Problem 4.3 (all sub-parts)

(4.3.1)

Without improvement:

Clock cycle time = I-Mem + Regs + Mux + ALU + D-Mem + Mux

= 400 + 200 + 30 + 120 + 350 + 30

= 1130ps

With improvement:

Latency for ALU = 300 + 120 = 420ps

Clock cycle time = I-Mem + Regs + Mux + ALU + D-Mem + Mux

= 400 + 200 + 30 + 420 + 350 + 30

= 1430ps

(4.3.2)

5% fewer cycles are needed for a program

So, speedup = (1/0.95) * (1130/1430) = 0.83

(4.3.3)

Total cost = 1000*1 + 30*2 + 10*3 + 100*1 + 200*1 + 2000*1 + 500*1 = 3890

Speed up = 0.83

New cost = 3890 + 600 = 4490

Relative cost = new cost / total cost = 4490/3890 = 1.15

Cost-performance ratio = Relative cost / speed up = 1.15 / 0.83 = 1.39

Problem 4.4 (all sub-parts)

(4.4.1)

Same as I-Mem, 200ps

(4.4.2)

Clock cycle time = I-Mem + sign extend + shift left 2 + Add + Mux

= 200 + 15 + 10 + 70 + 20

= 315 ps

(4.4.3)

Clock cycle time = I-Mem + Register + Mux + ALU + Mux

= 200 + 90 + 20 + 90 + 20

= 420 ps

(4.4.4)

The graph represents datapath with all conditional and unconditional branches. Shift left 2 element is used to get the offset to compute the new PC. Therefore, PC-relative branches instructions require this source.

(4.4.5)

PC-relative unconditional branch instruction. Because the conditional branches are not required on the critical path and it is essential on PC-relative branches.

(4.4.6)

Latency 1= Regs + Muxs + ALU = 90 + 20 + 90 = 200 ps

Latency 2 = sign extend + shift left 2 + Add = 15 + 10 + 70 = 95 ps

The latency of the shift left 2 must be increased by 105ps or more than that of the latency provided for the sake of affecting the clock cycle time.

Problem 4.7 (all sub-parts)

(4.7.1)

sign extend:	0000 0000 0000 0000 0000 0000 0001 0100
shift left 2:	0001 1000 1000 0000 0000 0000 0101 0000

(4.7.2)

1010 1100 0110 0010 0000 0000 0001 0100

6 msb indicates "sw" instruction

6 lsb indicates the function code used to select the "sw" instruction

The function code of the instruction is one of the input of the ALU control unit

The second input of the ALU control unit is least two bits of the instruction which indicates the ALU opcode

(4.7.3)

The value of PC get incremented by 4 after each line of instruction been executed

Path: PC >> PC+4 >> Branch Mux >> Jump Mux >> PC

(4.7.4)

The output of the Mux that select either PC+4 or a branch address will be PC+4

For sw instruction, the value of RegDst could be 0 or 1. The output of the Mux for the Write register input can be Instruction[20-16] or Instruction[15-11]. So, the Write Register input is 00010 or 00000.

The Mux right before the ALU must select the sign extended version of Instruction[15-0]. This means that the output of the Mux in decimal is 20.

(4.7.5)

ALU = -3 and 20

Add (PC+4) = PC and 4

Add (branch) = PC+4 and 10*4

(4.7.6)

Value for read register 1 input is Instruction[25-21], value is 00011.

Value for read register 2 input in Instruction[20-16], value is 00010.

For sw instruction, the value of RegDst is 0 or 1. The output of the Mux for the Write register input can be Instruction[20-16] or Instruction[15-11]. So, the Write Register input is 00010 or 00000.

Because sw instruction will not write into any register. The value of the register write input is 0.

Problem 4.8 (sub-parts 4.8.1 - 4.8.5)

(4.8.1)
For pipelined processor: 350 ps
For non-pipelined processor: 350+250+150+200+300 = 1250 ps

(4.8.2)
latency of lw with pipelined processor: 5 * 350 = 1750
latency of lw with non-pipelined processor: 250+350+150+300+200 = 1250

(4.8.3)
Splitting the longest stage is most useful way to decrease the cycle time. So we should split the ID stage into two new stages. The new clock cycle time will be 300 ps.

(4.8.4)
Utilization of the data memory = LW +SW = 20 + 15 = 35%

(4.8.5)
Utilization of the write register = ALU + LW = 45 + 20 = 65%

(4.8.6)

Single - cycle → one clock cycle

pipeline → one instruction in every cycle

multi-cycle → LW 5,SW 4, ALU 4, BEQ 4

Multi: 5*0.2 + 4*0.8 = 4.2

Single: 1250/350 = 3.5