ECE 485/585 – Computer Organization and Design

HOMEWORK #4

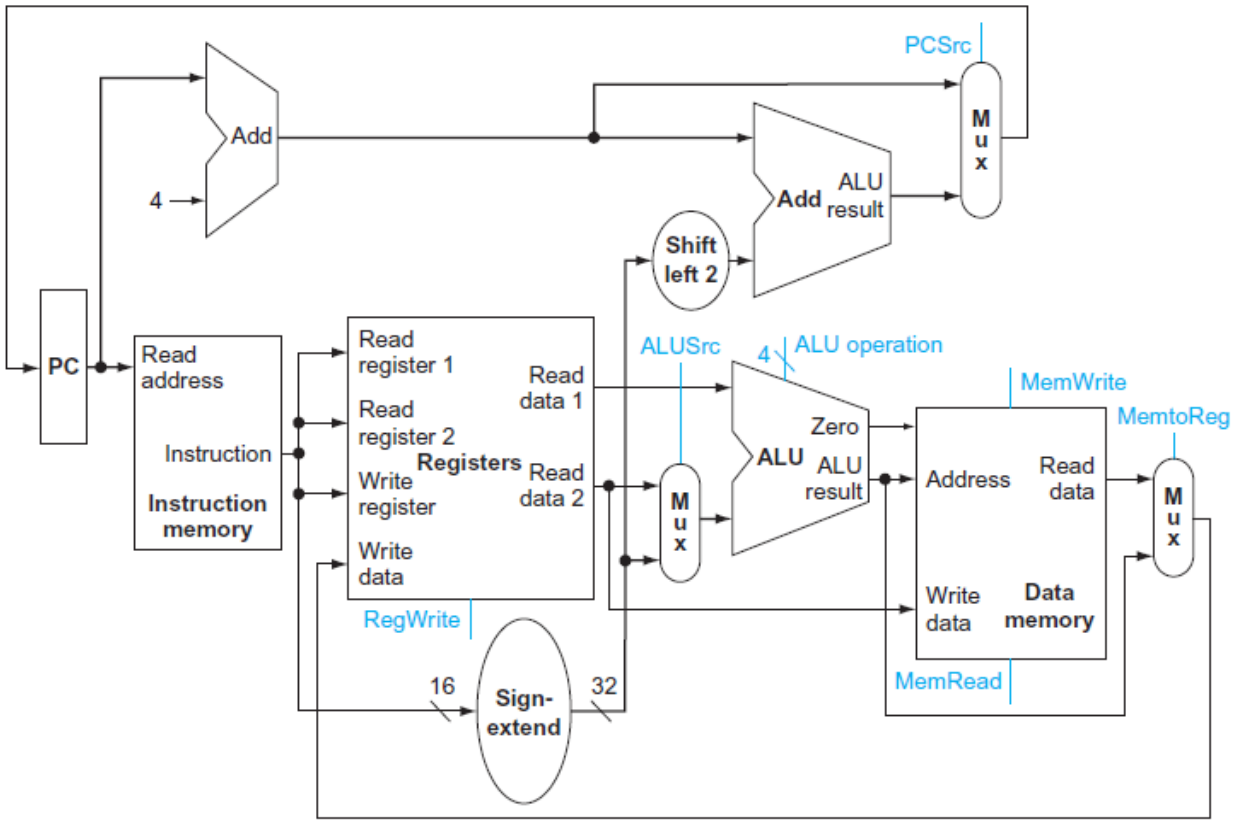
Due date: Friday November 18th 2022 11:59PM

Solve the following exercises. <u>You must show your work to get full points.</u>

Question 1. Assume the following computer system components within the processor's datapath have the following latencies:

Inst. Mem	Add	MUX	ALU	Reg File	Data Mem	Sign-Ext	Shifter
300 ps	50 ps	15 ps	100 ps	85 ps	400 ps	10 ps	5 ps

Also, consider the below figure.



a. What would be the total execution latency for an instruction *sw $t1, 20($t0)*? (5 Points)

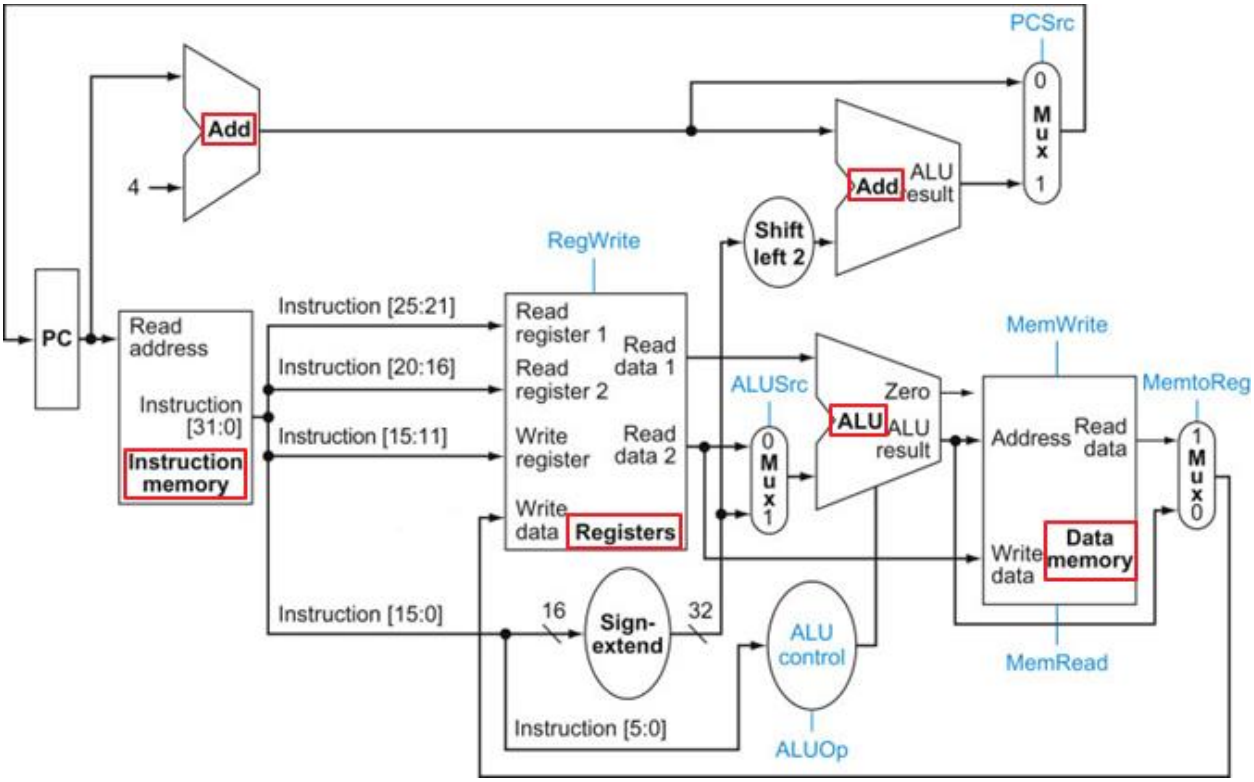
b. What would be the total execution latency for an instruction *beq $t1, $t0, LABEL*? (5 Points)

c. Show the critical path for all MIPS instruction set types. (5 Points)

d. Consider pipeline features to be added to the above figure. What would be the desired clock rate of the system to execute in pipeline mode. Prove your answer by showing your work leading to the desired clock rate. (You may discard latencies in the temporary registers in between stages) (10 Points)

ECE 485/585 – Computer Organization and Design

Question 2. You wish to run the **sw** instruction using the **single-cycle architecture** shown below.



a) List any HW modification required for **sw** instruction along with datapath and control signal (5 Points):

i._____

ii._____

iii._____

iv._____

b) Explain the operation at each component (marked red above) in RTL representation and all control signal values (indicated in blue above) for the **sw** instruction (10 Points):

Instruction Memory:

Registers:

ALU:

Data Memory:

Adders (both):

ECE 485/585 – Computer Organization and Design

Question 3. Assume there are no pipeline stalls and the breakdown of executed instructions is:

add	addi	not	beq	lw	sw
5%	15%	15%	35%	15%	15%

a. In what fraction of all cycles is the data memory used? (5 Points)

b. In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its input is not needed? (5 Points)

Question 4. Exercise 4.8 from Textbook (10 Points)

4.8 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	300ps	200ps

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

alu	beq	lw	sw
45%	20%	20%	15%

4.8.1 [5] <§4.5> What is the clock cycle time in a pipelined and non-pipelined processor?

4.8.2 [10] <§4.5> What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

4.8.3 [10] <§4.5> 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?

4.8.4 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the data memory?

4.8.5 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

4.8.6 [30] <§4.5> Instead of a single-cycle organization, we can use a multi-cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare clock cycle times and execution times with single-cycle, multi-cycle, and pipelined organization.

ECE 485/585 – Computer Organization and Design

Question 5. Exercise 4.7 from Textbook (10 Points)

4.7 In this exercise we examine in detail how an instruction is executed in a single-cycle datapath. Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction word:

10101100011000100000000000010100.

Assume that data memory is all zeros and that the processor's registers have the following values at the beginning of the cycle in which the above instruction word is fetched:

r0	r1	r2	r3	r4	r5	r6	r8	r12	r31
0	–1	2	–3	–4	10	6	8	2	–16

4.7.1 [5] <§4.4> What are the outputs of the sign-extend and the jump "Shift left 2" unit (near the top of Figure 4.24) for this instruction word?

4.7.2 [10] <§4.4> What are the values of the ALU control unit's inputs for this instruction?

4.7.3 [10] <§4.4> What is the new PC address after this instruction is executed? Highlight the path through which this value is determined.

4.7.4 [10] <§4.4> For each Mux, show the values of its data output during the execution of this instruction and these register values.

4.7.5 [10] <§4.4> For the ALU and the two add units, what are their data input values?

4.7.6 [10] <§4.4> What are the values of all inputs for the "Registers" unit?

(Figure 4.24 can be found below)

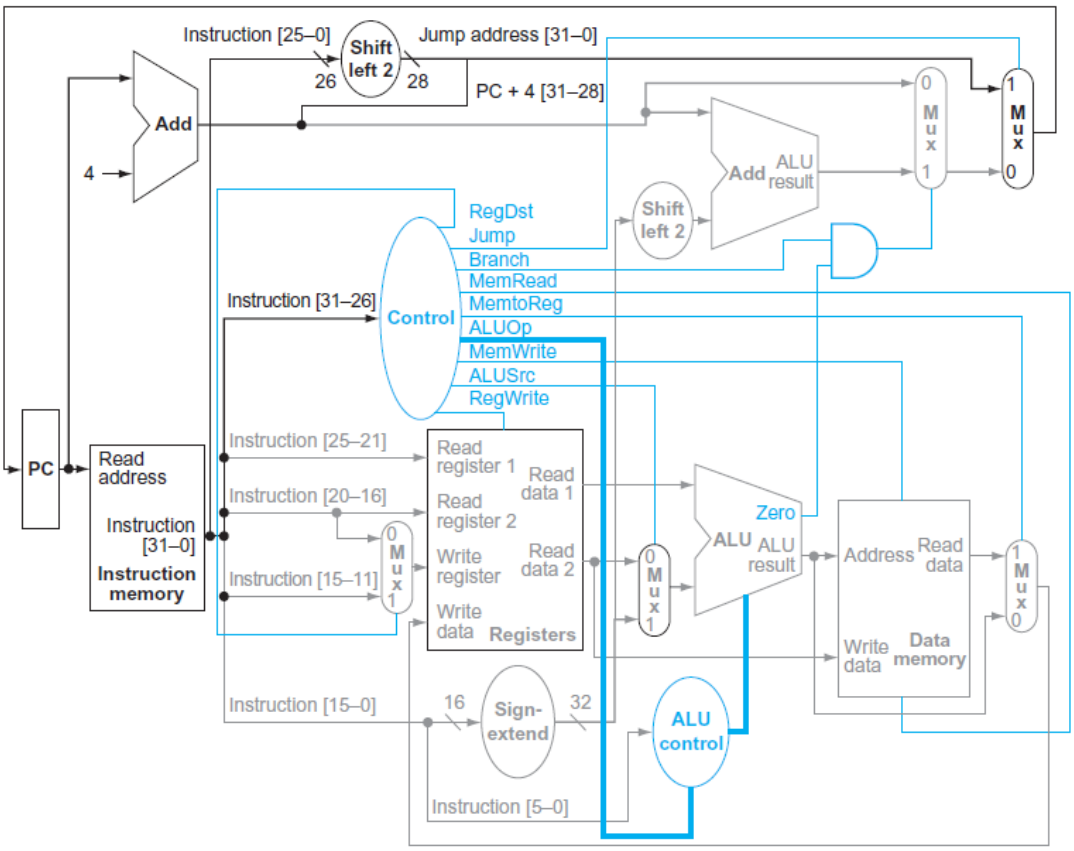


FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump control signal. The jump target address is obtained by shifting the lower 26 bits of the jump instruction left 2 bits, effectively adding 00 as the low-order bits, and then concatenating the upper 4 bits of PC + 4 as the high-order bits, thus yielding a 32-bit address.