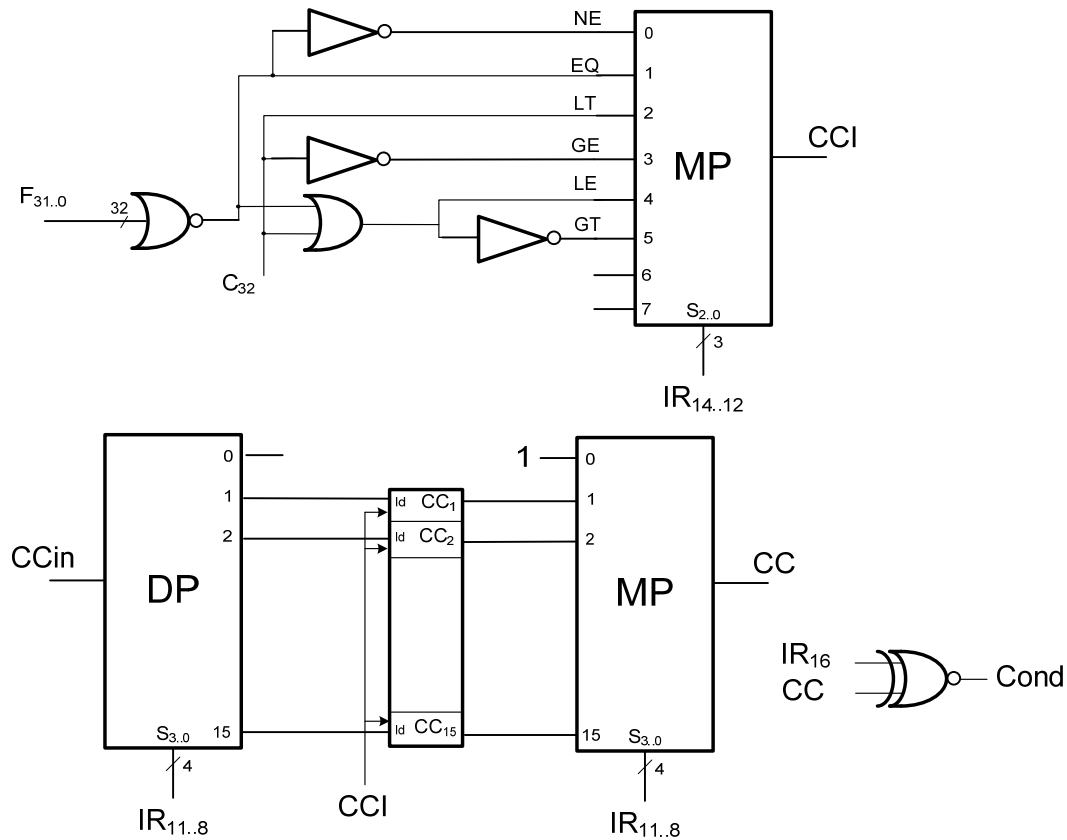


## Rešenja zadataka iz Arhitekture i organizacije računara 2, jun 2009.

1. a)



b)

```

; Instruction fetch
BEGIN:    PCout,MARin,Xin
          read,incA,ALUout,PCin
          wmfcc
          MDRout,IRin

; Instruction decode
          opcode

; Conditional instructions
COND:    regsel1,REGout,Xin
          regsel2,REGout,Yin
          sub,CCin,branch(!IR,BEGIN)
          bruncnd(INTH)

; ALU instructions
ALU:     regsel1,REGout,Xin, branch(!cond, lab1)
          regsel2,REGout,Yin
          ALUop,ALUout, regsel3, REGin, branch(!IR,BEGIN)
lab1:    branch(IR, INTH)
          bruncnd(BEGIN)
    
```

```

c)   SUB    CC0, R0, R0, R0    ; R0:=0
      LOAD  R5, #5
      LOAD  R6, #6
      LOAD  R3,b(R0)
      LOAD  R4,c(R0)
      ADD   CC0,R2,R3,R4      ; *   (1)
      GE    CC1,R3,R4        ; *   (2)
      ADD   CC1,R2,R2,R5      ; *   (3)
      SUB   !CC1,R2,R2,R6     ; *   (4)

```

Deo koda označen sa \* bi u klasičnoj arhitekturi izgledao ovako:

```

      ADD   R2,R3,R4
      CMP   R3,R4
      JN    LT           ; branch on less than (<0)
      ADD   R2, R2, R5    ; a:=a+5
      JMP   END          ; jump on END
LT:     SUB   R2, R2, R6   ; a:=a-6
END:    ...

```

U ovom primeru instrukcija (1) u prvom slučaju odgovara instrukciji ADD u drugom, instrukcija (2) odgovara instrukciji CMP, a instrukcije (3) i (4) instrukcijama ADD i SUB respektivno. U klasičnoj arhitekturi su, međutim potrebne i instrukcije uslovnog i безусловnog skoka koje u datoj realizaciji ne postoje. Prema tome, u klasičnoj arhitekturi je, bar za ovaj primer, potrebno više instrukcija za zapis istog segmenta. Takođe se kod klasične arhitekture izvršava u oba slučaja ishoda uslova instrukcija uslovnog skoka, a u jednom slučaju i instrukcija безусловnog skoka više nego kod date arhitekture. Kod date arhitekture se, opet, dohvata instrukcija više u oba slučaja ishoda koja se zapravo ne izvršava.