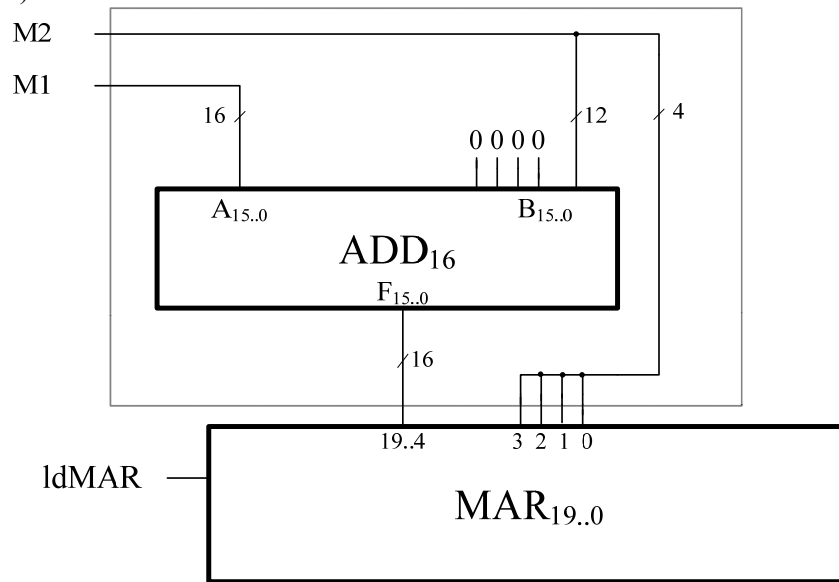
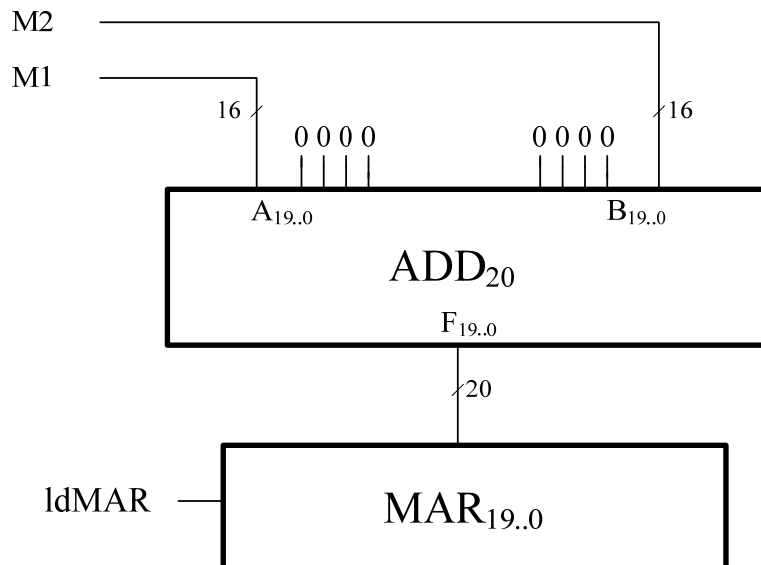


a)



Priznaje se i sledeće (“skuplje”) rešenje:



b)

```
begin:  IPout, CSout, ldMAR, XM2in
        read, ALUinc, ALUout, G, IPin
        wmfcc
        MDRout, IR1in
        IPout, CSout, ldMAR, XM2in, branch(!I, adrmod)
        read, ALUinc, ALUout, G, IPin
        wmfcc
        MDRout, IR2in, branch(STORE, LStore)
```

```
; čitanje prvog operanda, ako ga ima i ako ga treba čitati
    adrmod
```

```
memdir:    DSout, IR2out, MARin, bruncnd(readop)
```

```
regind:    regout, XM1in, IR2out, Yin
            ALUadd, ALUout, DSout, MARin
```

```
readop:    read
            wmfcc
            MDRout, TEMPin, opcase
```

```
regdir:    regout, TEMPin, opcase
```

```
imm, bezadresna:    IR2out, G, TEMPin, opcase
```

```
FJMP:      regsel, regout, CSin, bruncnd(JMP)
```

```
JZNVC:     branch(!cnd, NOP)
```

```
JMP:       TEMPout, IPin, branch(!IR, begin)
            bruncnd(INTH)
```

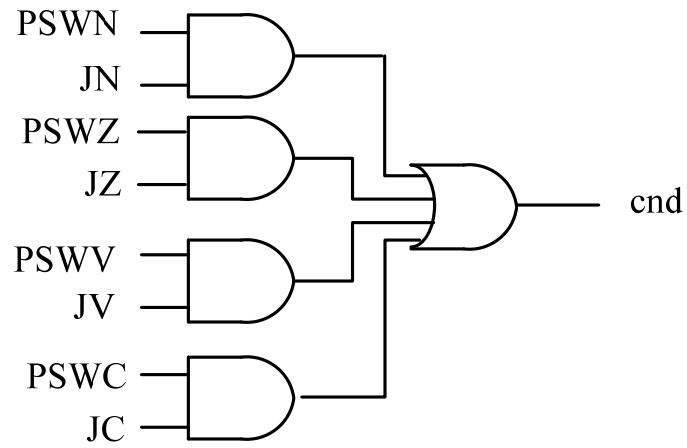
```
FCALL:     SPout, XM1in, SSout, ldMAR
            CSout, MDRin
            write, ALUinc, ALUout, G, SPin
            wmfcc
```

```
CALL:      regsel, regout, CSin
            SPout, XM1in, SSout, ldMAR
            IPout, G, MDRin
            write, ALUinc, ALUout, G, SPin, XM2in
            wmfcc
            TEMPout, IPin, branch(!IR, begin)
            bruncnd(INTH)
```

```
FRET, RET: SPout, XM2in
            ALUdec, ALUout, XM2in, SSout, ldMAR
            read, ALUdec, ALUout, G, SPin
            wmfcc
            MDRout, IPin, branch(RET, NOP)
            ALUdec, ALUout, SSout, ldMAR
            read, ALUdec, ALUout, G, SPin
            wmfcc
```

```
MDRout, CSin, branch(IR, INTH)
```

```
NOP:       branch(!IR, begin)
            bruncnd(INTH)
```



c)

```

      CMP  R0, #0
      JZ   LabB
      JMP  LabC
LabB:  FJMP LabA
LabC:  ADD  R1, R2
      ...
  
```