BLICEVI I ZADACI

ZA

1. LABORATORIJSKU VJEŽBU

sve je kopirano iz prošlogodišnjih tema sa FER2, god 2012/2013 nadalje

rješenja su također kopirana

PITANJA IZ BLICEVA

1) Gdje se spremaju mikroprogrami?  
- U posebnoj memoriji u upravljačkoj jedinici (tako nešto slično je bilo ponuđeno)  
  
2) Ako neka makroinstrukcija za izvođenje koristi mikroprogram sa tri mikroinstrukcije koliko onda zauzima mjesta u glavnoj memoriji?  
- Tu je mislim bilo ponuđeno 2B, 3B, 4B, i 3\*32b, nisam siguran kaj je točno pa nisam zaokružio  
  
3) Ako želimo izbrisati sadržaj nekog registra (postaviti na 0), koristi ćemo mikroinstrukciju:  
- Ne sjećam se točno instrukcija, no točna je ona koja sprema dva puta iste podatke u registar (npr. r1 <- 6, r2 <- 6) i zatim se na ALU odabere XOR i spremi u isti registar  
  
4) Statusni registar se u navedenom modelu nalazi:  
- U jednom od registara od r0 do r7 (inače je u r6, ali nije bilo ponuđeno konkretno)  
  
5) U našem modelu, IR se sastoji od:  
- 16 bita, od kojih prvih 6 bita ide operacijski kod (mislim da je to točno jer se MAR i MDR tretiraju posebno, ispravite me ako sam u krivu)  
  
Ostale su mi bile aritmetičke naredbe (tipa ako je u ovom registru spremljeno to, a   
u drugom ono, uz ove naredbe rezultat će bit bla bla) ... tak da se ne sjećam najbolje ..  
  
6) U koliko se najmanje mikroinstrukcija može ostvariti JMP(addr)?  
- Ja sam stavio 1 jer je se može samo stavit adresa u PC, možda sam u krivu no mislim da se može tak. Bilo je još ponuđeno 2,3,4 ...  
  
7) Ako se u svim registrima nalazi broj 7, i zatim izvede naredba XOR nad r6 i r7, te se rezultati spreme u r1 i r2, što će biti nakon toga u registrima? (Nije bilo baš tako zadano već mikroinstrukcijama, no poanta je ista)  
- Bit će: r1=0, r2=0, r6=7, r7=7

9) Što radi sljedeći kod:  
Makroinstrukcija: 000001 00 01 10 0000  
Mikroinstrukcija: opcode[1]: rj\_sel, rk\_sel, a\_sel=4, b\_sel=3, alu\_sel=ADD, r0\_write, goto fetch0; ?  
r1 + r2 -> r0  
  
10) Ako se mikro instrukcija sastoji od 5 instrukcija koliko će bita memorije trošiti u računalu?  
a) 16\*5  
b) 32  
c) 5  
d) 2  
e) 16  
(dunno?)  
  
11) Koliko bitova ima IR?  
 (malo sam se preduhitrio ipak je 16)  
  
12) S koliko bitova je definirana instrukcija (nešto u tom stilu)  
 6  
  
13) Kako je riješen PC u simulatoru?  
Simulator nema PC, koristimo R7 kao PC  
  
14) Od ponuđenih funkcija koja koristi samo 1 registar  
SUBA  
  
15) Neka mikroinstrukcija, nemogu se sjetiti, kako izgleda stanje registara nakon što se ona izvede,  
ako je u svim registrima u početku pisalo (neki broj).

16) Sto se tice onog zadatka ovako je isao:  
U svim opcim registrima upisan je broj 7. Nakon slijedece mikroinstrukcije sta ce se desiti?  
a\_sel=3, b\_sel=4, alu\_sel=ADDA, r1\_write, r2\_write;  
  
S tim da ovdje gledaju da su opci registri R1-R4.  
Uglavnom odgovor je: R1=7 R2=7 R3=7 R4=7

17) makroinstrukcija zauzima 16 bita ili 2 bajta u glavnoj memoriji, nebitno je koliko ima mikroinstrukcija jer se to ne sprema u glavnu memoriju

18) U svim registrima r0-r3 se nalazi vrijednost 7. Propuštaju se r3 i r4 na ALU,a ko se izvede SUBA i mikroinstrukcijske naredbe r0\_write i r1\_write, što će bit u registrima?  
  
R: r0=6, r1 = 6, r2=7, r3=7  
  
19) S obzirom na veličinu MDR-a i MAR-a, od koliko adresa je građena memorija? (nešto u tom smislu)  
  
R: Nisam siguran ali ako u IR možemo spremit adresu kao konstantu u donjih 8 bitova a MDR i MAR su veličine 8 bitova, onda mislim da je 2^8 8-bitnih adresa = 256 8-bitnih adresa  
  
20) U ovom simulatoru se ne može obaviti koja ALU operacija?  
  
R: NOR  
  
21) Gdje (ili kako) je zadan početak koda (prva instrukcija)?  
  
Odgovori: 1.) Jednoznačno u PCu 2.) Operacijom ADD nad adress\_true i adress\_false 3.) U makroprogramu 4.)....  
R: Nisam siguran   
  
22) Kakve konstante mogu biti u ovom simulatoru? (nešto takvo)  
  
R: 8-bitne ili 4-bitne s predznačnim proširenjem  
  
24) Ako želimo izvesti operaciju PUSH, koristit ćemo sljedeće mikroinstrukcije:  
  
R: Tu je malo navlakuša koliko sam skužio jer imaju odgovori gdje se na ALU pušta a\_sel = 6 ili a\_sel = SP. r6 je SR ali koliko ja znam a\_sel se ne može zadati kao a\_sel = SP nego samo broj pa onda ovdje kao SP koristimo registar r6 a ne r5 kako bi trebalo bit. I onda ima još SUBA ili SUBA i c\_in, onda bi trebala bit ova operacija koja nema aktivan c\_in jer SUBA radi sljedeće: a\_bus - 1 + c\_in i sad ako aktiviramo c\_in rezultat će ostat isti ( a\_bus - 1 + 1 = a\_bus ) pa nećemo smanjit kazaljku stoga.  
  
25) Zadatak ko i prošle godine, aktivni i a\_sel i b\_sel i rj\_sel i rk\_sel, sprema se u r0 s r0\_write i ima zadan makrokod. rj\_sel i rk\_sel imaju veći prioritet pa nema veze što su aktivni i a\_sel i b\_sel, gleda se po makrokodu koja 2 registra se zbrajaju i spremaju u r0  
  
26) Jedna makroinstrukcija zadana je s 5 mirkoinstrukcija, koliko zauzima mjesta u memoriji?  
  
R: Trebalo bi bit 2B, mikroinstrukcije imaju svoju memoriju a svaka markinstrukcija se sastoji od 16 bitova = 2B

27) s obzirom na velicinu MAR-a btw. naš ima 8 bitova, kolka je MAX veličina memorije, ja mislim da je 256 8- bitnih riječi  
  
28) u svim općim registrima su zapisane sedmice (7) ako mikro instrukcija glasi  
a\_sel = 3, b\_sel = 4, alu\_sel=SUB,r0\_write,r1\_write; stanja registara?  
mislim da je r0=0,r1=0,r2=7,r3=7 je bilo jedno od ponuđenih  
  
29)koliko najmanje mikroinstrukcija treba strojnoj instrukciji JMP addr  
mislim da je 1  
  
30) gdje su spremljene mikroinstrukcije?  
u posebnu upravljačku memoriju  
  
31) kako se znamo kolko traje čitanje i pisanje iz memorije  
mislim da je prema wait signalu, da nije unaprijed određeno, ali nisam siguran  
  
32) nešto tipa ako je zadana makroinstrukcija tipa 100010 01 10 11 0000  
i mikro instrukcije: rj\_sel,rk\_rel,ri\_sel,alu\_sel=ADD; što se događa:  
r1 <- r2 + r3  
  
ili je bilo sa a\_sel = 3,b\_sel =4 ne znam više... ugl nešto s tim određivanjem registara...  
  
33) koliko je velika makroinstrukcija ako postoji 5 mikroinstrukcija tako nešto, ugl. odgovor je 2B (16 bitova) jer je makronaredba uvijek te veličine  
  
34) gdje je operacijski kod u makronaredbi, nešto tipa to. uglavnom u najviših 6 bitova je op kod... u pripremi piše točno di je ostalo...

ZADACI NA LABOSU

1) Za napisati nam je došlo LOAD ri, (rj)  
S time da u r0 je zapisan broj 50 i na 50oj lokaciji je neki broj.  
I sada nisam ziher ali mislim da je zadano da napišeš LOAD r1 r0 (možda nei drugi registri)  
Aha, i nemorate pisati fetch, pretpostavlja se da je napisan...

rješenje:

.ucode

// ============= DIO OPERACIJSKIH KODOVA =============  
// 1) LOAD ri, (rj)  
opcode[1]: a\_sel=4, b\_sel=4, alu\_sel=XOR, r4\_write, goto opcode1.1;  
  
// ================= DIO EKSTENZIJE ================  
// LOAD ri, (rj)  
opcode1.1: rj\_sel, b\_sel=4, alu\_sel=OR, mar\_sel=LOAD, goto opcode1.2;  
opcode1.2: ir0\_sel=LOAD, read, if wait then goto opcode1.2 else goto opcode1.3 endif;  
opcode1.3: result\_sel=IR\_CONST8, ri\_sel, goto fetch0;

.mem

0: 000001 01  
1: 00 00 0000

2) zadatak napisati instrukciju STZR ri, postaviti sadržaj ri na 0

rješenje:

Ako moras smjestit 0 u ri jednostavno se napise:  
opcode[1]: a\_sel=4, b\_sel=4, alu\_sel=XOR, ri\_sel;  
i zapisat ce se 0 u ri...

3) napisati mikroprogram za makroinstrukciju: asl rj (aritmeticki posmak sadrzaja registra rj u lijevo za 1 mjesto). neka je operacijski kod instrukcije asl jednak 000100 pretpostavite da je mikroprogram za fazu pribavi vec definiran a da prva mikroinstrukcija pocinje sa fetch0. uloge registara su- r7 pc, r6 sr, r5 sp, r4 pomocni reg.

Moguće rješenje:

opcode[1]: a\_sel=4, b\_sel=4, alu\_sel=XOR, r4\_write, goto opcode1.1; //u r4=0  
  
opcode1.1: a\_sel=4, c\_in, alu\_sel=ADDA, r4\_write; //u r4 = 00000001;  
opcode1.2: rj\_sel, b\_sel=4, alu\_sel=AND, r4\_write; //spremio najnizi bit od rj u r4  
opcode1.3: rj\_sel, b\_sel=4, alu\_sel=ADD, r4\_write; //zbrojio najnizi bit sa rj  
opcode1.4: rj\_sel, b\_sel=4, alu\_sel=ADD, ri\_sel; //jos jednom zbrojio, ali ovaj put spremio u ri

4) u ri zapisati sumu rj i sadrzaj mem lok (rk + konst)

5)nama je bilo ostvariti ri <- rj + mem(konst + rk)

6) zadatak je bio aritmetički posmak u lijevo, ASL ri,rj

7) ROTL ri, rj   
rotiraj bitove iz registra rj i rezultat spremi u ri.  
r7 - PC,r6 - SR, r5 -SP, r4 - pomocni. ne treba postavljati zastavice za rezultat. napiši mikro i makro (s tim da je OP kod od ROTL zadan)

rješenje:

opcoide[1]: rj\_sel,alu\_sel=ADDA,r4\_write,goto opcode 1.1;

opcode 1.1: a\_sel = 4,b\_sel = 4,alu\_sel=ADD,r4\_write;

if c\_out then goto opcode 1.2 else goto opcode 1.3;

opcode 1.2: a\_sel = 4,c\_in,alu\_sel=ADDA,ri\_sel,goto fetch0;

opcode 1.3: a\_sel = 4,alu\_sel=ADDA,ri\_sel,goto fetch0;

8) zadatak- napiši naredbu ADD ri, rj, konst(rk) ==> ri<- rj+ MEM(konst+rk);

rješenje:

opcode[6]: result\_sel=IR\_CONST4, r4\_write;

rk\_sel, a\_sel=4, alu\_sel=ADD, mar\_sel=LOAD, goto opcode6.1;

opcode6.1: mdr\_sel=LOAD\_MEM, read, if wait then goto opcode6.1 endif;

result\_sel=MDR, r4\_write;

ri\_sel, rj\_sel, b\_sel=4, alu\_sel=ADD, goto fetch0;

9) LOAD Ri, konst(Rj) // MAR <- ir\_const4 + Rj, Ri <- MEM(MAR)

rješenje:

opcode[10]: result\_sel = ir\_const4, rj\_sel, b\_sel=4, alu\_sel=ADD, mar\_sel=LOAD, goto opcode10.1;

opcode10.1: read, while wait goto opcode 10.1 endif;

mdr\_sel=LOAD\_MEM, result\_sel=mdr, ri\_sel, goto fetch0;

10) zadatak- napiši naredbu ADD ri, rj, konst(rk) ==> ri<- rj+ MEM(konst+rk);

rješenje:

opcode[6]: result\_sel=IR\_CONST4, r4\_write;

rk\_sel, a\_sel=4, alu\_sel=ADD, mar\_sel=LOAD, goto opcode6.1;

opcode6.1: mdr\_sel=LOAD\_MEM, read, if wait then goto opcode6.1 endif;

result\_sel=MDR, r4\_write;

ri\_sel, rj\_sel, b\_sel=4, alu\_sel=ADD, goto fetch0;

%  
0: 000110 00  
1: 01 10 0101