HW6 comp org

Ayush Krishnappa

November 2022

1 12.14.5

- a) Value of ALU control unit's input is 1, which indicates that register t5 should be used as the first operand instruction for ALU operation
  Value of ALU control unit's input ALUSrcB is 0, which means that the immediate value 20 should be used as the second operand for the ALU operation.
  Value of the ALU control unit's input ALUOp is 2, which means the ALU should perform the addition operation.

- b) The path through which the new PC address is formed is illustrated as such, processor fetches the instruction word 0xadac0014 from memory, the instruction is then decoded and the ALU control unit's inputs are determined, the instruction is then executed and the ALU performs addition operation, the PC is then incremented by 4 and the new PC address is 0x00000000.

- c) The value of the PC is 0x00000000 and the instruction is sw t4, 20(t5), the instruction will store the contents of register t4 into memory at the address specified by the sum of registers t5 and 20.
  Mux1's inputs are the instruction and the PC The instruction is 0xadac0014. The PC is 0x00000000.
  Output of Mux1 is 0xadac0014.
  Mux2's inputs are register t5 and value 20. The output of Mux2 is 20
  Mux3's inputs are register t5 and value 20. The output of Mux3 is t5
  Mux4's inputs are register zero and register t5. The output of Mux4 is t5.
  Mux5's inputs are register t4 and register t5. The output of Mux5 is t4.
  Mux6's inputs are register t4 and value 20. The output of Mux6 is 0x00000000

- d) When processor fetches the instruction 0xadac0014, it decodes the instruction first. The instruction is sw t4, 20(t5), which is a store word

instruction. This instruction stores the contents of register t4 into memory at the address specified by the sum of register t5 and the value 20. To execute this instruction, the processor first calculates the sum of register t5 and the value 20 using the ALU. This sum is used as the address at which to store the contents of register t4. The contents of register t4 are then input into one of the add units, which calculates the sum of the contents of register t4 and value 20. This sum is used as the data to be stored in memory at the specified address. Finally, the PC is incremented by 4 to fetch the next instruction.

- e) The values of all inputs for the registers unit are:
  RegDst, RegWrite, MemtoReg, MemRead, MemWrite.

2 12.14.7

- a) Register read(30) + Mem(250) + Register file (150) + Mux(25) + Mem(200) + Mux(25) + Register setup(20) = 700 ps

- b) Register read(30) + Mem(250) + Register file(150) + Mux(25) + ALU(200) + Mem(250) + Mux(25) + Register setup(20) = 950 ps

- c) Register read(30) + Mem(250) + Register file(150) + ALU(200) + Mux(25) + Mem(250) = 905 ps

- d) Register read(30) + Mem(250) + Register file(150) + Mux(25) + ALU(200) + Single gate(5) + Mux(25) + Register setup(20) = 705 ps

- e) Register read(30) + Mem(250) + Register file(150) + Mux(25) + ALu(200) + Mux(25) + Register setup(20) = 700 ps

- f) Min clock period for this CPU is 950 ps

3 12.14.10

- a) Clock cycle time without improvement = mem(250) + reg(150) + mux(25) + alu(200) + adder(250) + d-mem(5) + sg(30) + sign(20) + sg(50) + control(50) = 930 ps

Clock cycle time with improvement = mem(250) + reg(160) + max(25) + alu(200) + adder(150) + d-mem(5) + sg(30) + sign(20) + sg(50) + control(50) = 940 ps

Assuming 100 instructions before improvement, number of instructions after improvment are 96, speed up = (930/100) / (940/96) = 0.95, speed up = 0.95

- b) Cost without improvement = 1000 + 200 + 10 + 100 + 30 + 2000 + 5 + 100 + 1 + 500 = 3946, cost per unit clock cycle without improvement 3946/930 = 4.24

  Cost with improvement = 1000 + 400 + 10 + 100 + 30 + 2000 + 5 + 100 + 1 + 500 = 4146, 4146/940 = 4.38.

  Change in cost = 4.38 - 4.24 = 0.14, cost increased by 0.14.

- c) From the cost to performance ratio calculated above we can see that there is a clear trade off between cost and performance, Adding more registers causes cost to increase but performance to improve, thus the cost is being increased for the sake of performance. So in a situation where performance is the main priority cost can be compromised to accommodate for performance improvement. An application of this model could be adding more construction workers to increase the speed of development for an apartment complex. This would increase cost as well as performance. The other situation would be when cost is the main priority, meaning we would need to compromise performance for cost. An application of this model would be a customer who needs to make improvements to their home but they have a budget. Here speed would be compromised for cost.

4 12.14.16

- a) Pipelining: all stages take a single clock cycle so, clock cycle must be long enough to accommodate the slowest operation. Therefore, the clock cycle is determined by the slowest stage Instruction decode = 350 ps

  Non-pipelining: each instruction goes through all stages, so clock cycle is determined by sum. 250 + 350 + 150 + 300 + 200 = 1250 ps

- b) Pipeline: 5 * 350 = 1750 ps

Non-pipelining: sum of stages 250 + 350 + 150 + 300 + 200 = 1250 ps

- c) Splitting the longest stage is the best way to reduce cycle time, after splitting the new cycle time is based on the new longest stage, in this case it would go from ID(350) to CT(300)

- d) Data memory is utilized only by load and store word instructions, given the distribution of the LW instruction on the processor 20 percent and the SW instruction on the processor is 15 percent, utilization of data memory is 20 + 15 = 35 percent of the clock cycles.

- e) Write registers may be utlized by ALU and LW instructions, given LW: 20 and SW: 45, utilization of write register port is 20 + 45 = 65 percent of clock cycles