15}$$

21.11.2011./2012.; slično LjIR, 2011./2012. (4. zadatak)
; slično 21.12.2009./2010. (3. zadatak)

② loop:
lw r1, 0(r5) # r1 = r5

nop

sw 0(r6), r1 # r6 = r1

add r5, r5, 4 # r5 += r5

add r6, r6, 4 # r6 += r6

bne r5, r7, loop # while r5 != r7

nop

- Programski odsječak kopira integere s jedne memorističke lokacije na drugu.
- Optimalan raspored instrukcija za procesor s jednostrukim izdavanjem:

loop: lw r1, 0(r5)

add r5, r5, 4

sw 0(r6), r1

bne r5, r7, loop

add r6, r6, 4

- Optimalan raspored instrukcija za procesor s dvostrukim statickim izdavanjem:

| CIKLUS | ALU/BRANCH | SW/LW |
|--------|------------------|--------------|
| 1. | nop | lw r1, 0(r5) |
| 2. | add r5, r5, 4 | nop |
| 3. | nop | sw 0(r6), r1 |
| 4. | bne r5, r7, loop | nop |
| 5. | add r6, r6, 4 | nop |

$\Rightarrow 5 \Delta T$

21.2011./2012.

- ③
- učestalost promušaja L1, L2: 3%, 0,4%
 - vrijeme pristupa L1, L2, RAM: 1T, 10T, 100T



- idealan slučaj: vrijeme pristupa = $N \cdot T$
- slučaj zustaja prirodnih memorija:
vrijeme pristupa = $0,97 \cdot N \cdot T + 0,026 \cdot N \cdot 10T + 0,004 \cdot N \cdot 100T$
= $1,63NT$
- vrijeme pristupa raste za $\frac{1,63NT}{NT} = 1,63$ puta, tj. performansa je pola za 63%.

21.2011./2012.

④

$$a=4$$

$$S = 1 \text{ kB} = 1024 \text{ B}$$

$$b = 64 \text{ B}$$

32-bitne adrese

bojtna zrnatost

→ sljed pristupa PM:

| | | |
|-------------|---|-----|
| \$0a00c00a | 0000 1010 0000 0000 1100 0000 0000 1010 | M |
| \$0a001832 | 0000 1010 0000 0000 0001 1000 0011 0010 | M |
| \$0a001c05 | 0000 1010 0000 0000 0001 1100 0000 0101 | M |
| \$0a001814 | 0000 1010 0000 0000 0001 1000 0001 0100 | H |
| \$0ff0740.5 | 0000 1111 1111 0000 0111 0100 0000 0101 | M |
| ... | ... | ... |

⑤ isteban račun

$$S = n \cdot b$$

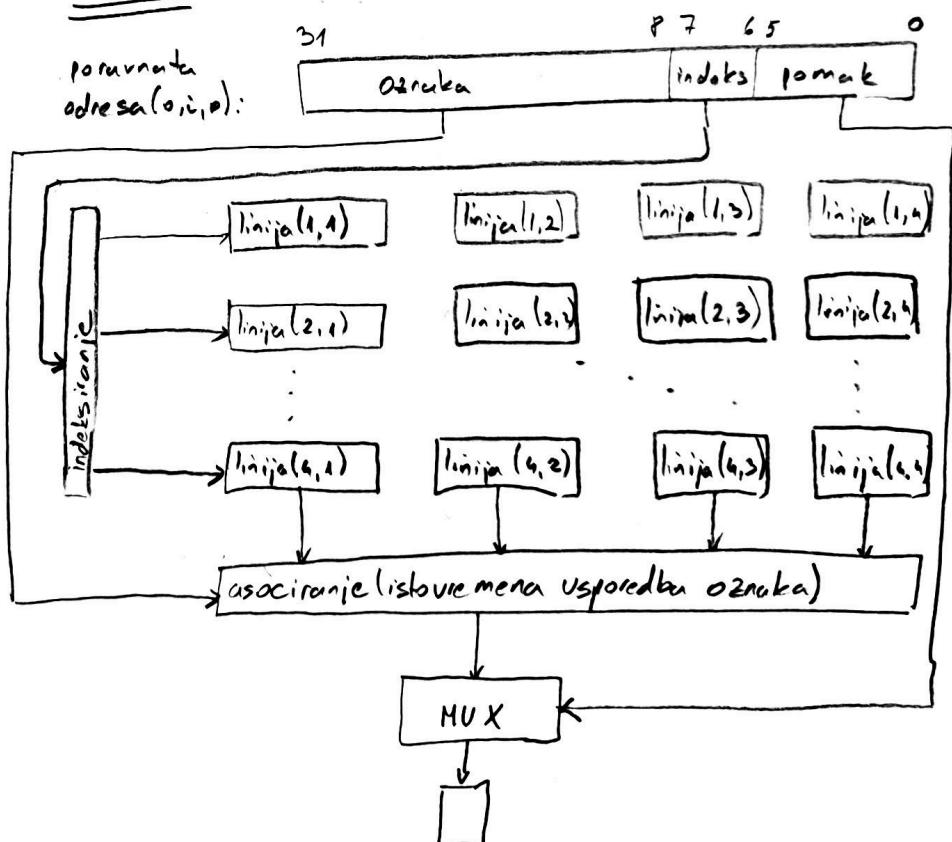
$$n = \frac{S}{b} = \frac{1024 \text{ B}}{64 \text{ B}} = 16 \text{ linija}$$

$$w(p) = \log_2 b = \log_2 64 = 6$$

$$w(i) = \log_2 (n/b) = \log_2 2^2 = 2$$

$$w(o) = 32 - 6 - 2 = 24$$

SKICA:



ZI, 2011./2012. i slično ZI, 2016./2017. (5.zadatak)
slično ZI, 2021./2022. (5.zadatak)

6. 4 dijetre (D_1, D_2, D_3 i D_4)

- procesor s tri protične strukture (visoko specijalizirane)

| | | | |
|---|---|---|--|
| 1 | 1 | | |
| 1 | 1 | | |
| 1 | | | |
| 1 | 1 | 1 | |
| 1 | 1 | | |

| | | | |
|---|---|---|--|
| | | 2 | |
| | 2 | 2 | |
| 2 | | | |
| | | | |

| | | | |
|---|---|---|--|
| | | 3 | |
| 3 | 3 | 3 | |
| 3 | 3 | | |
| | | | |

| | | |
|---|---|---|
| 4 | 4 | |
| 4 | 4 | |
| | | |
| | | 4 |

(a) superskalarni model procesora

| | | | |
|---|---|---|---|
| 1 | 1 | | |
| 1 | 1 | | |
| 1 | | | |
| 1 | 1 | 1 | |
| 1 | 1 | | |
| | | | |
| 2 | | | |
| | 2 | 2 | |
| 2 | | | |
| | | | |
| | 3 | | |
| | | | |
| 3 | 3 | 3 | |
| 3 | 3 | | |
| 4 | 4 | | |
| 4 | | | |
| | | | |
| 4 | 4 | | |
| | | | |
| | | | 4 |

✓ (b) višedretveni (finskovalni) model procesora (MT)

| | | | |
|---|---|---|--|
| 1 | 1 | | |
| | | 2 | |
| | | 3 | |
| 4 | 4 | | |
| 1 | 1 | 2 | |
| | | 2 | |
| 3 | 3 | 3 | |
| | | 6 | |
| 1 | | | |
| 2 | | | |
| 3 | 3 | | |
| | | 4 | |
| 4 | 4 | | |
| 1 | 1 | 1 | |
| | | | |
| 4 | | | |
| 1 | 1 | | |

21.11.2011./2012.; slično 21.10.2016./2017. (5.zadatak)
i slično 21.10.2021./2022. (5.zadatak)

6.-nastavak

(c) specijalizirani model procesora koji podržava simultanu višedrevenost (SUT) ?

| | | |
|---|---|---|
| 1 | 1 | 2 |
| 4 | 4 | 3 |
| 1 | 1 | 4 |
| * | 2 | 2 |
| 3 | 3 | 3 |
| 1 | 4 | 4 |
| 2 | * | * |
| 3 | 3 | 4 |
| 1 | 1 | * |

| | | |
|---|---|---|
| 1 | 1 | 2 |
| 4 | 4 | 3 |
| 1 | 1 | 4 |
| 1 | 2 | 2 |
| 3 | 3 | 3 |
| 2 | 4 | 4 |
| 3 | 3 | 1 |
| 1 | 1 | 4 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

(d) nespecijalizirani model procesora koji podržava simultanu višedrevenost (SUT) ?

| | | |
|---|---|---|
| 1 | 1 | 2 |
| 3 | 4 | 4 |
| 1 | 1 | 2 |
| 2 | 3 | 3 |
| 3 | 4 | 1 |
| 2 | / | / |
| 3 | 3 | 4 |
| 4 | 1 | 1 |
| 1 | / | / |
| 4 | / | / |
| 1 | 1 | 1 |

(e) višedreveni grubozrnni /

| | | |
|---|---|---|
| 1 | 1 | |
| | | 2 |
| | | 3 |
| 4 | 4 | |
| | | 4 |
| 1 | 1 | |
| 1 | | |
| 2 | 2 | 2 |
| 2 | | |
| 3 | 3 | 3 |
| 4 | 4 | |
| 1 | 1 | 1 |
| 3 | 3 | 4 |
| 1 | 1 | |
| 1 | 1 | |

Z1R, 2011./2012.; slično

i
i love zano (BPM 0) s L71R, 2011./2012. (4. zad u talk)

②

mysub: add r6,r5,r6

loop: lw r1,0(r5)

nop

add r1,r1,r7

sw r1,0(r5)

addi r5,r5,4

bne r5,r6,loop

nop

jre r31

⑤. PM

- $a = 8$
 - 32-bitne adrese na računalu
 - $S = 16 \text{ kB}$
 - $\text{roznost} = 1 \text{ B}$
 - Širina linije = 32 B
 - 2 servisna bita (v, d)

(a) struktura adrese i ukupni broj bitova svake linije PU = ?

$$s = n \cdot b$$

$$n = \frac{16 \text{ kB}}{32 \text{ B}} = \frac{16 \cdot 1024 \text{ kB}}{2^3 \cdot 32 \text{ B}} = \boxed{512}$$

$$w(i) = \log_2(n/a) = \log_2\left(\frac{512}{8}\right) = \boxed{6}$$

$$w(p) = \log_2\left(\frac{32}{1}\right) = \boxed{5}$$

$$w(0) = 32 - 6 - 5 = \boxed{21}$$

$$\text{Ukupni - broj - bitova - slike - linije} = w(l) + 2(v, \sigma) + 32 \cdot 8 = \boxed{279}$$

(b) u koji skup linija se smješta adresa 0x21006543?

`0x21006543` | 0010 0001 0000 0000 0110 0101 0100 0011
 ↓
 42.

LJUB, 2011./2012.

- ③ Relativni adresni način za mem.
instrukciju STR:

① $ALU \leftarrow MDR[23:0], ALU \leftarrow PC$

$\rightarrow EMDR : C_5 = I_1 \cdot \phi$

:
.....

① $MAR \leftarrow PC + MDR[23:0]$

$\rightarrow EMAR : C_5 = I_1 \cdot (\phi_8 + \phi_9)$

$\rightarrow : C_{12} = I_1 \cdot (\phi_8 + \phi_9)$

$\rightarrow ADD : C_0 = I_1 \cdot (\phi_8 + \phi_9)$

$\rightarrow LALU : C_{15} = I_1 \cdot \phi_9$

$\rightarrow EALU : C_6 = I_1 \cdot (\phi_{10} + \phi_{11})$

$\rightarrow LMAR : C_{13} = I_1 \cdot \phi_{11}$

② $MDR \leftarrow A$

$\rightarrow EA : C_7 = I_1 \cdot (\phi_{12} + \phi_{13})$

$\rightarrow LMAR : C_{14} = I_1 \cdot \phi_{13}$

③ $M[MAR] \leftarrow MDR$

$\rightarrow WE, TE : C_6 = I_1 \cdot (\phi_{14} + \phi_{15})$

LJIR, 2011./2012.; slično 21, 2011./2012. (2. zadatak)
, slično 21, 2009./2010. (3. zadatak)

④ mysub: add r6,r5,r6

```
loop: lw r1,0(r5)      # r1 = *r5
      nop
      add r1,r1,r7      # r1 = r1 + r7
      sw r1,0(r5)        # *r5 = r1
      add r5,r5,4         # r5 ++
      bne r5,r6,loop     # while rs != r6
      nop
      jr r31
```

(a) Odgovarajuća funkcija u C-u:

```
int *s1=&start;
int *s2=&end;
while(s1 < s2){
    int t = *s1;
    t += a;
    *s1 = t;
    s1++;
}
```

| L1
(bolje)

```
void oddScalarToVector(int scalar, int *p, int size)
{
    int *end = p + size;
    while(p != end)
    {
        *p = *p + scalar;
        ++p;
    }
}
```

(b) Optimalan redoslijed instrukcija za procesor s jednostrukim izdvajanjem:

mysub: add r6,r5,r6

LJILJ, 2011./2012. ; slično zl. 2011./2012. (c.zad.)
; slično zl. 2016./2017. (s.zad.)

(6.)

JIR, 2011./2012.

③ AMDAHLOV ZAKON

s = faktor ubrzanja promatranih segmenta obrade

x = udio promatranih dijela u cijelom postupku

p = ubrzanje

$$p = \frac{1}{(1-x) + \frac{x}{s}}$$

(a)

$$4 = \frac{1}{(1-x) + \frac{x}{s}} \Rightarrow 4 = \frac{1}{(1-x) + \frac{x}{16}} \Rightarrow 4 - 4x + \frac{x}{4} = 1 \Rightarrow 4x - \frac{x}{4} = 3 \\ x\left(4 - \frac{1}{4}\right) = 3$$

$$\frac{15}{4}x = 3 / \cdot \frac{4}{15}$$

$$x = \frac{4}{15} = \boxed{0.8}$$

(b)

$$p = \frac{1}{(1-x) + \frac{x}{s}}$$

maksimalno ubrzanje... $x = 1$

$$8 = \frac{1}{(1-x) + \frac{x}{16}} \Rightarrow 8 - 8x + \frac{x}{2} = 1 \Rightarrow 8x - \frac{x}{2} = 7 \Rightarrow x\left(8 - \frac{1}{2}\right) = 7 \Rightarrow \frac{15}{2}x = 7 / \cdot \frac{2}{15} \\ x = \frac{14}{15} = \boxed{0.933}$$

(c) $p=2$

$$2 = \frac{1}{(1-x) + \frac{x}{728}} \Rightarrow 2 - 2x + \frac{1}{64}x = 1 \Rightarrow 2x - \frac{1}{64}x = 1 \Rightarrow x\left(2 - \frac{1}{64}\right) = 1 \\ \frac{127}{64}x = 1 / \cdot \frac{64}{127} \\ x = \frac{64}{127} = \boxed{0,50393}$$

~~JIR~~, 2011./2012. i slično 31.2021./2022. (2. zadatak)

DIR, 2011./2012.

① suba x ($A \leftarrow M[x] - A$)

faza izvrši:

① MAR $\leftarrow MDR[23:0]$

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

$$\rightarrow LMAR: C_{13} = I_g \cdot \phi_9$$

② MDR $\rightarrow M[MAR]$

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

$$\rightarrow LMAR: C_{14} = I_g \cdot \phi_{11}$$

③ $A \leftarrow MDR - A$

$$\rightarrow EMAR: C_5 = I_g \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow COMP: C_{18} = I_g \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow ADD: C_0 = I_g \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow LALU: C_{15} = I_g \cdot \phi_{13}$$

$$\rightarrow EALU: C_6 = I_g \cdot (\phi_{14} + \phi_{15})$$

$$\rightarrow LA: C_{10} = I_g \cdot \phi_{15}$$

D12, 2011./2012.

(4.)

MI, 2012./2013.

① RAM \rightarrow 8192 B

ROM \rightarrow 8192 B

U/I \rightarrow 4 B

- moduli imaju po dva priključka E
- 16-bitna adresna sabirница
- 8-bitna podatkovna sabirница
- RAM (od adresac o) \rightarrow U/I \rightarrow ROM

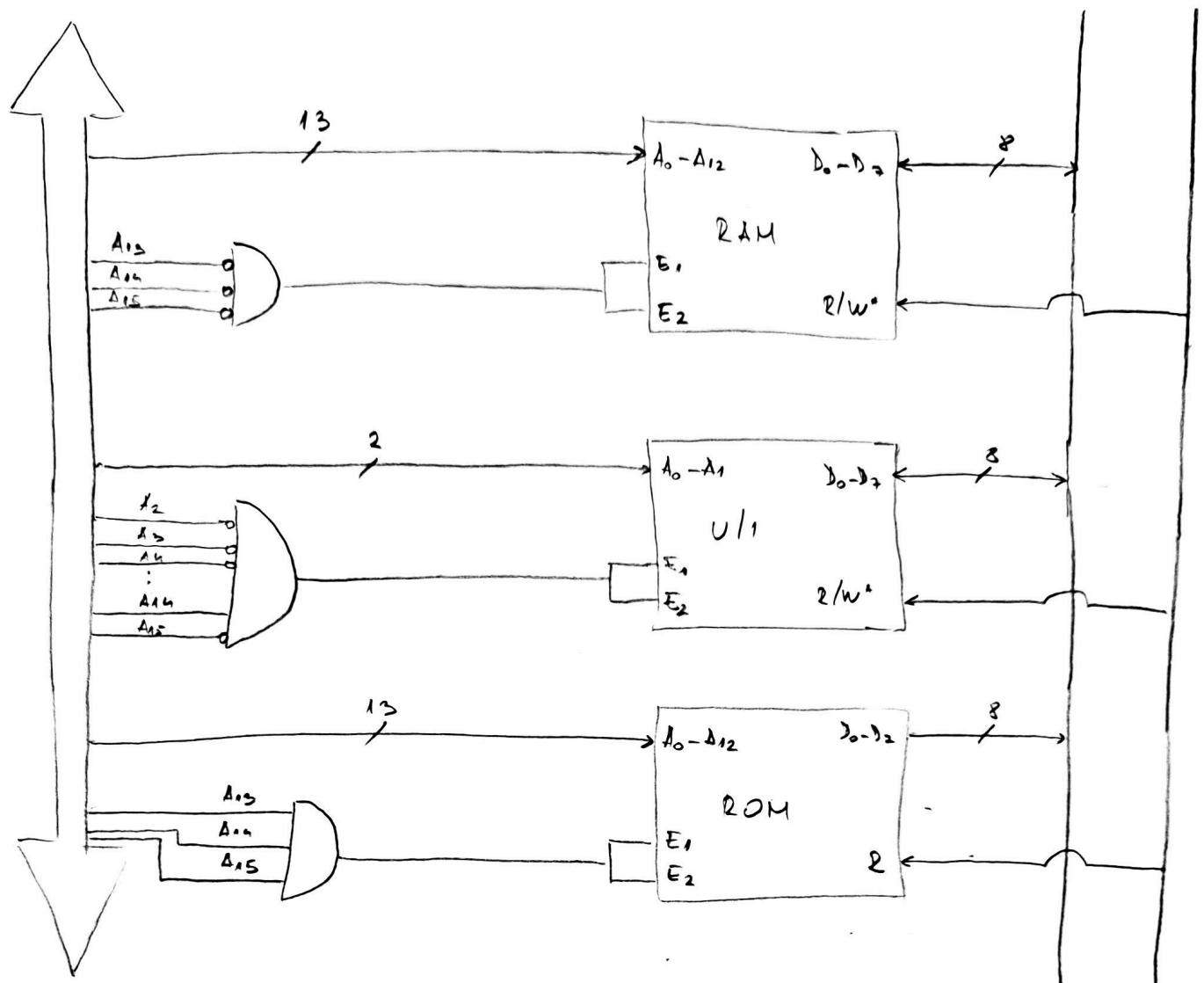
$$\text{RAM} \quad 8192 = 2^{13}$$

RAM od 0000 0000 0000 0000 \rightarrow { početnih 8192 B
do 0001 1111 1111 1111 \rightarrow 8191

U/I od 0100 0000 0000 0000 . { bilo gdje u sredini.
do 0100 0000 0000 0011

ROM od 1110 0000 0000 0000 { zadnjih 8192 B
do 1111 1111 1111 1111

1. nastavale



adresna
sabirnica

produktna Sabirnica upravljačka Sabirnica

DIR, 2012./2013.

① RAM \rightarrow 32 K, B \rightarrow 2¹⁵ B

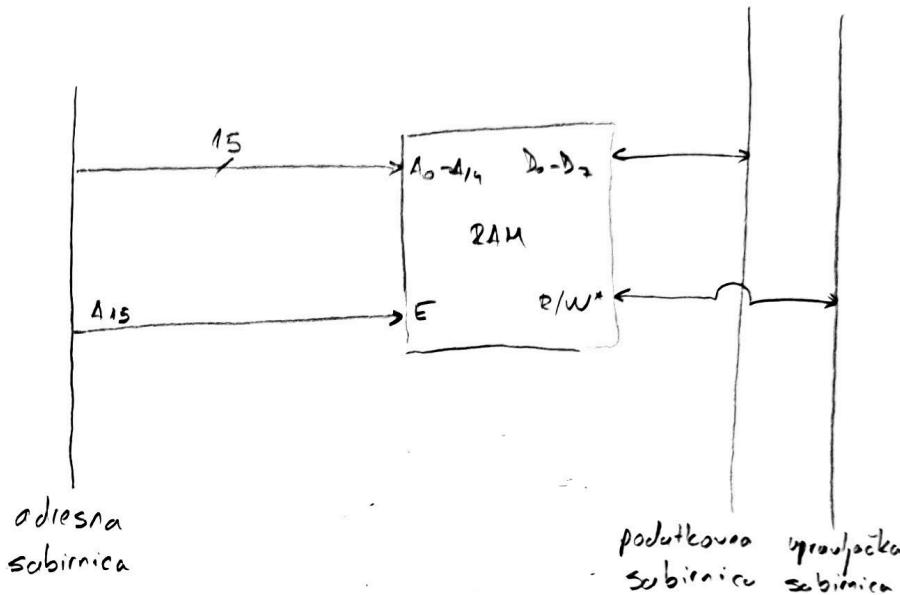
- jedan priključak E

- 16-bitna adresna, 8-bitovna podatkovna
subirnica

(a) modul je u kont. adres. prostoru na najvišim mogućim adresama

od 1000 0000 0000 0000

do 1111 1111 1111 1111



(b) pola na \$0000-\$3FFF,

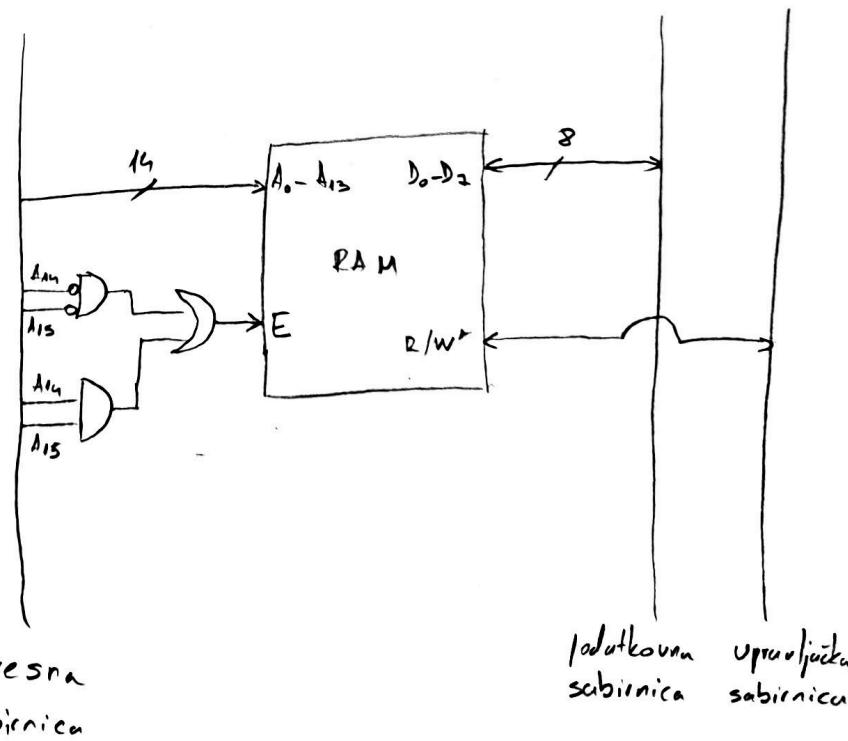
a pola na \$C000-\$FFFF

(1) od 0000 0000 0000 0000

do 0011 1111 1111 1111

(2) od 1100 0000 0000 0000

do 1111 1111 1111 1111



DIR, 2012./2013.

- ② add (zbrojanje dva registrarska operanda)
add (zbrojanje registrarskog operanda i usputne konstante)
32-bitni procesor RISC arhitekture

(a) formati navedenih instrukcija

(B.1) Aritmetičko-logičke instrukcije s registrarskim operandima

npr. add \$s1, \$s2, \$s3

R-tip format:

| 6bita 5bita 5bita 5bita 6bita | | | | | |
|-------------------------------|----|----|----|-------|-------|
| op. | rs | rt | rd | shamt | funct |
| | | | | | |

- polja rs i rt određuju izvorišne registre (operande), a polje rd odredišni register (rezultat)
- polje funct određuje funkciju ALU (op. kod je isti za sve A.-L. instrukcije)
- polje shamt određuje iznos pomaka

(B.2) Aritmetičko-logičke instrukcije s neposrednim operandom

npr. addi \$s1, \$2, 1

I-tip format:

| 6bita 5bita 5bita 16bita | | | |
|--------------------------|----|----|-------|
| op | rs | rt | konst |
| | | | |

(b) ADD R7, R7, R7

$$R7 = \$1FFF$$

$$\begin{array}{r} 0001\ 1111\ 1111\ 1111\ 1111 \\ +\ 0001\ 1111\ 1111\ 1111\ 1111 \\ \hline 0011\ 1111\ 1111\ 1111\ 1110 \end{array}$$

$$R7 = \$3FFE$$

ADDI R7, R7, \$1FFF

* konstantu se predzvono proširuje

$$\begin{array}{r} 0001\ 1111\ 1111\ 1111\ 1111 \\ +\ 1111\ 1111\ 1111\ 1111\ 1111 \\ \hline 1...0\ 001\ 1111\ 1111\ 1111\ 1110 \end{array}$$

DIR, 2012./2013., JIR, 2012./2013. (4. zadatak)

- ③ shr x (posmaki u desno sadržaj memorijске lokacije s adresom x)
• napisati logičke jednadžbe upravljačkih signala za tazu izvrši ove instrukcije

① propusti x u MAR ($\text{MAR} \leftarrow \text{MDR}[23:0]$)

$$\text{EMDR: } C_5 = I_g (\phi_8 + \phi_9)$$

$$\text{LMAR: } C_{13} = I_g \cdot \phi_9$$

② dohvati $\text{MEM}[x]$ ($\text{MDR} \leftarrow \text{M}[\text{MAR}]$)

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

$$\text{LMDR: } C_{14} = I_g \cdot \phi_{11}$$

③ u A stavi shr MDR ($A \leftarrow \text{MDR}$)

$$\text{EMDR: } C_5 = I_g \cdot (\phi_{12} + \phi_{13})$$

$$\text{LA: } C_{10} = I_g \cdot \phi_{13}$$

~~EX:~~

④ $(A \leftarrow \text{shr}(A))$

$$\text{SHR: } C_{12} = I_g \cdot \phi_{14}$$

⑤ $(\text{MDR} \leftarrow A)$

$$\text{EA: } C_7 = I_g \cdot (\phi_{15} + \phi_{16})$$

$$\text{LMDR: } C_{14} = I_g \cdot \phi_{16}$$

⑥ $(\text{M}[\text{MAR}] \leftarrow \text{MDR})$

$$\text{WRITE: } C_4 = I_g \cdot (\phi_{17} + \phi_{18})$$

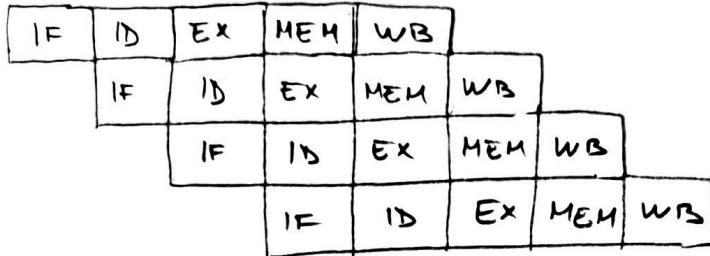
DIR, 2012./2013. | DIR, 2012./2013. (3. zadatuk)

④ procesor arhitekture MIPS

• 5-segmentni put podataka

| | | | | |
|----------------------|--|--|----------|----------|
| lw \$r1, 2(\$r6) | | | WAR \$r6 | |
| add \$r6, \$r2, \$r2 | | | RAW \$r6 | RAW \$r1 |
| sw \$r6, 2(\$r1) | | | | |
| lw \$r5, -8(\$r5) | | | RAW \$r5 | |
| sw \$r5, -8(\$r5) | | | WAR \$r5 | RAW \$r5 |
| add \$r5, \$r5, \$r5 | | | | |

(*)



(a) bez prosjedivanja

lw \$r1, 2(\$r6)

| IF | ID | EX | MEM | WB |
|----|----|----|-----|----|
| IF | ID | EX | MEM | WB |
| IF | ID | EX | MEM | WB |
| IF | ID | EX | MEM | WB |

add \$r6, \$r2, \$r2

~~sw \$r6, 2(\$r1)~~

nop

nop

nop

sw \$r6, 2(\$r1)

lw \$r5, -8(\$r5)

nop

nop

nop

sw \$r5, -8(\$r5)

add \$r5, \$r5, \$r5

| IF | ID | EX | MEM | WB |
|----|----|----|-----|----|
| | | | | |
| | | | | |
| | | | | |

| IF | ID | EX | MEM | WB |
|----|----|----|-----|----|
| IF | ID | EX | MEM | WB |
| | | | | |
| | | | | |

DIR, 2012./2013.

4.- nastavak

(b) proslijeđivanjem

lw \$r1, 2(\$r6)

add \$r6, \$r2, \$r2

nop

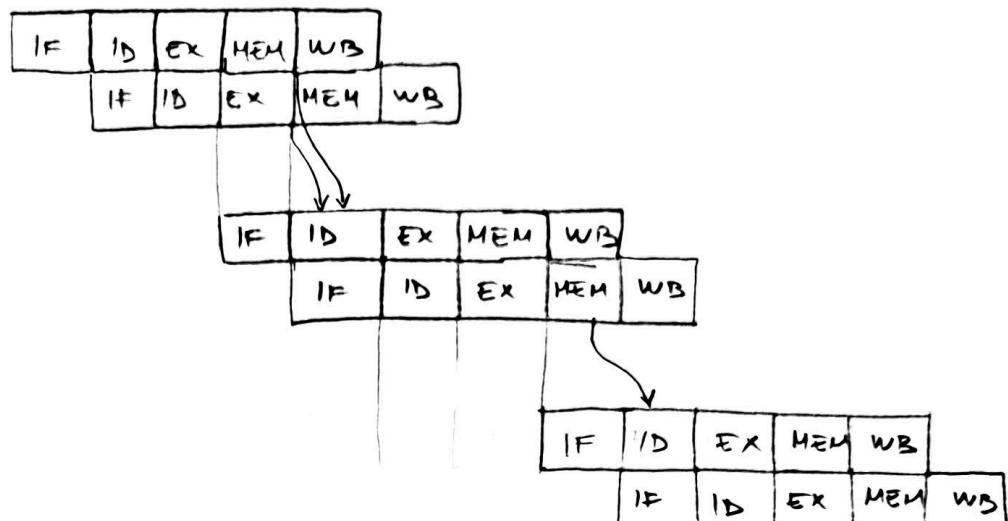
sw \$r6, 2(\$r1)

lw \$r5, -8(\$r5)

nop, nop

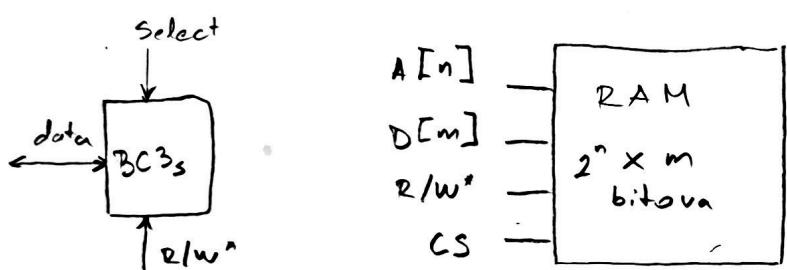
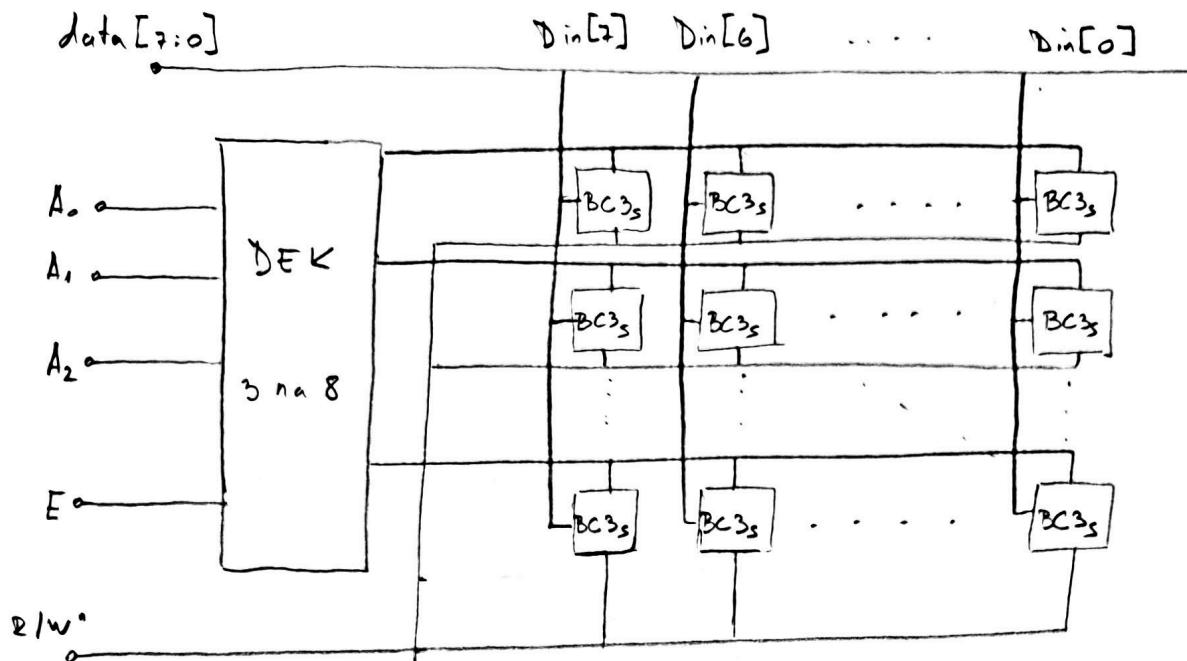
sw \$r5, -8(\$r5)

add \$r5, \$r5, \$r5



MI, 2013./2014.

- ① (a) izvedba memoriskog modula kapaciteta 8 8-bitnih rijeci pomoću binarnih ćelija s 3 stanja ($BC3S$)



①

2012.11.20 13./2014.

1. nustavak

16-bitna adresna

8-bitna podatkovna

- 3 modula koja se javljaju na kontinuiranim adresama počevši od adrese \$7F80

$$\$7F80 \equiv 0111\ 1111\ 1000\ 0000$$

prvi modul je od-do:

$$\$7F80 \equiv 0111\ 1111\ 1000\ 0000$$

$$\$7F87 \equiv 0111\ 1111\ 1000\ 0111$$

drugi modul je od-do:

$$\$7F88 \equiv 0111\ 1111\ 1000\ 1000$$

$$\$7F8F \equiv 0111\ 1111\ 1000\ 1111$$

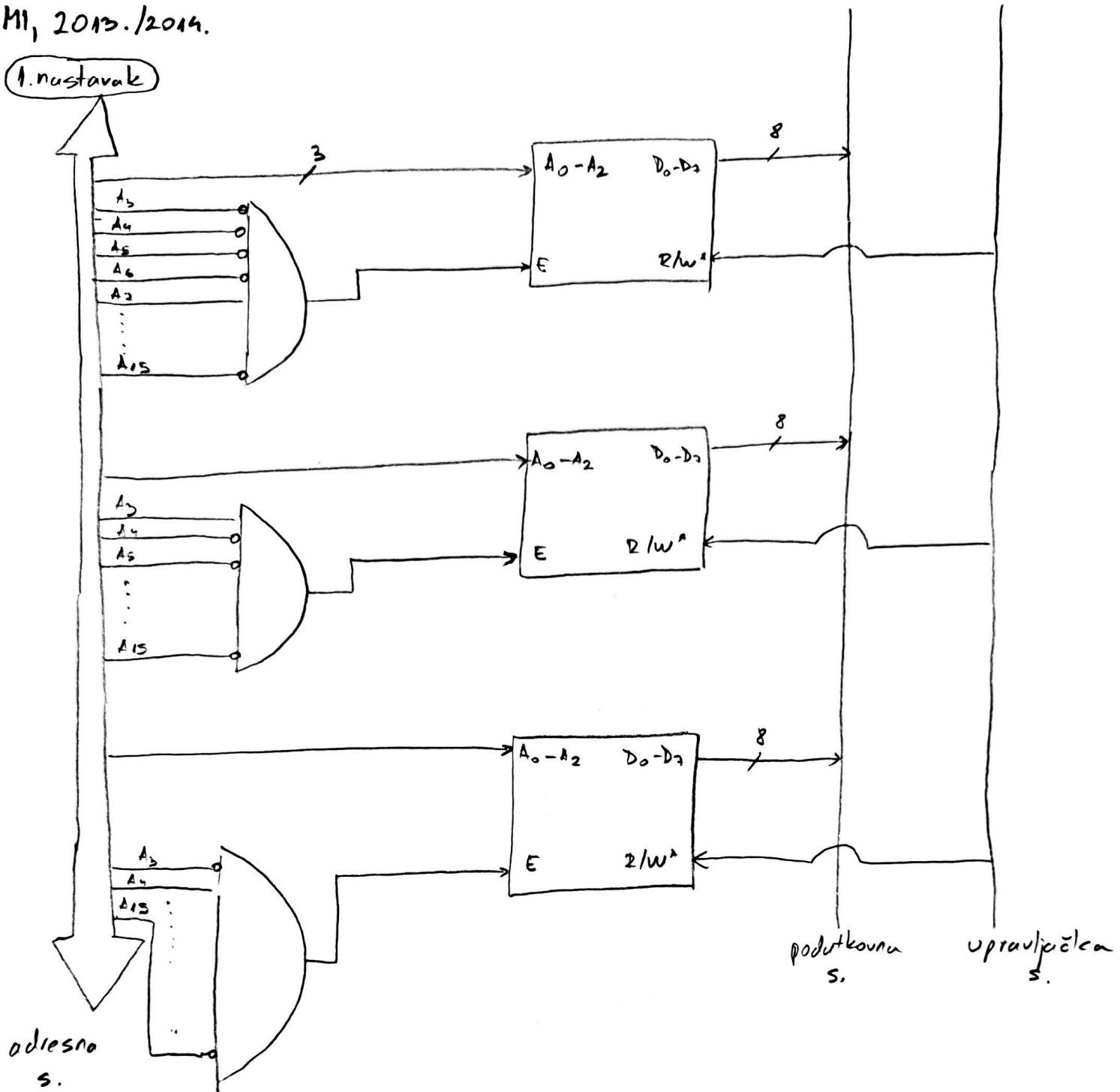
treći modul je od-do:

$$\$7F90 \equiv 0111\ 1111\ 1001\ 0000$$

$$\$7F97 \equiv 0111\ 1111\ 1001\ 0111$$

M1, 2013./2014.

1. nustavak



MI, 2013./2014.

②

$A[15:0]$, $\delta[7:0]$, R, W

...

$(\$1000, \$7C, 1, 0)$

$(\$1001, \$20, 1, 0)$

$(\$1002, \$00, 1, 0)$

$(\$2000, \$7C, 1, 0)$

$(\$1003, \$23, 1, 0)$

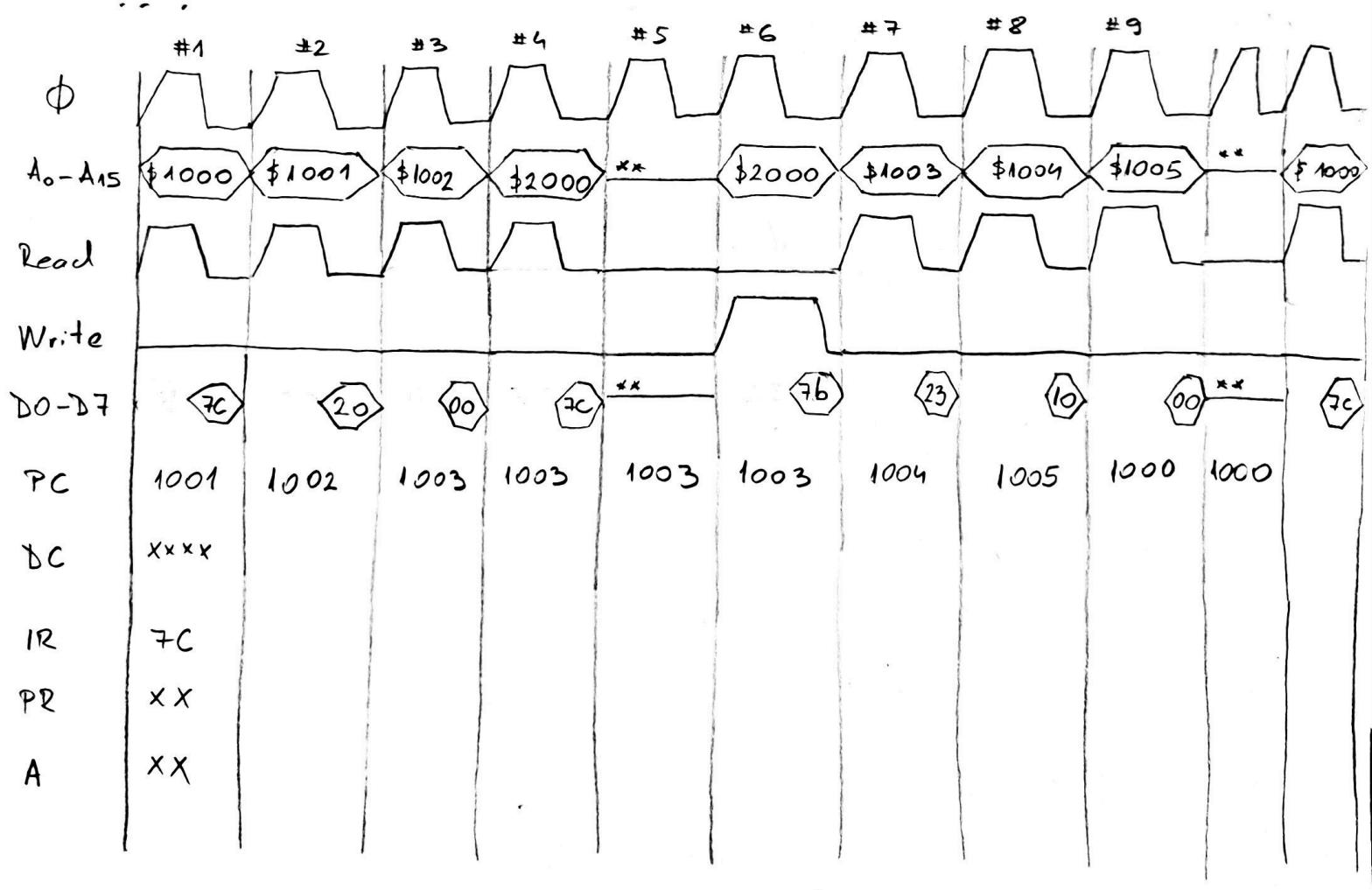
$(\$1004, \$10, 1, 0)$

$(\$1005, \$00, 1, 0)$

$(\$2, \$2, 0, 0)$

$(\$1000, \$7C, 1, 0)$

...



①

← faza pričekaji (fetch) →

← faza izvrši (execute) →

← faza pričekaji (fetch) →

← faza izvrši →

MI, 2013./2014.

2. - nastavak

program (odgovarajući) u asemblerskoj sintaksi:

DEC \$2000

JUMP \$1000

Početni sadržaj memorije izgleda sljedeće:

\$1000 7C

\$1001 20

\$1002 00

\$1003 23

\$1004 10

\$1005 00

:

\$2000 7C

MI, 2013./2014.

②

(-\$1000, \$7C, 1, 0)

(-\$1001, \$20, 1, 0)

(-\$1002, \$00, 1, 0)

(-\$2000, \$7C, 1, 0)

(z, z, 0, 0)

(-\$2000, \$7b, 0, 1)

(-\$1003, \$23, 1, 0)

(-\$1004, \$10, 1, 0)

(-\$1005, \$00, 1, 0)

(z, z, 0, 0)

(-\$1000, \$7C, 1, 0)

....

Memorijs:

\$1000 7C
\$1001 20 } DEC \$2000
\$1002 00 }

\$1003 23
\$1004 10 } JHP \$1000
\$1005 00 }

⋮
\$2000 7C

①

- ③ BRL X (Branch and Link) - instrukcija koja poziva potprogram na adresi x
spremajući pri tome povratnu adresu u porezni register L
RET (Return) koja ostvaruje povratak iz potprograma

faza izvrši instrukcije BRL X:

① $L \leftarrow PC$

$$\rightarrow EPC: C_8 = I_g \cdot (\phi_8 + d_8)$$

$$\rightarrow LL: C_{1g} = I_g \cdot \phi_g$$

② $PC \leftarrow MDR[23:0]$

$$\rightarrow EMDR: C_5 = I_g \cdot (\phi_{10} + d_{10})$$

$$\rightarrow LPC: C_{11} = I_g \cdot \phi_{11}$$

faza izvrši za instrukciju RET:

① $PC \leftarrow L$

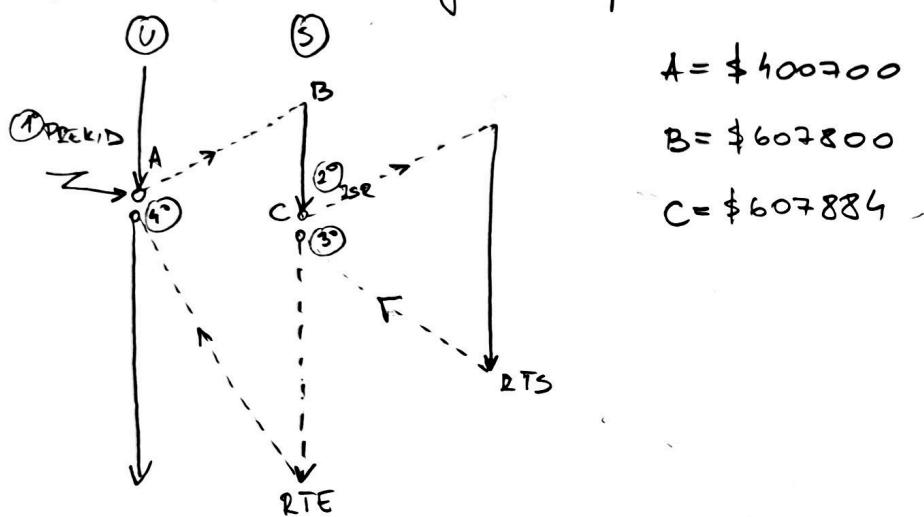
$$\rightarrow EL: C_{18} = I_{10} \cdot (\phi_e + d_g)$$

$$\rightarrow LPC: C_{11} = I_{10} \cdot d_g$$

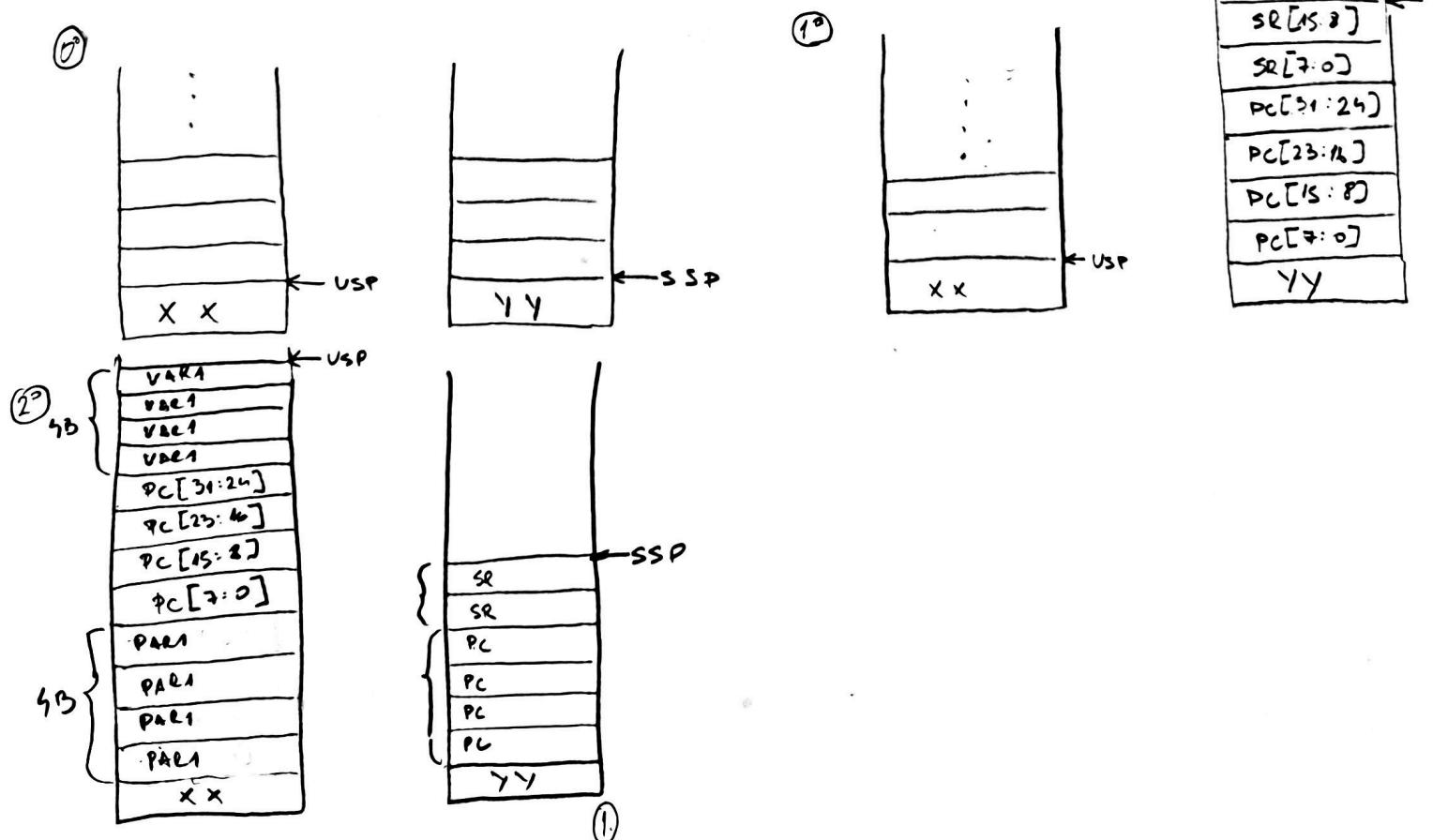
M1, 2013./2014.

- ④ • MC68000
- korisnički način rada
 - adresa \$400700; instrukcija (u bajta); dogoda se prekid
 - prekidni potprogram; adresa \$607800
 - poziva drugi potprogram instrukcijom CALL \$400900 (ona se vrati na adresu \$607884 i zauzima 6 bajtova)

(a) grafički prikaz opisanog scenarija



(b) prikaz stanja stoga u karakterističnim točkama (1, 2^o, 3^o, 4^o)



21.2013./2014.

① lista 32-bitnih memorijskih referenci biločih podataka:

| | (a) | (b) | (c) | |
|------|----------------|-----|-----|----|
| 0x01 | 0000 0001 | M | M | M |
| 0x86 | 1000 0110 | M | M | M |
| 0xd4 | 1101 0100 | M | M | M |
| 0x01 | 0000 0001 | H* | H* | H* |
| 0x87 | 1000 0111 | M | H* | M |
| 0xd5 | 1101 0101 | M | H* | M |
| 0xa2 | 1010 0010 | M | M | M |
| 0xa1 | 1010 0001 | M | M | H* |
| 0x02 | 0000 0010 | M | M | M |
| 0x2C | 0010 1100 | M | M | M |
| 0x29 | 0010 1001 | M | M | M |
| 0xdd | 1101 1101 | M | M | M |

(a) 8 linija, veličine 1 B

$$n = 8$$

$$b = 1 \text{ B}$$

$$S = n \cdot b = 8 \text{ B}$$

$$w(p) = \log_2 2^0 = 0$$

$$w(i) = \log_2 2^3 = 3$$

$$w(o) = 32 - 3 - 0 = 29$$

* izravno preslikavanje

\Rightarrow punjenje linije = 25 T

\Rightarrow vremena pristupa:

$$t_{C_1} = 2T$$

$$t_{C_2} = 3T$$

$$t_{C_3} = 5T$$

Odrediti najbolju i najgoru organizaciju

(b) 4 linije, veličine 2 B

$$n = 4$$

$$b = 2 \text{ B}$$

$$S = 8 \text{ B}$$

$$w(p) = \log_2 2^1 = 1$$

$$w(i) = \log_2 2^2 = 2$$

$$w(o) = 32 - 2 - 1 = 29$$

$$t_1 = 12 \cdot 2T + 11 \cdot 25T$$

$$= 299 \cdot T$$

$$t_2 = 12 \cdot 3T + 9 \cdot 25T = 261 \cdot T$$

$$t_3 = 12 \cdot 5T + 10 \cdot 25T = 310 \cdot T$$

(c) 2 linije, veličine 4 B

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

$$w(i) = \log_2 2^1 = 1$$

$$w(o) = 32 - 2 - 1 = 29$$

21.2013./2014.

③)

| | f | CPI_A | CPI_B | CPI_C | CPI_D |
|----|---------|---------|---------|---------|---------|
| P1 | 1,5 GHz | 1 | 2 | 4 | 4 |
| P2 | 2 GHz | 2 | 2 | 2 | 3 |
| P3 | 2,5 GHz | 3 | 1 | 4 | 1 |

• program P je zadatak s 10^6 instrukcija distribuiranih na sljedeći način:
 A(10%), B(20%), C(50%) i D(20%)

• koja implementacija će brže izvesti zadani program?

• odrediti ukupan broj taklova, kao i globalni CPI u oba slučaja.

$$P1: CPI_{x_1} = \frac{1}{10} \cdot 1 + \frac{2}{10} \cdot 2 + \frac{5}{10} \cdot 4 + \frac{2}{10} \cdot 4 = 3,3$$

$$P2: CPI_{x_2} = \frac{1}{10} \cdot 2 + \frac{2}{10} \cdot 2 + \frac{5}{10} \cdot 2 + \frac{2}{10} \cdot 3 = 2,2$$

$$P3: CPI_{x_3} = \frac{1}{10} \cdot 3 + \frac{2}{10} \cdot 1 + \frac{5}{10} \cdot 4 + \frac{2}{10} \cdot 1 = 2,7$$

$$P1: t_{CPU} = n_i \cdot CPI_{x_1} \cdot \frac{1}{f_1} = 10^6 \cdot 3,3 \cdot \frac{1}{1,5 \cdot 10^9} = 0,022 \text{ s}$$

$$P2: t_{CPU} = n_i \cdot CPI_{x_2} \cdot \frac{1}{f_2} = 10^6 \cdot 2,2 \cdot \frac{1}{2 \cdot 10^9} = 0,0011 \text{ s}$$

$$P3: t_{CPU} = n_i \cdot CPI_{x_3} \cdot \frac{1}{f_3} = 10^6 \cdot 2,7 \cdot \frac{1}{2,5 \cdot 10^9} = 0,0108 \text{ s}$$

Najbrža implementacija je P2; naisporija je P1.

Ukupni broj taklova:

$$P1: 0,022 \times 1,5 \cdot 10^9 = 3300000 \text{ taklova}$$

$$P2: 0,0011 \times 2 \cdot 10^9 = 2100000 \text{ taklova}$$

$$P3: 0,0108 \times 2,5 \cdot 10^9 = 2700000 \text{ taklova}$$

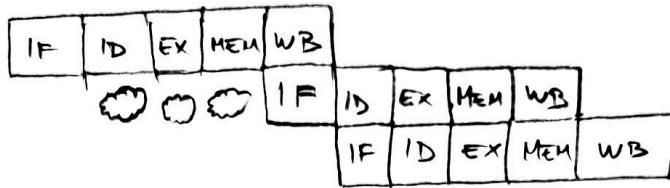
21.2013./2014.

4. MIPS arhitektura

(a) $lw \$t5, -16(\$t5)$ RAW $\$t5$
 $sw \$t5, -16(\$t5)$ RAW $\$t5$
 $add \$t5, \$t5, \$t5$ WAR $\$t5$

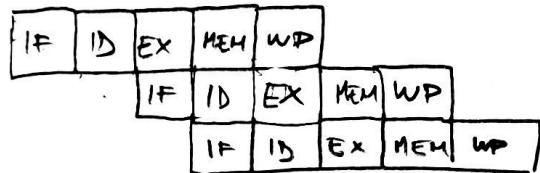
(b) bez prospektivnja

$lw \$t5, -16(\$t5)$
nop, nop, nop, $sw \$t5, -16(\$t5)$
 $add \$t5, \$t5, \$t5$



(c) s prospektivnjem

$lw \$t5, -16(\$t5)$,
nop, $sw \$t5, -16(\$t5)$
 $add \$t5, \$t5, \$t5$



21.8.2013./2014.

② ($A[15:0]$, $D[7:0]$, e, w)

...
(\$0100, \$B6, 1, 0)
(\$0101, \$10, 1, 0)
(\$0102, \$11, 1, 0)
(\$1011, \$CA, 1, 0)
(\$0103, \$87, 1, 0)
(\$0104, \$20, 1, 0)
(\$0105, \$00, 1, 0)
(\$9FFF, \$06, 0, 1)
(\$9FFE, \$01, 0, 1)
...

Zadano: LDA \$B6
SUB \$10
INC \$01
STA \$CA
CALL \$87
JMP \$20
NOP \$00
ADD \$06

\$0100: B6
\$0101: 10
\$0102: 11 } LDA \$1011
\$1013: 87
\$0104: 20
\$0105: 00 } CALL \$2000 \longleftrightarrow^* CALL ~~1000~~ zahvjeta slog =>
nu poziciju na koju pokazuje SP
~~SP~~ koji je očito A000 stavljamo
\$1006 i to unatrag !!!

21.2013./2014.

⑥ pojednostavljeni model procesora

- little-endian način pohrane podataka
- prva instrukcija na adresi: \$1320

\$1320: LDA \$1324 //op. load 34
STA \$3142 //op. load A2

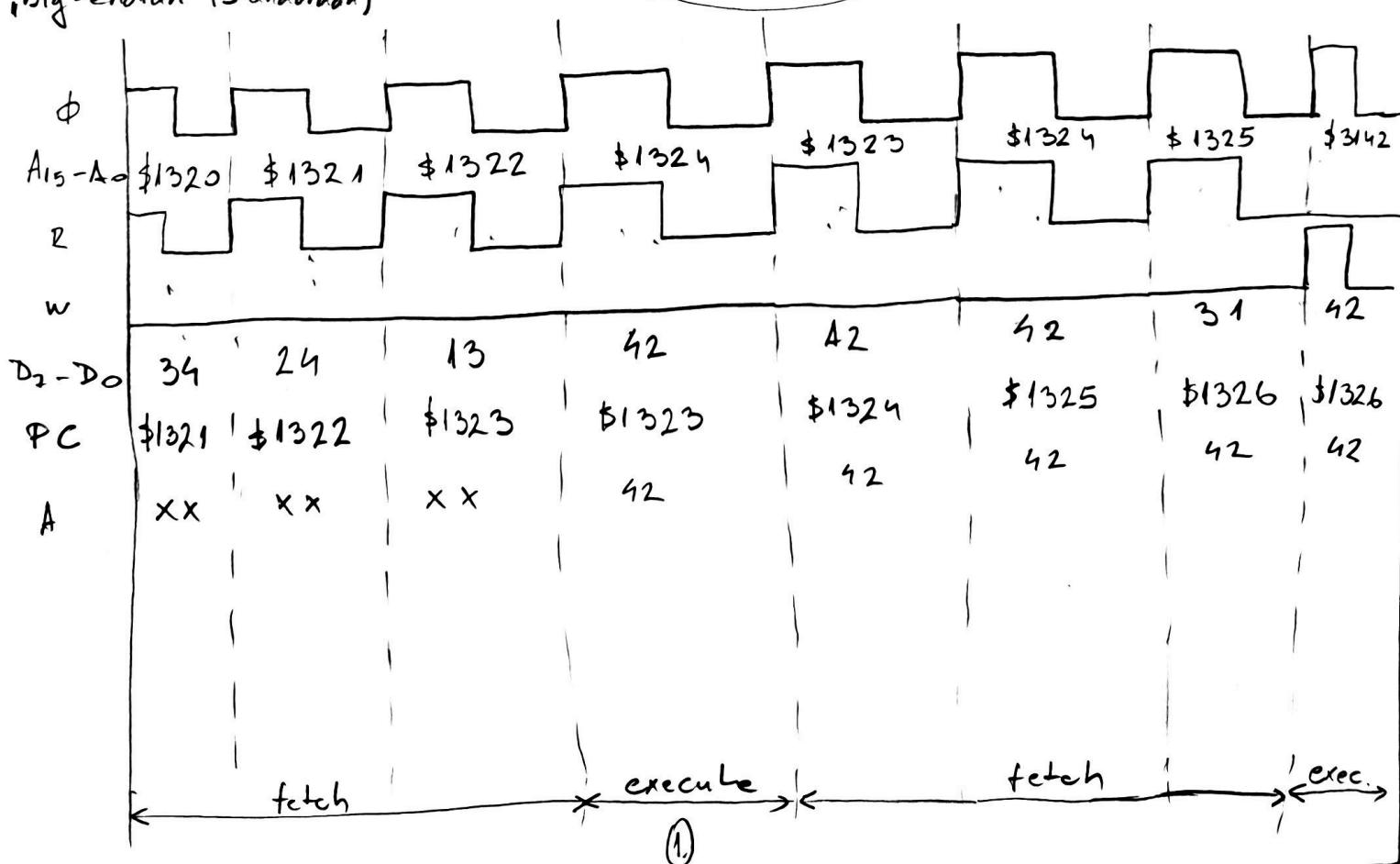
Sadržaj memorije:

| | |
|--------|----|
| \$1320 | 34 |
| \$1321 | 13 |
| \$1322 | 24 |
| \$1323 | A2 |
| \$1324 | 31 |
| \$1325 | 42 |
| : | |
| \$3142 | XX |

\$1320 34
\$1321 24
\$1322 13
\$1323 A2
\$1324 42
\$1325 31
:
\$3142 XX

"little-endian"

"big-endian" (standarden)



JIR, 2014./2015.

- ① • indeksirano adresiranje instrukcijom LDA IND
• LDAIND ADDR \rightarrow učitava podatak koji se nalazi na adresi $X + ADDR$, gdje je X sadržaj indeksnog registra

Najprije moramo dodati taj register X, spojiti ga na INTBUS te dodati signale $C_{18}(EX)$ i $C_{19}(LX)$. To tom treba kako bi podatak x mogli postaviti na Q od ALU sklopa. Radi dodavanja novih instrukcija dekoder mijenjamo u 4/16, 12 IR uzimamo bit više te u generatoru sljedova dolazi linija više.

① $A \leftarrow MDR[23:0]$
 $\rightarrow EMDR : C_5 = \bar{I}_g \cdot (\phi_8 + \phi_9)$
 $\rightarrow LA : C_{10} = I_g \cdot \phi_9$

② $MAR \leftarrow X + A$
 $\rightarrow EX : C_{18} = I_g \cdot (\phi_{10} + \phi_{11})$
 $\rightarrow ADD : C_0 = I_g \cdot (\phi_{10} + \phi_{11})$
 $\rightarrow LALU : C_{15} = I_g \cdot \phi_{11}$
 $\rightarrow EALU : C_6 = I_g \cdot (\phi_{12} + \phi_{13})$
 $\rightarrow LMAR : C_{13} = \bar{I}_g \cdot \phi_{13}$

③ $MDR \leftarrow MEM[MAR]$
 $\rightarrow READ : C_3 = I_g \cdot (\phi_{14} + \phi_{15})$
 $\rightarrow LMAR : C_{14} = I_g \cdot \phi_{15}$

④ $A \leftarrow MDR$
 $\rightarrow EMDR : C_5 = I_g \cdot (\phi_{16} + \phi_{17})$
 $\rightarrow LA : C_{10} = I_g \cdot \phi_{17}$

JIR, 2014./2015.

⑦ 32-bitne adrese; izravno preslikavanje

-bitovi 31-14: oznaka

-bitovi 13-4: indeks

-bitovi 3-0: pomak

$$(a) w(p) = \boxed{4} = \log_2 b \Rightarrow b = 2^4 = 16 \text{ B}$$

$$w(i) = \boxed{10} = \log_2 n \Rightarrow n = 2^{10} = 1024 \text{ linija}$$

$$w(o) = \boxed{8}$$

$$\underline{\underline{s}} = n \cdot b = 16 \text{ kB}$$

| | |
|-----|------|
| (a) | 1 |
| | 2 |
| | 4 |
| | 8 |
| | 16 |
| | 32 |
| | 64 |
| | 128 |
| | 256 |
| | 512 |
| | 1024 |
| | 2048 |
| | 4096 |

(b)

| | | |
|------|-------------------------------|-----|
| 0 | 0000 0000 0000 0000 0000 0000 | M |
| 4 | 0000 0000 0100 | H ← |
| 16 | 0000 0001 0000 | M |
| 132 | 0000 1000 0100 | M |
| 232 | 0000 1110 1000 | M |
| 160 | 0000 1010 0000 | M |
| 1024 | 0100 0000 0000 | M |
| 30 | 0000 0001 1110 | H ← |
| 140 | 0000 1000 1100 | H ← |
| 3100 | 1100 0001 1100 | M |
| 180 | 0000 1011 0100 | M |
| 2180 | 1000 1000 0100 | H |

Omjer pogodaka je $3/12 = 0,25 = 25\%$.

ZIR, 2015./2016.

$$\textcircled{3} \quad a_1 = 2$$

$$S_1 = 512 \text{ B}$$

$$b_1 = 16 \text{ B}$$

LRU.

$$a_2 = 8$$

$$S_2 = 1 \text{ MB}$$

$$b_2 = 16 \text{ B}$$

LRU₂

0x0001, 0x0086, 0x00d4, 0x0001, 0x0187, 0x00d5, 0x01d2, 0x0281, 0x0002,
0x008c, 0x0189, 0x00dd

$$t_{PH} = 1 \text{ T}$$

$$T_{GH} = 100 \text{ T}$$

| | | L1 | L2 |
|--------|---------------------|----|----|
| 0x0001 | 0000 0000 0000 0001 | H | H |
| 0x0086 | 0000 0000 1000 0110 | H | H |
| 0x00d4 | 0000 0000 1101 0100 | H | H |
| 0x0001 | 0000 0000 0000 0001 | H | H |
| 0x0187 | 0000 0001 1000 0111 | H | H |
| 0x00d5 | 0000 0000 1101 0101 | H | H |
| 0x01d2 | 0000 0001 1101 0010 | H | H |
| 0x0281 | 0000 0010 1000 0001 | H* | H |
| 0x0002 | 0000 0000 0000 0010 | H | H |
| 0x008c | 0000 0000 1000 1100 | H* | H |
| 0x0189 | 0000 0001 1000 1001 | H* | H |
| 0x00dd | 0000 0000 1101 1101 | H | H |

Racun (potreban):

$$\textcircled{1^o} \quad S_1 = n_1 \cdot b_1$$

$$n_1 = \frac{S_1}{b_1} = \frac{512 \cancel{\text{B}}}{16 \cancel{\text{B}}} = 32$$

[32 linije PH]

$$w(i) = \log_2(\lceil \frac{n}{a} \rceil) = 4$$

$$w(o) = 32 - 4 - 4 = 24$$

$$\textcircled{3^o} \quad n_2 = \frac{S_2}{b_2} = \frac{1 \cdot 1024 \cdot 1024 \cancel{\text{B}}}{16 \cancel{\text{B}}} = 65536$$

$$= 65536$$

\textcircled{4^o} struktura adrese:

$$w(p) = \log_2 b = 4$$

$$w(i) = \log_2(\lceil \frac{n}{a} \rceil) = 13$$

$$w(o) = 32 - 13 - 4 = 15$$

$$\textcircled{5^o} \quad T_1 = 4 \cdot 1 \text{ T} + 8 \cdot 100 \text{ T} = 804 \text{ T}$$

$$\frac{800 \text{ T}}{12} = \boxed{66,67 \text{ T}}$$

$$T_2 = 4 \cdot 1 \text{ T} + 2 \cdot 10 \text{ T} + 6 \cdot 100 \text{ T}$$

$$= 624 \text{ T}$$

$$\frac{624 \text{ T}}{12} = 52 \text{ T}$$

\textcircled{2^o} struktura adrese:

$$w(p) = \log_2 b = 4$$

212, 2015./2016. i slično 21, 2021./2022. (2. zadatak)
i slično 212, 2015./2016. (3. zadatak)

JIR, 2015./2016.

② · pojednostavljeni model procesora

· početna adresa \$0020

· sljedeći niz bajtova: \$22, \$20, \$24, \$25, \$26, \$20

· poznato je sljedeće: INC - \$20

DEC - \$21

LDA - \$22

ADD - \$

STA - \$24

SUB - \$25

JMP - \$26

\$0020 : \$22 } INC \$2024

\$0021 : \$20 }

\$0022 : \$24 }

\$0023 : \$25 }

\$0024 : \$26 }

\$0025 : \$20 }

\$0026

:

\$2024 :

JIR, 2015./2016.

- ② \$ 0020 → \$22
\$ 0021 → \$20
\$ 0022 → \$24
\$ 0023 → \$25
\$ 0024 → \$26
\$ 0025 → \$20

LDA

①

JIR, 2015./2016.; slično dostu 21, 2021./2022. (2. zadatuk.)

③ $p = \$4$
 $p_2 = \$5$
 $z = \$6$

beg \$4, \$5, EXIT

LOOP: lw r0, 0(\$4) # $r0 = *p$

sw r0, 0(\$6) # $*z = *p$

addi \$6, \$6, #4

addi \$4, \$4, #4

bne \$4, \$5, LOOP

EXIT: ...

NEOPTIMALNI:

beg \$4, \$5, EXIT

nop

LOOP: lw r0, 0(\$4)

nop

nop

nop

sw r0, 0(\$6)

addi \$4, \$4, #4

addi \$6, \$6, #4

bne \$4, \$5, LOOP

EXIT: ...

OPTIMALNI:

beg \$4, \$5, EXIT

nop

LOOP: lw r0, 0(\$4)

addi \$6, \$6, #4

addi \$4, \$4, #4

bne \$4, \$5, LOOP

nop

nop

sw r0, -4(\$6)

EXIT:

?

718, 2015./2016.

④ L1

$$\overline{T}_1 = 2,4 \text{ T}$$

$$\rightarrow \text{pogodak} \Rightarrow 1 \text{ T}$$

$$\rightarrow \text{pomosaj} \Rightarrow 80 \text{ T}$$

$$L_2 \Rightarrow \text{pogodak} \Rightarrow 6 \text{ T}$$

JIR, 2015./2016.; slično JIR, 2014./2015. (5. zadatak)

i slično LJIR, 2011./2012. (6. zadatak)

i slično ZI, 2011./2012. (6. zadatak)

21.2016./2017.

(2)

| rozred | A | B | C | D |
|------------|-----|-----|-----|-----|
| CPI | 1 | 2 | 4 | 4 |
| Učest.losť | 0.1 | 0.2 | 0.4 | 0.3 |

(a) Vekupní CPI = ?

$$CPI_{vk} = 1 \cdot 0,1 + 2 \cdot 0,2 + 4 \cdot 0,4 + 4 \cdot 0,3 \\ = \boxed{3,3}$$

(b) potrebná prémiera parametra CPI_B ke kde bi se performanca ročnula poholšala za 20%?

$$p = 1,2, x = 0,2$$

* Amdahlov zákon *

$$\textcircled{P} = \frac{1}{(1-x) + \frac{x}{S}} \Rightarrow 1,2 = \frac{1}{(1-0,2) + \frac{0,2}{S}} \Rightarrow 1,2 \cdot 0,8 + \frac{0,2}{S} = 1 \\ \Rightarrow \frac{0,2}{S} = 0,04 \Rightarrow \boxed{S = 5}$$

$$CPI_B^2 = \frac{CPI_B}{S} = \boxed{0,333}$$

$$(C) P = \frac{1}{(1-x) + \frac{x}{S}}$$

$$P = ?, x = \frac{0,4}{2} = 0,2; S = 3$$

$$P = \frac{1}{(1-0,2) + \frac{0,2}{3}} = \frac{15}{13} = \boxed{1,15385}$$

21, 2016./2017.

- ③ (a) izravno preslikavanje, vrijeme pogotka je $2AT$
 (b) asocijativno preslikavanje u skupove od 2 linije; vrijeme lag. je $3AT$
 (c) $\overline{\text{---}} \text{II} \overline{\text{---}} \quad \overline{\text{---}} \text{I} \overline{\text{---}}$ 4 linije; $\overline{\text{---}} \text{II} \overline{\text{---}}$ 4 linije

- $b = 4B$
- $s = 32 B$
- LRU
- cijena promatršija = $10 AT$
- pozivi:

| | | (a) | (b) | |
|----|-------------------------|-----|-----|--|
| 22 | 0000.....0000 0001 0110 | M | M | |
| 26 | 0000....0000 0001 1010 | M | M | |
| 23 | 0000....0000 0001 0111 | H | H | |
| 27 | 0000....0000 0001 1011 | H | H | |
| 16 | 0000...0000 0001 0000 | M | M | |
| 3 | 0000...0000 0000 0011 | M | M | |
| 17 | 0000...0000 0001 0001 | H | H | |
| 2 | 0000...0000 0000 0010 | H | H | |
| 34 | 0000...0000 0010 0010 | M* | M | |
| 18 | 0000...0000 0001 0010 | H | M | |

128
64
32
16
8
4
2
1

struktura adrese:

$$(a) s = n \cdot b$$

$$n = \frac{s}{b} = 8$$

$$w(i) = \log_2 8 = 3$$

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

$$w(o) = 32 - 3 - 2 = 27$$

$$(b) w(i) = \log_2 (n/a) = 2$$

$$\textcircled{3} a = 2$$

$$w(p) = \log_2 4 = 2$$

$$w(o) = 32 - 2 - 2 = 28$$

ZI, 2016./2017. (5.zadatak) ; slično ZI, 2011./2012. (6.zadatak)
; slično ZI, 2021./2022. (5.zadatak)

JIR, 2020./2021.

(2.)

JIR, 2020./2021.

③.

312, 2020./2021. (5. zád.) ; slíčno 21, 2011./2012. (6. zád.)
; slíčno 312, 2014./2015. (5. zád.)

(5.)

JIR, 2020./2021. ; slično zl, 2020./2021. (2. zadatak)
i slično JIR, 2015./2016. (3. zadatak)

MI, 2021./2022.

①

(A)



①

MI, 2021./2022.

$$\textcircled{3.} \quad t_{CPU} = \eta_i \cdot CPI \cdot T_{clock} = \frac{\eta_i \cdot CPI}{f_{CPU}} = \frac{\eta_i}{f_{CPU}}$$

$$CPI_p = \left[\frac{N_{MEM}}{N_{NS}} \right] \cdot C_{MEM} + \left[\frac{N_{INT}}{N_{NS}} \right] \cdot C_{INT} + \left[\frac{N_{FP}}{N_{NS}} \right] \cdot C_{FP} + \left[\frac{N_{BR}}{N_{NS}} \right] \cdot C_{BR}$$

$$N_{NS} = 5 + 2 + N_{array} \cdot 11 = \boxed{\quad}$$

$$CPI_p \approx \frac{C_{MEM} \times 4 + C_{INT} \times 4 + C_{FP} \times 2 + C_{BR} \times 1}{11} \times \frac{N_{NS}}{N_{NS}}$$

$$CPI_{p_1} = \frac{5 \cdot 4 + 1 \cdot 4 + 3 \cdot 2 + 4 \cdot 1}{11} = \frac{34}{11}$$

$$CPI_{p_2} = \frac{4 \cdot 4 + 2 \cdot 4 + 2 \cdot 2 + 3 \cdot 1}{11} = \frac{31}{11}$$

\textcircled{1}

MI, 20.21./2022.

INC \$0020

LDA \$0020

ADD #\\$18

$24 \equiv 0010\ 0100$

$18 \equiv 0001\ 1000$

$$\begin{array}{r} 0011 \\ + 1100 \\ \hline 0011\ 1100 \end{array} \equiv 3C$$

- ④
 0020: \$23
 0021: \$20
 0022: \$18
 0023: \$20
 0024: \$9B
 0025: \$18

| | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 |
|------------------------------|-------|---------|------|-------|---------|------|-------|---------|------|------|------|-----|
| Φ | | | | | | | | | | | | |
| R | | | | | | | | | | | | |
| W | | | | ** | | | | | | | ** | |
| PC | 0021 | 0022 | 0022 | 0022 | 0023 | 0024 | 0024 | 0025 | 0025 | 0026 | 0026 | |
| DC | xxxx | 0020 | 0020 | 0020 | 0020 | 0020 | 0020 | 0020 | 0020 | 0020 | 0020 | |
| IR | 23 | 23 | 23 | 23 | 23 | 18 | 18 | 18 | 9B | 9B | 9B | |
| PR | xx | xx | 23 | 24 | 24 | 24 | 24 | 24 | 18 | 18 | 18 | |
| Acc | xx | xx | xx | xx | xx | xx | xx | 24 | 24 | 24 | 3C | |
| A ₁₅₋₁₀ | 0020 | 0021 | 0020 | 0020 | 0020 | 0022 | 0023 | 0020 | 0024 | 0025 | ** | |
| D _{2-D₀} | 23 | 20 | 23 | 23 | 24 | 18 | 20 | 24 | 9B | 18 | ** | |
| | fetch | execute | | fetch | execute | | fetch | execute | | | | |

21.2021./2022.; slično dosta JIR, 2015./2016. (3. zadatak)

② neoptimirano:

lw \$t_0, 0(\$t_4) // $t_0 = p$

lw \$t_1, 0(\$t_5) // $t_1 = p_2$

lw \$t_2, 0(\$t_6) // $t_2 = 2$

LOOP: lw \$t_3, 0(\$t_0) // $t_3 = *p$

nop

sw \$t_3, 0(\$t_2) // $*q = *t_3$

addi \$t_0, \$t_0, 4 // $+r_p$

addi \$t_2, \$t_2, 4 // $++q$

bne \$t_0, \$t_1, LOOP // while $p \neq p_2$

optimizirano:

lw \$t_0, 0(\$t_4)

lw \$t_1, 0(\$t_5)

lw \$t_2, 0(\$t_6)

LOOP: lw \$t_3, 0(\$t_0)

addi \$t_0, \$t_0, 4

sw \$t_3, 0(\$t_2)

addi \$t_2, \$t_2, 4

bne \$t_0, \$t_1, LOOP

zl. 2021/2022. ; slično zl. 2016./2017. (5. zad.)
; slično zl. 2011./2012. (6. zad.)

| 5) | D1 | D2 | D3 | D4 |
|----|----|----|----|----|
| | 11 | 22 | 31 | 41 |
| | 11 | 12 | 11 | 11 |
| | 11 | 12 | 13 | 14 |
| | 11 | 11 | 11 | 44 |
| | 11 | 22 | 33 | 11 |
| | 11 | 21 | 13 | 14 |

(a) SMT

(b) finažnata višedr.

(c) grubožernata višedretnost

| | |
|----|----|
| 11 | 11 |
| 22 | 22 |
| 31 | 12 |
| 41 | 12 |
| 11 | 31 |
| 12 | 41 |
| 13 | 11 |
| 14 | 11 |
| 11 | 11 |
| 12 | 22 |
| 33 | 21 |
| 44 | 13 |
| 11 | 14 |
| 22 | 44 |
| 13 | 33 |
| 14 | 13 |
| 21 | 14 |

VI, 2022./2023.

4. na varijsku sabirnicu CISC procesora spojen je ulazni broj koji ima 2 s-bitne registre: (i) register stanja i (ii) podatkovni register

-register stanja inicira se na adresi \$F000, a podatkovni register na adresi \$F001

LDA \$F001

1000: \$B6

BEQ \$03

1001: \$F0

STA \$4000

1002: \$01

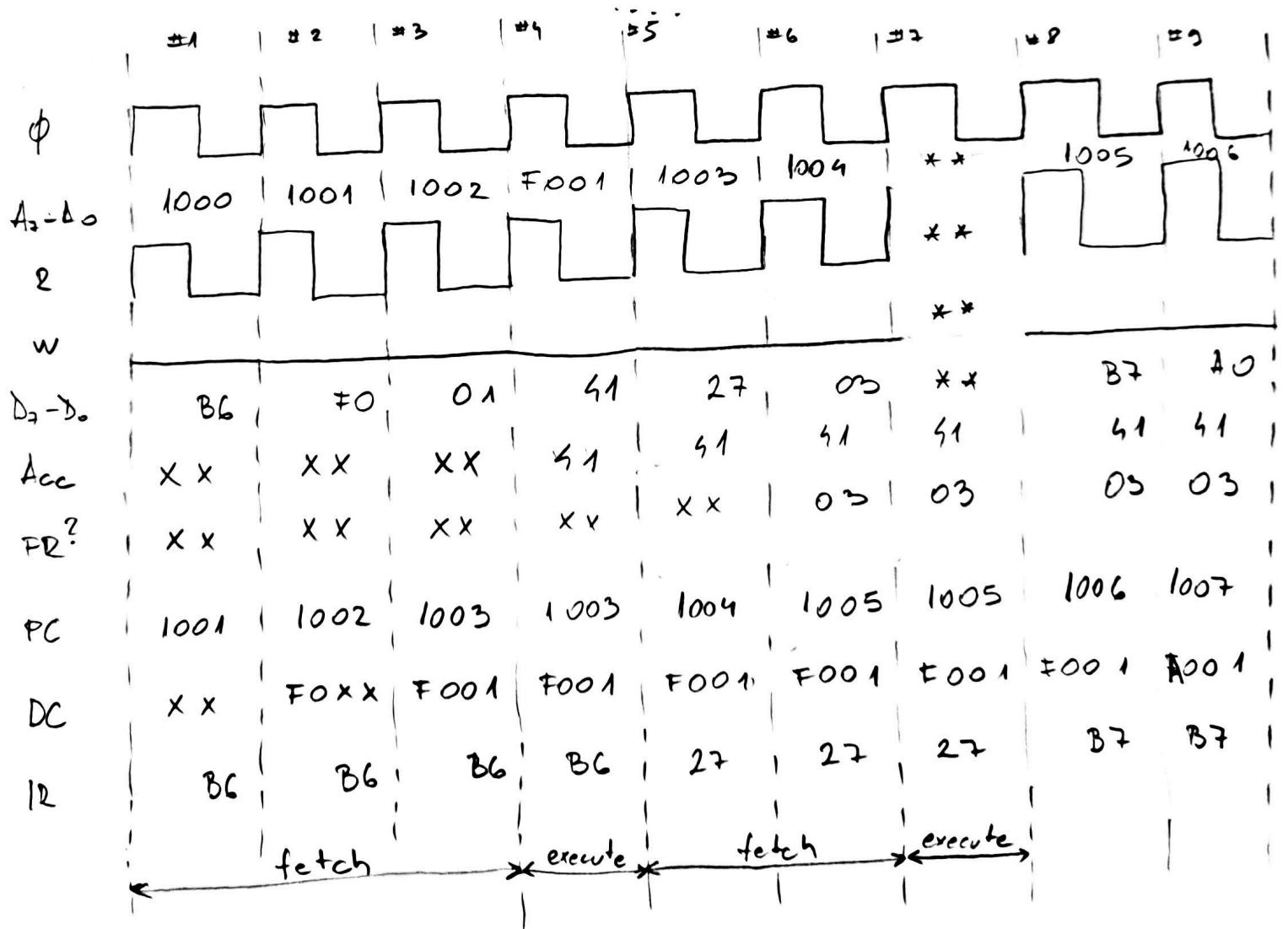
1003: \$27

1004: \$03

1005: \$B7

1006: \$A0

1007: \$00



MI, 2022./2023.

⑤ LDAR $x \rightarrow$ efektivna adresa memoriskog operanda se tvori kao $PC + x$

- (i) opisati promjene puta podatka
- (ii) opisati izmjene upravljačke jedinice

⑥ izvechi navedenu ~~kontrolu~~ instrukciju

① $A \leftarrow MDR[23:0]$

$$\rightarrow EMDR: C_5 = I_g \cdot (\phi_8 + d_9)$$

$$\rightarrow LA: C_{10} = I_g \cdot \phi_9$$

② $MAR \leftarrow A + PC$

$$\rightarrow EPIC: C_8 = I_g \cdot (\phi_{10} + \phi_{11})$$

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

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

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

$$\rightarrow LMAR: C_{13} = I_g \cdot \phi_{13}$$

③ $MDR \leftarrow M[44:e]$

$$\rightarrow READ: C_3 = I_g \cdot (\phi_{14} + \phi_{15})$$

$$\rightarrow LMDR: C_{14} = I_g \cdot \phi_{15}$$

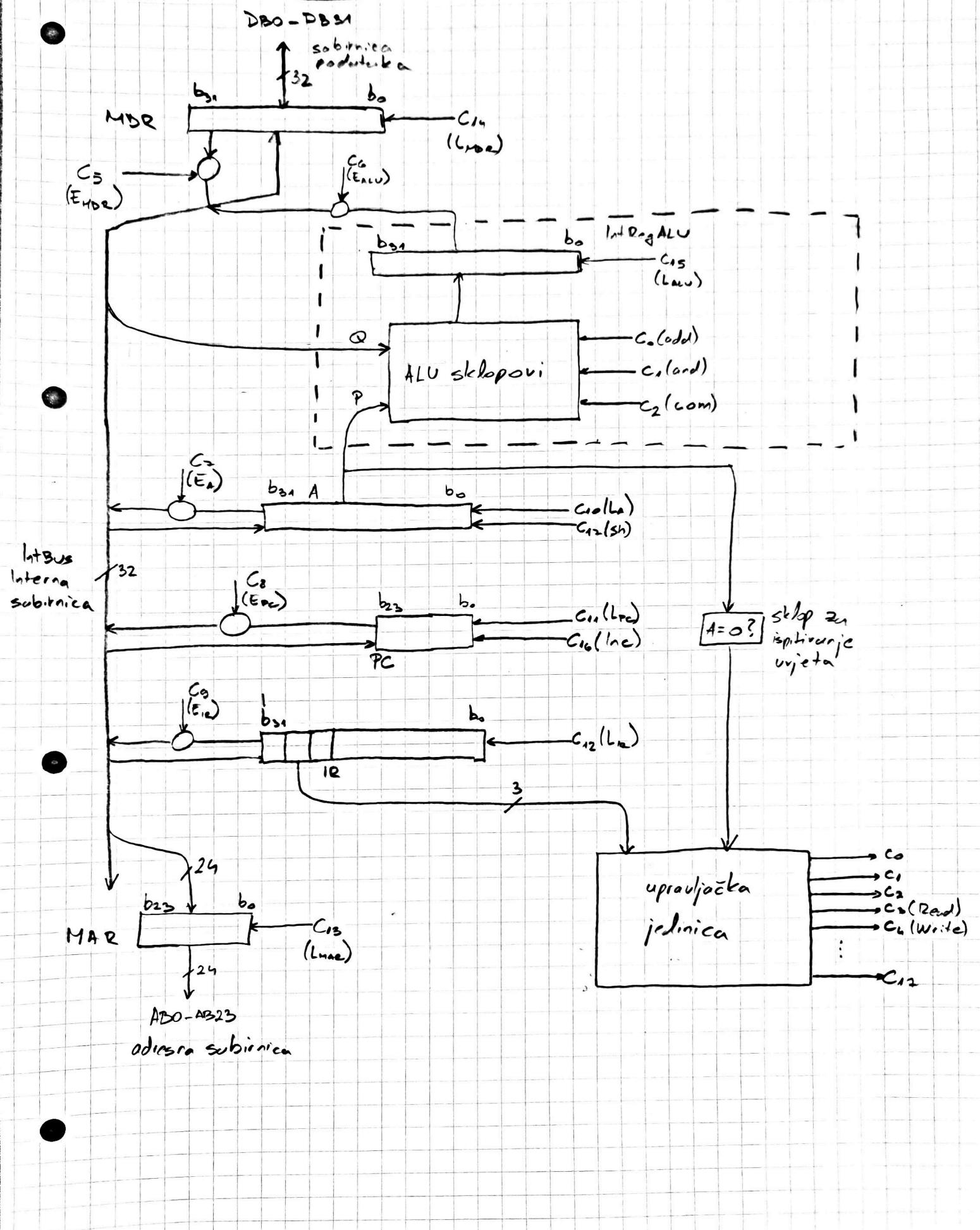
④ $A \leftarrow MDR$

$$\rightarrow EMDR: C_5 = I_g \cdot (\phi_{16} + \phi_{17})$$

$$\rightarrow LA: C_{10} = I_g \cdot \phi_{17}$$

Zbog dodavanja novih instrukcija instrucijski dekoder mijenjamo iz 3/8 u 4/16. Iz instrucijskog registra uzimamo bit više [31:28], a i u generator sljedova dodajemo liniju više (po modulu 17).

Detaljni model organizacije računala akumulatorske arhitekture



PONOVIMO:

detaljna izvedba faze pribavlj.

$$① \text{MAR} \leftarrow \text{PC}$$

$$\rightarrow \text{EPC}: C_8 = \phi_1 + \phi_2$$

$$\rightarrow \text{LMAR}: C_{13} = \phi_2$$

$$② \text{MDR} \rightarrow M[\text{MAR}]$$

$$\rightarrow \text{READ}: C_3 = \phi_3 + \phi_4$$

$$\rightarrow \text{LMDR}: C_{14} = \phi_4$$

$$③ \text{PC} \leftarrow \text{PC} + 1, \text{IR} \leftarrow \text{MDR}[31:24]$$

$$\rightarrow \text{INC}: C_{16} = \phi_5$$

$$\rightarrow \text{EHDR}: C_5 = \phi_5 + \phi_6$$

$$\rightarrow \text{LIR}: C_{12} = \phi_6$$

④ dekodiranje operacijskog koda
 $\rightarrow \phi_7$

detaljna izvedba faze izvrši za instrukciju adda (I_2):

$$① \text{MAR} \leftarrow \text{MDR}[23:0]$$

$$\rightarrow \text{EMDR}: C_5 = I_2 \cdot (\phi_2 + \phi_3)$$

$$\rightarrow \text{LMAR}: C_{13} = I_2 \cdot \phi_3$$

$$② \text{MDR} \leftarrow M[\text{MAR}]$$

$$\rightarrow \text{READ}: C_3 = I_2 \cdot (\phi_{10} + \phi_{11})$$

$$\rightarrow \text{LMDR}: C_{14} = I_2 \cdot \phi_{11}$$

$$③ A \leftarrow A + \text{MDR}$$

$$\rightarrow \text{EHDR}: C_5 = I_2 \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow \text{ADD}: C_0 = I_2 \cdot (\phi_{12} + \phi_{13})$$

$$\rightarrow \text{LALU}: C_{15} = I_2 \cdot \phi_{13}$$

$$\rightarrow \text{EALU}: C_6 = I_2 \cdot (\phi_{14} + \phi_{15})$$

$$\rightarrow \text{LA}: C_{10} = I_2 \cdot \phi_{15}$$

detaljna izvedba faze izvrši za instrukciju sta (I_1):

$$① \text{MAR} \leftarrow \text{MDR}[23:0]$$

$$\rightarrow \text{EMDR}: C_5 = I_1 \cdot (\phi_8 + \phi_9)$$

$$\rightarrow \text{LMAR}: C_{13} = I_1 \cdot \phi_9$$

$$② \text{MDR} \leftarrow A$$

$$\rightarrow \text{EA}: C_2 = I_1 \cdot (\phi_{10} + \phi_{11})$$

$$\rightarrow \text{LMDR}: C_{14} = I_1 \cdot \phi_{11}$$

$$③ M[\text{MAR}] \leftarrow \text{MDR}$$

$$\rightarrow \text{WRITE}: C_6 = I_1 \cdot (\phi_{12} + \phi_{13})$$

detaljna izvedba faze izvrši za instrukciju jmpz (I_6):

$$① \text{PC} \leftarrow \text{MDR}[23:0], \text{ ali}$$

samo ako $A = 0$

$$\rightarrow \text{EMDR}: C_5 = I_6 \cdot (\phi_8 + \phi_9)$$

$$\rightarrow \text{LPC}: C_{10} = I_6 \cdot \phi_9 \cdot (A=0)$$

detaljna izvedba faze izvrši za instrukciju com (I_5):

$$① A \leftarrow 7A$$

$$\rightarrow \text{COM}: C_2 = I_5 \cdot \phi_8$$

$$\rightarrow \text{LALU}: C_{15} = I_5 \cdot \phi_8$$

$$\rightarrow \text{EALU}: C_6 = I_5 \cdot (\phi_9 + \phi_{10})$$

$$\rightarrow \text{LA}: C_{10} = I_5 \cdot \phi_{10}$$

detaljna izvedba faze izvrši za instrukciju shr (I_7):

$$① A \leftarrow \text{shr}(A)$$

$$\rightarrow \text{SHR}: C_{12} = I_7 \cdot \phi_8$$

- * (a) procesor; superskalarni konvencionalne arhitekture
 može izdržati do 4 instrukcije u jednoj periodi signala vremenskog vodenja
 protobne strukture nisu specijalizirane i trebaju se izvesti sledeće tri direkcie:

* JIR, 2014./2015. (5.zadatak)
 + prezentacija

direkt1:

| | | | | | | | |
|---|---|---|---|--|--|--|--|
| 1 | 1 | | | | | | |
| 1 | | | | | | | |
| 1 | 1 | 1 | | | | | |
| 1 | 1 | | | | | | |
| 1 | 1 | 1 | 1 | | | | |
| | | | | | | | |
| 1 | | | | | | | |
| 1 | 1 | 1 | | | | | |
| 1 | 1 | | | | | | |

direkt2:

| | | | | | | | |
|---|---|---|---|--|--|--|--|
| 2 | 2 | 2 | | | | | |
| 2 | 2 | | | | | | |
| 2 | | | | | | | |
| 2 | | | | | | | |
| . | | | | | | | |
| 2 | 2 | 2 | 2 | | | | |
| 2 | 2 | | | | | | |
| 2 | 2 | | | | | | |

direkt3:

| | | | | | | | |
|---|---|---|--|--|--|--|--|
| 3 | 3 | 3 | | | | | |
| 3 | 3 | | | | | | |
| 3 | | | | | | | |
| 3 | | | | | | | |
| . | | | | | | | |
| 3 | 3 | 3 | | | | | |
| 3 | 3 | | | | | | |
| 3 | 3 | | | | | | |

!ješanje (a)

| | | | | | | | |
|---|---|---|---|--|--|--|--|
| 1 | 1 | | | | | | |
| 1 | 1 | 1 | 1 | | | | |
| 1 | 1 | | | | | | |
| 2 | 2 | 2 | 2 | | | | |
| 2 | 2 | | | | | | |
| 2 | | | | | | | |
| 2 | | | | | | | |
| 2 | 2 | 2 | 2 | | | | |
| 2 | 2 | | | | | | |
| 2 | 2 | | | | | | |
| 3 | 3 | 3 | | | | | |
| 3 | | | | | | | |
| 3 | | | | | | | |
| 3 | 3 | 3 | | | | | |
| 3 | 3 | | | | | | |
| 3 | 3 | | | | | | |

(d) superskalarni simultano višedretveni
 procesor, prot. struk. nisu specijaliz.

???

| | | | | | | | |
|---|---|---|---|--|--|--|--|
| 1 | 1 | 2 | 2 | | | | |
| 2 | 3 | 3 | 3 | | | | |
| 1 | 2 | 2 | 1 | | | | |
| 1 | 1 | 1 | 2 | | | | |
| 3 | 3 | 1 | 1 | | | | |
| 2 | 3 | 1 | 1 | | | | |
| 2 | 3 | 1 | 1 | | | | |
| 3 | 3 | 3 | 1 | | | | |
| 2 | 2 | 2 | 2 | | | | |
| 1 | 2 | 2 | 1 | | | | |
| 1 | 1 | 1 | 2 | | | | |
| 1 | 1 | 2 | 1 | | | | |

 nastavak

* JIR, 2014./2015. (5. zadatak)

+ prezentacija

(b) višedretveni (finozrnat) superskalarni procesor

(c) višedretveni (gritozrnat) superskalarni procesor

a prosijanje se dogodi kod vertikalno neiskorištenih priključaka i traje jedan periodu

| | | | |
|---|---|---|---|
| 1 | 1 | | |
| 2 | 2 | 2 | |
| 3 | 3 | 3 | |
| 1 | | | |
| 2 | 2 | | |
| 3 | 3 | | |
| 1 | 1 | 1 | |
| 2 | | | |
| 3 | | | |
| 1 | 1 | 1 | 1 |
| 2 | | | |
| 3 | 3 | 3 | |
| 1 | | | |
| 2 | 2 | 2 | 2 |
| 1 | 1 | 1 | |
| 2 | 2 | | |
| 1 | 1 | | |
| 2 | 2 | | |

| | | | |
|---|---|---|---|
| 1 | 1 | | |
| 1 | | | |
| 1 | 1 | 1 | |
| 1 | 1 | | |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | |
| 2 | 2 | | |
| 2 | | | |
| 2 | | | |
| 3 | 3 | 3 | |
| 1 | | | |
| 1 | 1 | 1 | |
| 1 | 1 | | |
| 2 | 2 | 2 | 2 |
| 2 | 2 | | |
| 2 | 2 | | |
| 3 | | | |
| 3 | | | |
| 3 | 3 | 3 | |