HW-2

Enes Garip
150116034

1)

(4.6.1) If there is a stuck-at-0 fault on a wire, we should change the signal 0 to 1 to test that if there is a stuck-at-0 fault. The way to do that using load instruction. If we load an immediate value that is equals to zero there is two cases we may see. First one is if stuck-at-0 happened, the value of the register is the value that we load. Secondly, if there is not stuck-at-0 the value of the register is zero.

(4.6.2) We can not test stuck-at-0 and stuck-at-1 simultaneously because the wire can not be equal 0 and 1 at the same time. We can add 0 to the register with addi instruction. If there is a stuck-at-1 the value in the register will be a different value.

(4.6.3) It is possible if we find all the instructions that sets the registers to a value and changing them with other instructions to find correct values. Addition and subtraction instructions can be very helpful.

Enes Garip
150116034

2)

(4.8.1) Pipelined → 350 ps (the slowest operation)
Non-pipelined → 1250 ps (sum of all the latencies)

(4.8.2)
LW → %20 → 5 cycles
Pipelined → 350 x 5 = 1750 ps
Non-pipelined → 1250 ps

(4.8.3)
- Splitting the ID (instruction decode) part into the two new part 350/2 = 175 ps.
- The new clock cycle is 300 ps. (MEM ⇒ 300 ps)

(4.8.4)

lw = %20
sw = %15
data memory utilization → %35

(4.8.6)

lw completed in 5 cycles
sw completed in 4 cycles (excluding WB)
alu completed in 4 cycles (excluding MEM)
beq completed in 4 cycles (excluding WB)

Multi-cycle execution time → 5*0.20 + (0.45+0.20+0.15)*4
= 1.0 + 3.20 = 4.20

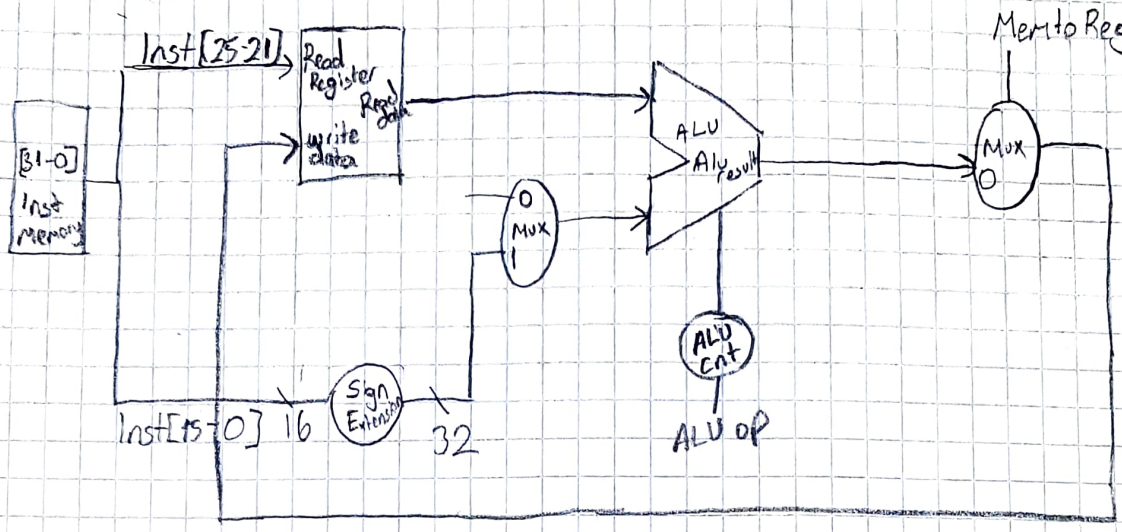
Single-cycle execution time → 1250/350 = 3.57

Enes Garip
150116034

3)

	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch
addi	0	1	0	1	0	0	0

ALUOp1	ALUOp2
1	0



4)