Name: Ningyuan Zhang

Section: K

Instructor: Joseph Zambreno

Date: 10/23/2016

Home Work #6 Solutions

2).

(a). P2 may be more reasonale for me because data memory dose take more latency to process and its every instructions are taking reasonable latencies.

(b). The path for ALU instructions: I-Mem, Regs, Mux, ALU, Mux, Regs

P1: 400 + 200 + 30 + 120 + 30 + 200 = 980 ps

P2: 500 + 220 + 100 + 180 + 100 + 220 = 1320 ps

(c). The path for ALU instructions: I-Mem, Regs, Mux, ALU, D-Mem, Mux, Regs

P1: 400 + 200 + 30 + 120 + 350 + 30 + 200 = 1330 ps

P2: 500 + 220 + 100 + 180 + 1000 + 100 + 220 = 2320 ps

(d). The answer will be the same as part (c)

3).

(a). Bit 7 of the instruction word is only used as part of an immediate/offset part of the instruction, so one way to test would be to execute ADDI $1, zero, 128 which is supposed to place a value of 128 into $1. If instruction bit 7 is stuck at zero, $1 will be zero because value 128 has all bits at zero except bit 7.

(b). The test for stuck-at-zero requires an instruction that sets the signal to 1

and the test for stuck-at-1 requires an instruction that sets the signal to 0. Because

the signal cannot be both 0 and 1 in the same cycle, we cannot test the same signal

simultaneously for stuck-at-0 and stuck-at-1 using only one instruction.

(c) It is possible to work around this fault, but it is very diffi cult. We must fi nd all instructions that have zero in this bit of the offset or immediate operand and replace them with a sequence of “safe” instruction. For example, a load with such an offset must be replaced with an instruction that subtracts 128 from the address register, then the load (with the offset larger by 128 to set bit 7 of the offset to 1), then subtract 128 from the address register.

We cannot work around this problem, because it prevents all instructions from storing their

result in registers, except for load instructions. Load instructions only move data from memory to registers, so they cannot be used to emulate ALU operations “broken” by the fault.