EXAMPLE 2

Instruction Memory's file: 5th_fig451_MemEx2/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). No data values are required.

Description: A simple sequence of three "add" and four "nop" instructions with PIPE EXE data hazard.

ADD r2,r1,r3
NOP
NOP
ADD r3,r2,r1 //EXE HAZARD
NOP
NOP
ADD r1,r3,r2 //EXE HAZARD

ADD r2,r1,r3 – type R instruction
opcode = 0 rs = 1 rt = 3 rd = 2 sh = 0 func = 32
000000 00001 00011 00010 00000 100000
0x00231020

NOP
0x00000000

NOP
0x00000000

ADD r3,r2,r1 – type R instruction
opcode = 0 rs = 2 rt = 1 rd = 3 sh = 0 func = 32
000000 00010 00001 00011 00000 100000
0x00411820

NOP
0x00000000

NOP
0x00000000

ADD r1,r3,r2 – type R instruction
opcode = 0 rs = 3 rt = 2 rd = 1 sh = 0 func = 32
000000 00011 00010 00001 00000 100000
0x00620820

The hexadecimal code example is:

ADD r2,r1,r3 – 0x00231020
NOP – 0x00000000
NOP – 0x00000000
ADD r3,r2,r1 – 0x00411820
NOP – 0x00000000
NOP – 0x00000000
ADD r1,r3,r2 – 0x00620820

Calculations check (with linear initial register values):

ADD r2,r1,r3 – r2 = 4
NOP
NOP
ADD r3,r2,r1 – r3 = 5
NOP
NOP
ADD r1,r3,r2 – r1 = 9