EXAMPLE 4

Instruction Memory's file: 5th_fig451_MemEx4/inst.rom

This Mips pipeline implementation can handle branchs (with "nop" put by compiler) and can't handle data hazards

The user should set initial register values (linear) and the DataRam (linear + 2). No data values are required.

Description: A simple sequence of eight instructions with no data hazard.

LW r1,0(r4)
LW r2,4(r4)
NOP
NOP
ADD r3,r1,r2
NOP
NOP
SW r3, 8(r4)

LW r1,0(r4) – type I instruction
opcode = 35 rs = 4 rt = 1 imediate = 0
100011 00100 00001 0000000000000000
0x8C810000

LW r2,4(r4) – type I instruction
opcode = 35 rs = 4 rt = 2 imediate = 4
100011 00100 00010 0000000000000100
0x8C820004

NOP
0x00000000

NOP
0x00000000

ADD r3,r1,r2 – type R instruction
opcode = 0 rs = 1 rt = 2 rd = 3 sh = 0 func = 32
000000 00001 00010 00011 00000 100000
0x00221820

NOP
0x00000000

NOP
0x00000000

SW r3,8(r4) – type I instruction
opcode = 43 rs = 4 rt = 3 imediate = 8
101011 00100 00011 0000000000001000
0xAC830008

The hexadecimal code example is:

```
LW r1,0(r4)    – 0x8C810000
LW r2,4(r4)    – 0x8C820004
NOP            – 0x00000000
NOP            – 0x00000000
ADD r3,r1,r2   – 0x00221820
NOP            – 0x00000000
NOP            – 0x00000000
SW  r3, 8(r4)  – 0xAC830008
```

Calculations check (with linear initial register values):

1. LW r1,0(r4) – R1 = M[1] = 3
2. LW r2,4(r4) – R2 = M[2] = 4
3. NOP
4. NOP
5. ADD r3,r1,r2 – R3 = 7
6. NOP
7. NOP
8. SW r3, 8(r4) – M[3] = 7