EXAMPLE 3

Instruction Memory’s file: 5th_fig465_MemEx3/inst.rom

This Mips pipeline implementation can handle branchs (with "nop" put by compiler) and all data hazards

The user should set initial register values (linear). No data values are required.

Description: A simple sequence of three “add” instructions with PIPE MEM data hazard.

ADD r2,r1,r3
ADD r4,r4,r1
ADD r3,r2,r2 // MEM 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

ADD r4,r4,r1 – type R instruction

opcode = 0	rs = 4	rt = 1	rd = 4	sh = 0	func = 32
000000	00100	00001	00100	00000	100000

0x00812020

ADD r3,r2,r2 – type R instruction

opcode = 0	rs = 2	rt = 2	rd = 3	sh = 0	func = 32
000000	00010	00010	00011	00000	100000

0x00411820

The hexadecimal code example is:

ADD r2,r1,r3 – 0x00231020
ADD r3,r2,r1 – 0x00812020
ADD r1,r3,r2 – 0x00421820

Calculations check (with linear initial register values):

ADD r2,r1,r3 – r2 = 4
ADD r4,r4,r1 – r4 = 5
ADD r3,r2,r2 – r3 = 8