



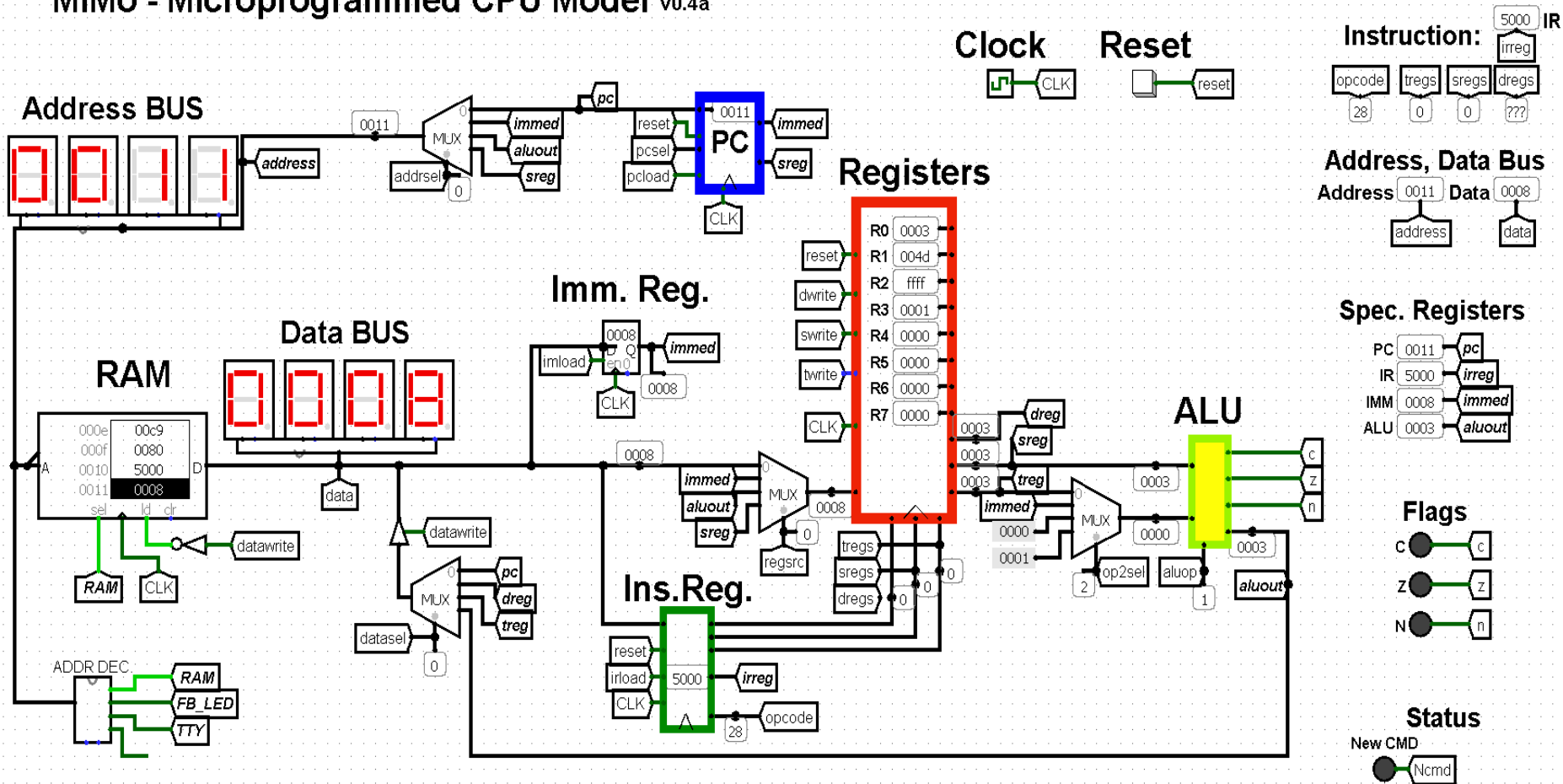
# ORGANIZACIJA RAČUNALNIKOV

## Laboratorijske vaje

### **Vaja 5: Implementacija strojnih ukazov z mikropodprogrami v MiMo**

# MiMo – Podatkovna enota

MiMo - Microprogrammed CPU Model v0.4a



# MiMo – Kontrolna enota

Vsak mikroukaz določa :

Mikroukaz = elementarni korak

- stanje vseh ?
- naslednji ?

Vhodi v KE: 

**opcode** – operacijska koda ukaza  
**C, Z, N** zastavice

Izhodi iz KE 

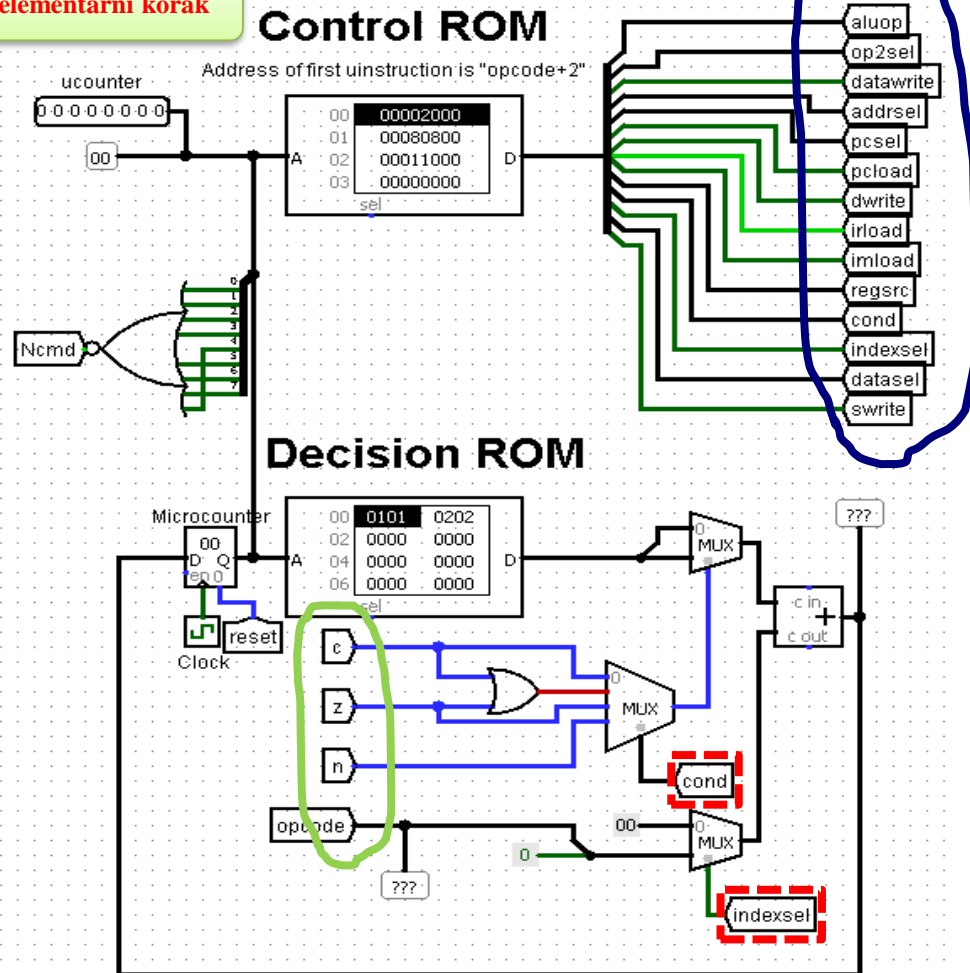
Vsi kontrolni signali

Kontrolni signali:

**cond** – izbira pogoja (C, CorZ, Z, N)

**indexsel** – opcode\_jump  
 (skok na opcode+2)  
 $u\text{counter}(uPC)=2$

## Microcode Control Unit



# Implementacija ukazov v mikroprogramski CPE MiMo

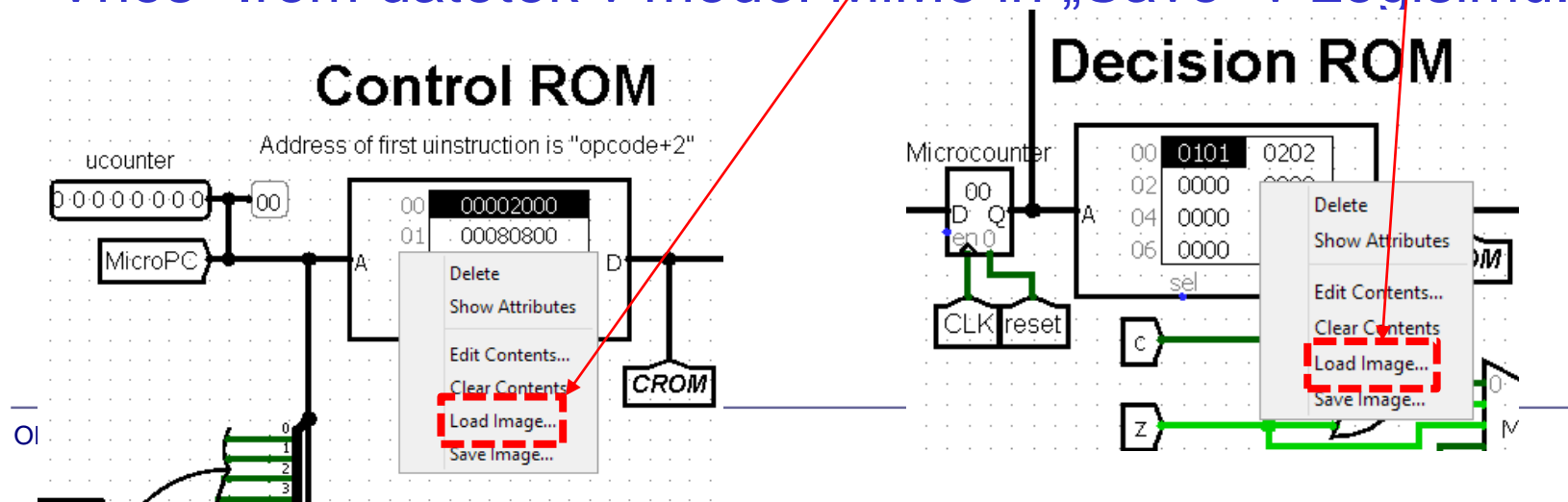
## I. Mikroprogramski nivo:

1. Mikroprogramska realizacija v *basic\_microcode.def*
2. Prevajanje: *basic\_microcode.def* -> *ucontrol,udecision.rom*

```
# sub Rd,Rs,Rt (1)
#   Rd <- Rs - Rt      PC <- PC + 1
1: aluop=sub op2sel=treg dwrite=1 regsrc=aluout, goto fetch
```

```
C:\winIDEA\MiMo\Distribucija_2017_18>micro_assembler.exe basic_microcode_sub.def
00: 00002000 0101      # fetch:      addrsel=pc imload=1
01: 00080800 0202      #           pload=1 pcsel=pc, opcode_jump
02: 00011000 0000      # 0:         aluop=add op2sel=treg dwrite=1 regsrc=aluout, goto fetch
03: 00011001 0000      # 1:         aluop=sub op2sel=treg dwrite=1 regsrc=aluout, goto fetch
2a: 00004000 8282      # 40:        addrsel=pc imload=1
41: 00001000 8484      # 63:        addrsel=pc dwrite=1 regsrc=databus, goto pcincr
43: 00004000 8383      # 65:        addrsel=pc imload=1
82: 00040021 8485      #           aluop=sub op2sel=const0, if z then pcincr else jump
83: 001000c0 8484      #           addrsel=immed databwrite=1 datasel=dreg, goto pcincr
84: 00000800 0000      # pcincr:    pload=1 pcsel=pc, goto fetch
85: 00000a00 0000      # jump:      pload=1 pcsel=immed, goto fetch
```

## 3. Vnos \*.rom datotek v model MiMo in „Save“ v Logisimu:



## II. Nivo zbirnega jezika:

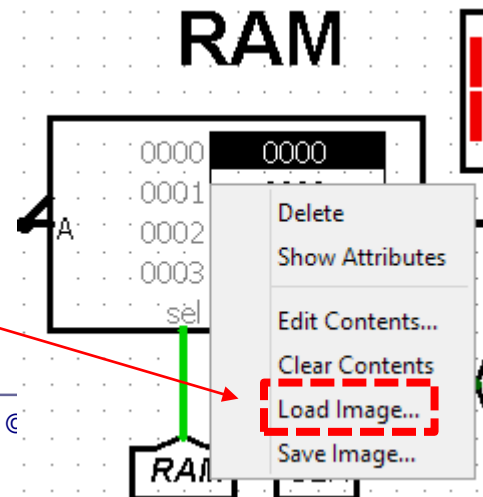
### 1. Uporaba ukaza v testnem programu:

```
main:  li  r1, 2           # r1 is the counter
      li  r2, 1           # Used to decrement r1
loop:  sub r1, r1, r2     # r1--
      jnez r1, loop      # loop if r1 != 0
      sw  r2, 16         # Save the r2
```

### 2. Prevajanje: *ime.s* -> *ime.ram*

```
C:\winIDEA\MiMo\Distribucija_2017_18>assembler.exe basic_program1_sub.s
0000: 00007e01 0111111000000001  main:  li  r1, 2
0001: 00000002 0000000000000010          li  r2, 1
0002: 00007e02 0111111000000010  loop:  sub r1, r1, r2
0003: 00000001 0000000000000001          jnez r1, loop
0004: 00000289 0000001010001001          sw  r2, 16
0005: 00005008 0101000000001000
0006: 00000004 0000000000000100
0007: 00008202 1000001000000010
0008: 00000010 0000000000001000
```

### 3. Vnos ime.ram datoteke v model MiMo :



## III. Preizkus delovanja:

1. Reset (po potrebi) :

# Reset



2. Izvajanje po mikroukazih:

- 2 x pritisk na „Ctrl+T“ (ena urina perioda)

3. Tekoče izvajanje (brez ustavljanja):

- Vklop
- Frekvenca urinega signala

