

Shema izvajanja programa v zbirniku v MiMo modelu

v 0.3

RAM		Format strojnega ukaza				Program v zbirniku	Kontrolni naslov		Mikroprogram	Decision ROM	
Nasl.	Vsebina strojni uk.	Op.koda	Treg	Sreg	Dreg	oznaka: ukaz operandi	Dec	Hex	Kontrolni signali, naslednji mikroukaz	T	F
							00	00	fetch: addrsel=pc irload=1	01	01
							01	01	pload=1 pcsel=pc, opcode_jump	02	02
0000:	7e01	63			1	main: li r1, 2	65	41	addrsel=pc dwrite=1 regsrc=databus, goto pcincr	84	84
0001:	0002	Tak. operand						84	pcincr: pload=1 pcsel=pc, goto fetch	00	00
0002:	7e02	63			2	li r2, -1	65	41	addrsel=pc dwrite=1 regsrc=databus, goto pcincr	84	84
0003:	ffff	Tak. operand						84	pcincr: pload=1 pcsel=pc, goto fetch	00	00
0004:	0089	0	2	1	1	loop: add r1,r1,r2	2	2	aluop=add op2sel=treg dwrite=1 regsrc=aluout, goto fetch	00	00
0005:	5008	40			1	jnez r1, loop	40	2a	addrsel=pc imload=1	82	82
0006:	0004	Tak. operand						82	aluop=sub op2sel=const0, if z then pcincr else jump	84	85
								84	pcincr: pload=1 pcsel=pc, goto fetch	00	00
								85	jump: pload=1 pcsel=immed, goto fetch	00	00
0007:	8202	65			1	sw r2, 16	67	43	addrsel=pc imload=1	83	83
0008:	0010	Tak. operand						83	addrsel=immed datawrite=1 datasel=dreg, goto pcincr	84	84
								84	pcincr: pload=1 pcsel=pc, goto fetch	00	00

Program: basic_program1.s :

```

main:  li r1, 2           # r1 is the counter
      li r2, -1         # Used to decrement r1
loop:  add r1, r1, r2    # r1<-r1+r2 (r2=-1 -> r1 decrements)
      jnez r1, loop     # if r1 != 0 then jump to loop:
      sw  r2, 16       # Save r2 to MEM[16]

```

```

0000: 00007e01 01111111000000001  main: li r1, 2
0001: 00000002 00000000000000010
0002: 00007e02 01111111000000010      li r2, -1
0003: 0000ffff 11111111111111111
0004: 00000089 0000000010001001      loop: add r1, r1, r2
0005: 00005008 0101000000001000      jnez r1, loop
0006: 00000004 0000000000000100
0007: 00008202 1000001000000010      sw r2, 16
0008: 00000010 00000000000010000

```

```

00: 00002000 0101 # fetch:addrsel=pc irload=1
01: 00008000 0202 # pload=1 pcsel=pc, opcode_jump
02: 00011000 0000 # 0: aluop=add 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 datawrite=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

```