Name: _____

Single Cycle Implementation Homework

Only problems 1 through 4 are due for credit.

1. Disassemble the following machine instructions:

   (a) `0x016a6020`

   (b) `0x014b6824`

   (c) `0x35134321`

   (d) `0x20000000`

   (e) `0x000a5140`

   (f) `0x11cffffa`

   (g) `0x8e0b000c`

   (h) `0x8e4efff8`

   (i) `0xae2d0010`

   (j) `0x08100000`

Name: _____

2. For each instruction below, determine the value at each of the labeled points in Figure 1. Assume that register x has the value $1000 * x + 10 * x$. For example, assume `$0` is 0, `$1` is 1010, `$2` is 2020, `$31` is 31310, etc. In addition: Assume the current instruction is at address `0x0040001C`, label `fwd6` is `0x00400034`, and `bwd6` is `0x00400004`. Represent the contents of memory location x as `M[x]`. Some values (e.g., the output of the ALU when doing a jump) are undefined. List all well-defined values — even if that value isn't used by the instruction.

Point	`add $12, $3, $2`	`addi $3, $2, 131`	`addi $3, $2, -16`	`lw $5, 0x5ABC($24)`
A	`0x0040 0020`			
B	`3`			
C	`2`			
D	`12`			
E	`12`			
F	`24608 --- 0x6020`			
G	`2020 --- 0x7E4`			
H	`0x00006020`			
I	`3030 --- 0xBD6`			
J	`2020 --- 7E4`			
K	`5050 --- 0x13BA`			
L	`0`			
M	`5050 --- 0x13BA`			

Point	`sw $4, -8($2)`	`beq $3, $2, fwd6`	`beq $3, $3, bwd6`	`j fwd6`
A				
B				
C				
D				
E				
F				
G				
H				
I				
J				
K				
L				
M				

Name: _____

3. Suppose the `branch` control wire is set to 1 whenever the op code is 2 (for `j`). Will the CPU operate correctly? If so, explain why. If not, give an example instruction that will not operate correctly.

4. Suppose your CPU has a hardware bug and places 1 on the `regWrite` control wire for `sw`. Describe the effects of this bug when running when running `sw $t5, -4($a1)`.

5. What is the purpose of the ALU labeled "A" in Figure 2?

6. Why does the input to the ALU labeled "A" in Figure 2 have the constant value 4?

7. What is the purpose of the "shift-left 2" labeled "B" in Figure 2?

8. Why are jump targets encoded as a word address instead of a byte address?

9. What is the purpose of the sign extender in Figure 2? Why is it necessary?

10. What is the purpose of the mux labeled "C" in Figure 2?

11. What is the purpose of the "shift-left 2" labeled "D"' in Figure 2?

12. What is the purpose of the mux labeled "F" in Figure 2?

13. What is the purpose of the and gate labeled "G" in Figure 2?

14. What is the collective purpose of the muxes labeled "H" and "I" in Figure 2?

15. Does it matter whether mux "H" comes first or mux "I"?

16. What is the purpose of the mux labeled "J" in Figure 2?

17. Add components to Figure 2 to show how to add a "branch if not equal" instruction to the single-cycle CPU.

Name: _____

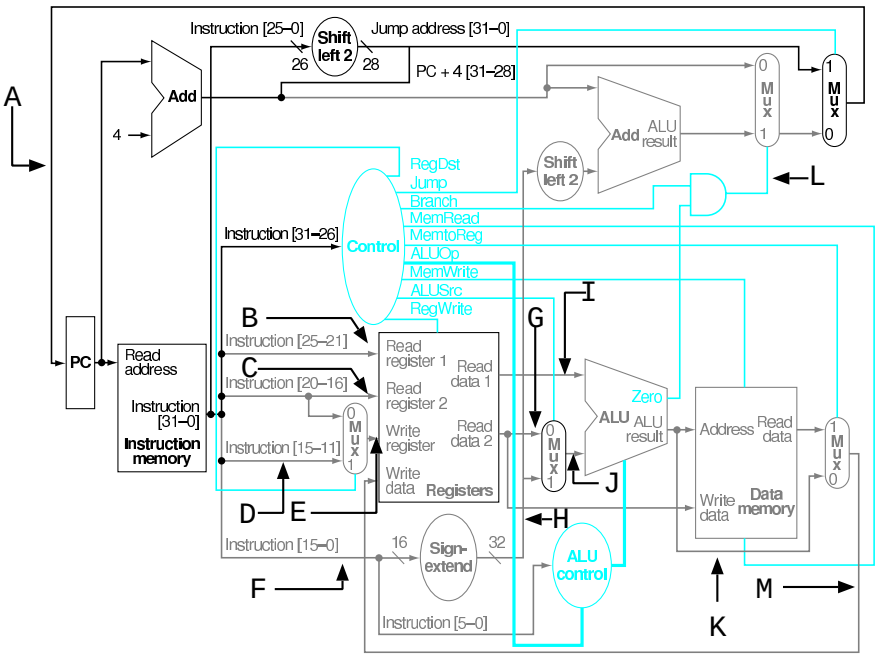

FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. *An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump 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. Copyright © 2009 Elsevier, Inc. All rights reserved.*

Figure 1: Single Cycle CPU with labeled points

Name: _____

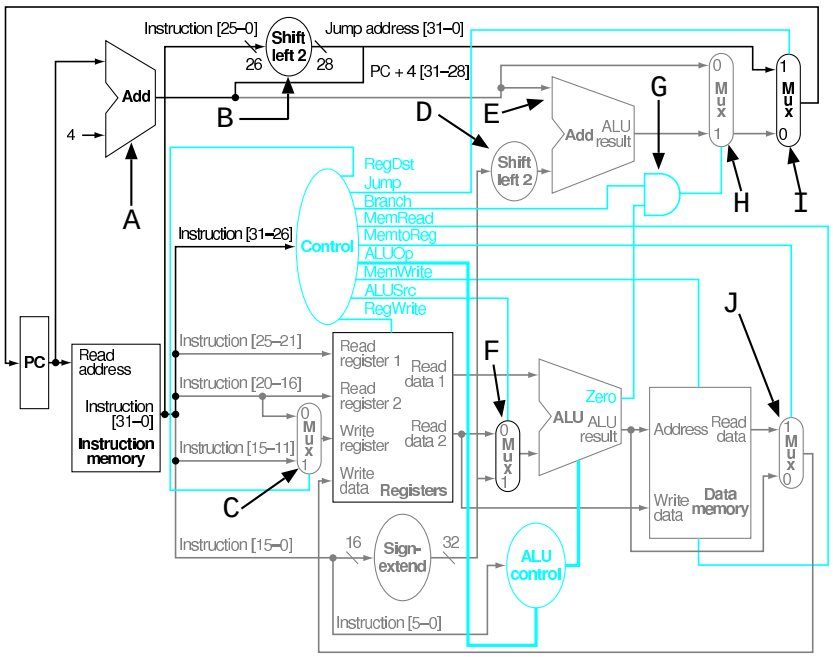

FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. *An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump 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. Copyright © 2009 Elsevier, Inc. All rights reserved.*

Figure 2: Single Cycle CPU with labeled parts

Name: _____

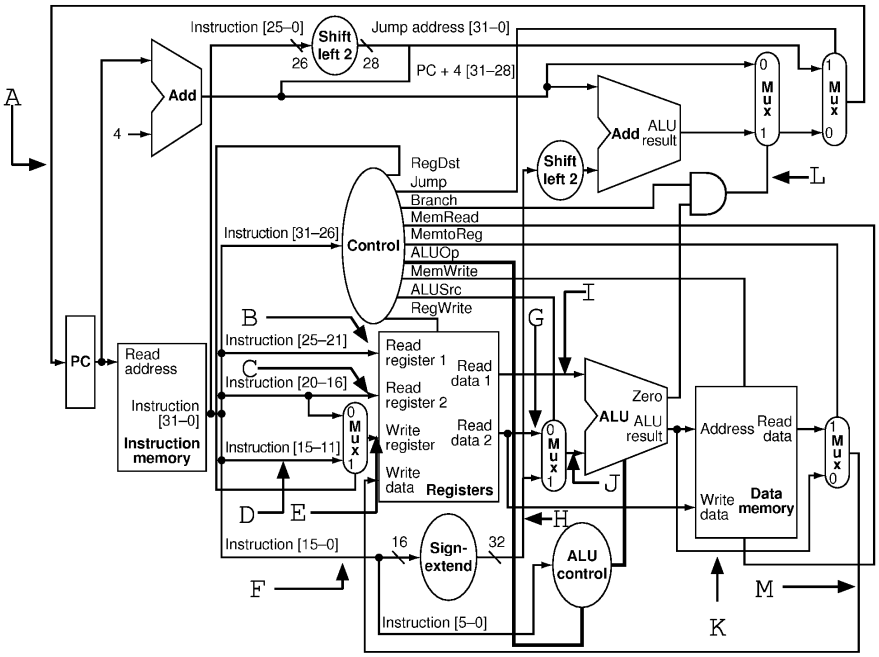

FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. *An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump 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.*

Figure 3: Single Cycle CPU with labeled points (black & white)

Name: _____

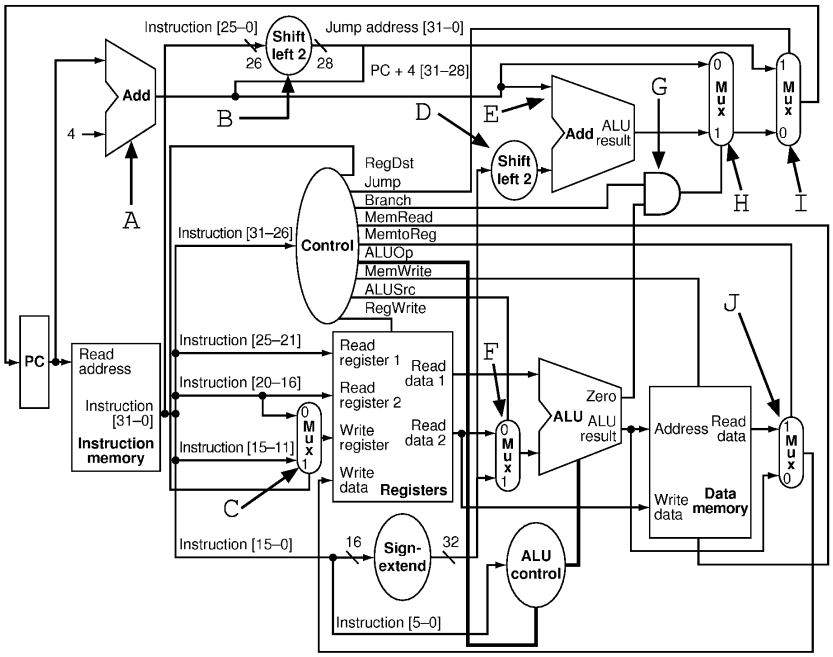

FIGURE 4.24 The simple control and datapath are extended to handle the jump instruction. *An additional multiplexor (at the upper right) is used to choose between the jump target and either the branch target or the sequential instruction following this one. This multiplexor is controlled by the jump 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.*

Figure 4: Single Cycle CPU with labeled parts (black & white)