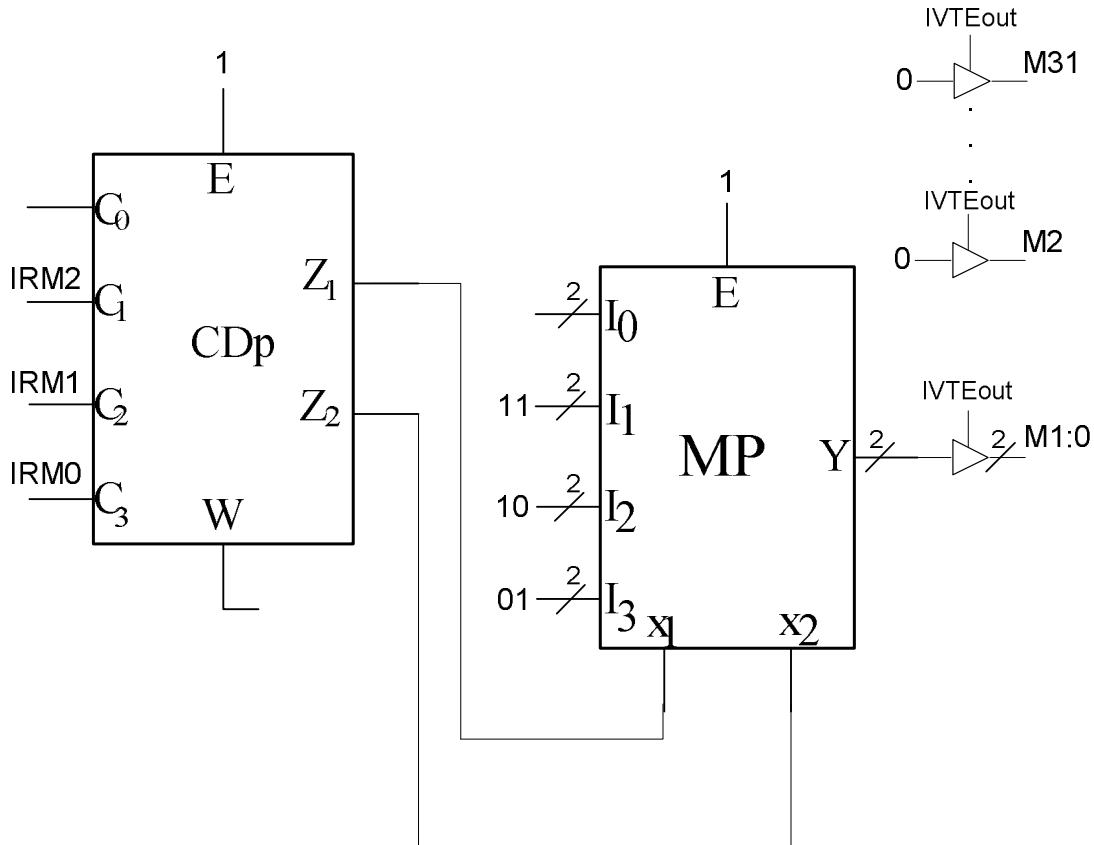


Ispit iz Arhitekture i organizacije računara 2

a) (5p)



b) (20p)

2p

```
; Dohvatanje instrukcije
BEGIN:   PCout, MARin, Xin
         read, incA, ALUout, PCin
         wmfc
         MDRout, IRin
; Dekodovanje instrukcije
         opcase
; LOAD instrukcija
LOAD:   admodld          ; način adresiranja za LOAD
```

2p

```
; Neposredno adresiranje
LDIMM:  PCout, MARin, Xin
         read, incA, ALUout, PCin
         wmfc
         MDRout, regsel1, REGin, branch(IRR, INTH)
         branch(, BEGIN)
```

3p

```
; Memorijsko direktno adresiranje
LDMD:   PCout, MARin, Xin
         read, incA, ALUout, PCin
         wmfc
         MDRout, MARin
         read
```

```

                                wmfC
                                MDRout, regsel1, REGin, branch(IRR, INTH)
                                branch(, BEGIN)

2p
; Registrarsko direktno adresiranje
LDRD:    regsel2, REGout, TEMPin
          TEMPout, regsel1, REGin, branch(IRR, INTH)
          branch(, BEGIN)

3p
; Registrarsko indirektno adresiranje
LDRI:    regsel2, REGout, MARin
          read
          wmfC
          MDRout, regsel1, REGin, branch(IRR, INTH)
          branch(, BEGIN)

3p
; Registrarsko indirektno adresiranje sa pomerajem
LDRISP:   PCout, MARin, Xin
          read, incA, ALUout, PCin
          wmfC
          MDRout, Xin
          regsel2, REGout, Yin
          add, ALUout, MARin
          read
          wmfC
          MDRout, regsel1, REGin, branch(IRR, INTH)
          branch(, BEGIN)

5p
; Obrada prekida
INTH:     SPout, MARin
          PCout, MDRin
          write, SPout, Xin
          wmfC
          incA, ALUout, MARin, SPin
          PSWout, MDRin
          write, SPout, Xin
          wmfC
          incA, ALUout, SPin
          IVTEout, Yin
          IVTPout, Xin
          add, ALUout, MARin
          read
          wmfC
          MDRout, PCin, clPSWI, branch(, BEGIN)

```

```

c) (5p)
START:    OR      R3, R3, R3    ; R3:=R3 OR R3
          JZ      END          ; if(R3=0)goto end
          LOAD    R1, #0h      ; R1:=0 (pokazivač i)
          LOAD    R2, #1h      ; indikator parno/neparno(0/1)
LOOP:     LOAD    R0, 100h[R1]; R0:=a[i]
          OR      R2, R2, R2    ; R2:=R2 OR R2
          JZ      PAR          ; if(R2=0)goto par
          ADD     R0, R0, R4     ; R0:=R0+R4
          DEC     R2            ; R2:=R2-1 - sledeci paran
          JMP     ENDLOOP       ; goto endloop
PAR:      SUB     R0, R0, R4     ; R0:=R0-R4
          INC     R2            ; R2:=R2+1 - sledeci neparan
ENDLOOP:  STORE   R0, 100h[R1]; a[i]:=R2
          INC     R1            ; R1:=R1+1
          DEC     R3            ; R3:=R3-1
          JNZ     LOOP          ; NEXT
END:

```