2023-1 Computer Architecture Homework #3

Due: 5/31 (Wed) 11:59 p.m.

4.7 [30] Problems in this exercise assume that the logic blocks used to implement a processor's datapath have the following latencies:

I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign Extend	Control
250ps	150ps	25ps	180ps	140ps	5ps	30ps	30ps	60ps	50ps

"Register read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and Register File. Consider the datapath presented in Figure 4.17.

4.7.1 [5] <§4.4> What is the latency of an R-type instruction?

4.7.2 [5] <§4.4> What is the latency of `lw`?

4.7.3 [5] <§4.4> What is the latency of `sw`?

4.7.4 [5] <§4.4> What is the latency of `beq`?

4.7.5 [5] <§4.4> What is the latency of an arithmetic, logical, or shift I-type (non-load) instruction?

4.7.6 [5] <§4.4> What is the minimum clock period for this CPU?

4.8 [20] Consider the following instruction mix:

R-type	I-type (non-lw)	lw	sw	beq
25%	27%	25%	11%	12%

4.8.1 [5] <§4.4> What fraction of all instructions use data memory?

4.8.2 [5] <§4.4> What fraction of all instructions use instruction memory?

4.8.3 [5] <§4.4> What fraction of all instructions use the sign extend?

4.8.4 [5] <§4.4> Suppose you could build a CPU where the clock cycle time was different for each instruction. What would the speedup of this new CPU be over the CPU presented in Figure 4.17? Use the results from Exercise 4.7.

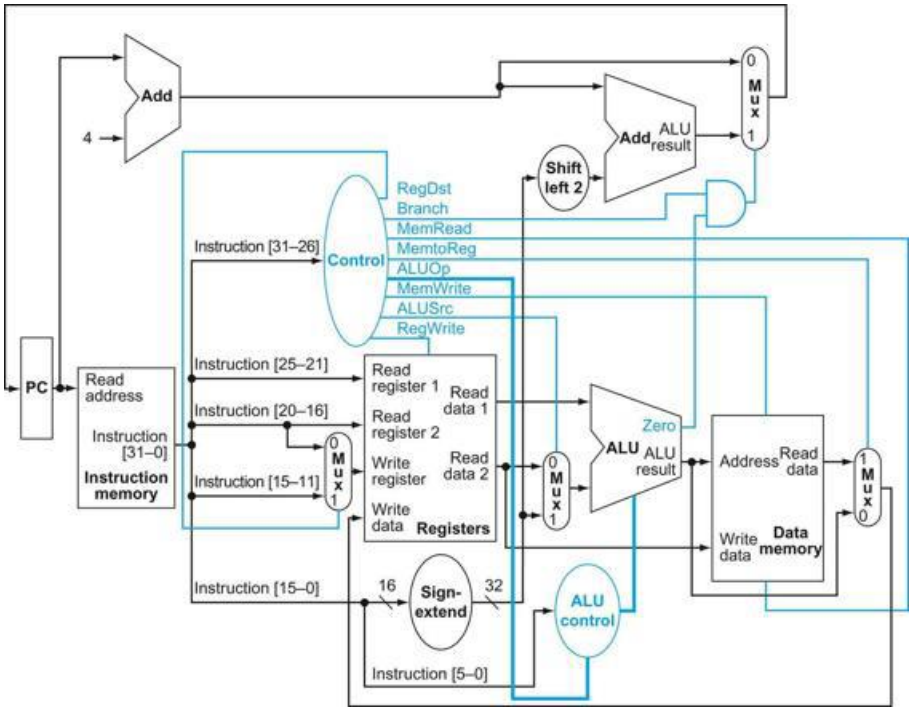

FIGURE 4.17 The simple datapath with the control unit.

The input to the control unit is the 6-bit opcode field from the instruction. The outputs of the control unit consist of three 1-bit signals that are used to control multiplexors (RegDst, ALUSrc, and MemtoReg), three signals for controlling reads and writes in the register file and data memory (RegWrite, MemRead, and MemWrite), a 1-bit signal used in determining whether to possibly branch (Branch), and a 2-bit control signal for the ALU (ALUOp). An AND gate is used to combine the branch control signal and the Zero output from the ALU; the AND gate output controls the selection of the next PC. Notice that PCSrc is now a derived signal, rather than one coming directly from the control unit. Thus, we drop the signal name in subsequent figures.

4.16 [20] In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	400ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	Load	Store
45%	15%	25%	15%

4.16.1 [5] <§4.6> What is the clock cycle time in a pipelined and non-pipelined processor?

4.16.2 [5] <§4.6> What is the total latency of an `lw` instruction in a pipelined and non-pipelined processor?

4.16.3 [5] <§4.6> If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

4.16.4 [5] <§4.6> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

4.25 [40] Consider the following loop.

```
LOOP: lw $s0, 0($s3)
      lw $s1, 4($s3)
      add $s2, $s0, $s1
      sw $s2, 0($s4)
      addi $s3, $s3, -8
      addi $s4, $s4, 4
      bnez $s2, LOOP
```

4.25.1 [10] <§4.8> If there is no forwarding or hazard detection, insert `NOPs` to ensure correct execution.

4.25.2 [10] <§4.8> Now, change and/or rearrange the code to minimize the number of `NOPs` needed. You can assume register `$t0` can be used to hold temporary values in your modified code.

4.25.3 [20] <§4.8> Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, that the pipeline has full forwarding support, and that branches are resolved in the EX (as opposed to the ID) stage. Show a pipeline execution diagram for the first two iterations of this loop.