EXAMPLE 4

Instruction Memory's file: 5th_fig456_MemEx4/inst.rom

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

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

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

ADD r1,r1,r1
SLT r2, r15, r1
BEQ r2,r0, -3
NOP
NOP

ADD r1,r1,r1 – type R instruction

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

0x00210820

SLT r2,r15,r1 – type R instruction

opcode = 0	rs = 15	rt = 1	rd = 2	sh = 0	func = 42
000000	01111	00001	00010	00000	101010

0x01E1102A

BEQ r2,r0,-3 – type I instruction

opcode = 4	rs = 2	rt = 0	imediate = -3
000100	00010	00000	1111111111111101

0x1040FFFD

NOP
0x00000000

NOP
0x00000000

The hexadecimal code example is:

ADD r1,r1,r1 – 0x00210820
SLT r2, r15, r1 – 0x01E1102A
BEQ r2,r0, -3 – 0x1040FFFD
NOP – 0x00000000
NOP – 0x00000000

Calculations check (with linear initial register values):

1. ADD r1,r1,r1 – R1 = 2
2. SLT r2, r15, r1 – R2 = 0
3. BEQ r2,r0, -3 – BACK TO THE FIRST INSTRUCTION
4. NOP
5. NOP
    1. ADD r1,r1,r1 – R1 = 4
    2. SLT r2, r15, r1 – R2 = 0
    3. BEQ r2,r0, -3 – BACK TO THE FIRST INSTRUCTION
    4. NOP
    5. NOP
        1. ADD r1,r1,r1 – R1 = 8
        2. SLT r2, r15, r1 – R2 = 0
        3. BEQ r2,r0, -3 – BACK TO THE FIRST INSTRUCTION
        4. NOP
        5. NOP
            1. ADD r1,r1,r1 – R1 = 16
            2. SLT r2, r15, r1 – R2 = 0
            3. BEQ r2,r0, -3 – BRANCH IS NOT TAKEN
            4. NOP
            5. NOP – END.