

JON CRAWFORD

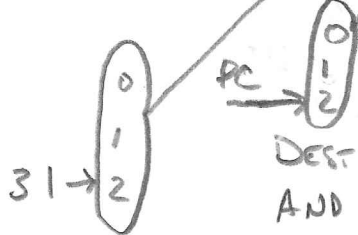
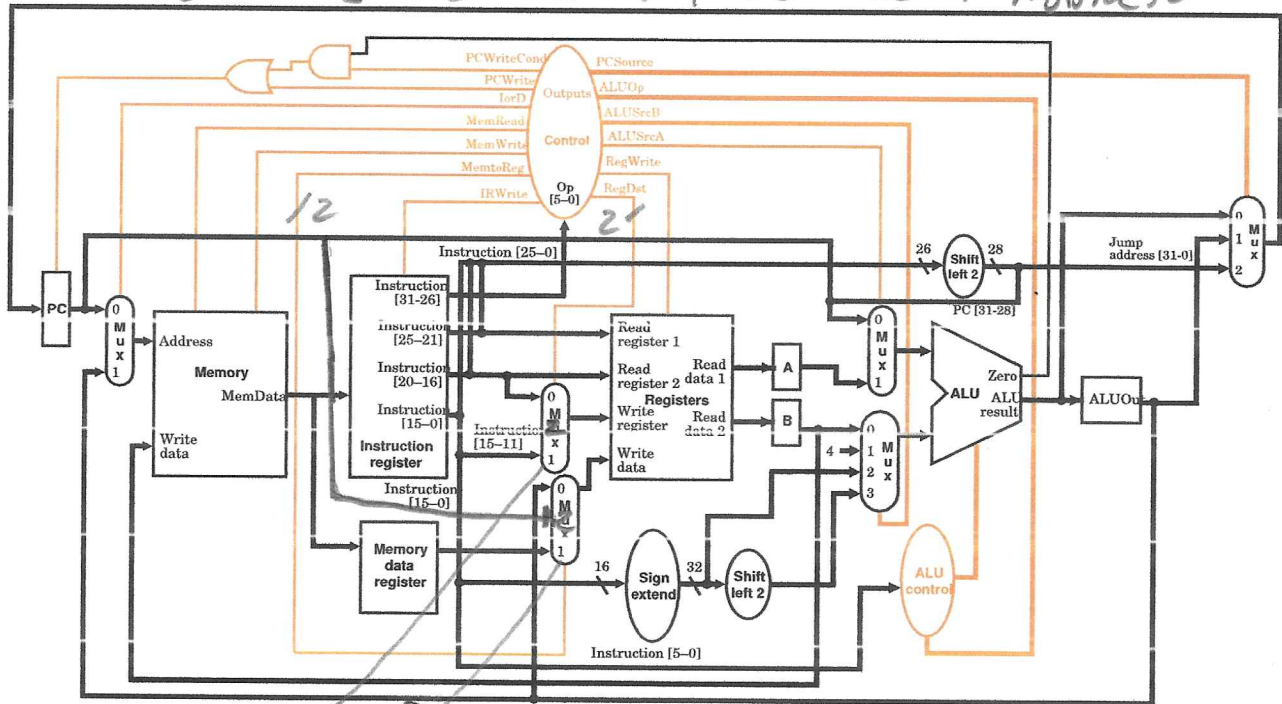
CDA3101
Assignment 4
Multicycle Datapath and Control

Objectives. Learn how to enhance the datapath and control for a multicycle implementation of a simple ALU.

Put all your answers on these two sheets.

1. Modify the datapath and control for the multicycle implementation to add the *jal* (jump and link) instruction. Remember that the *jal* instruction is like a *j* (jump) instruction, but it also places the address of the instruction following the *jal* instruction in register \$31 as a return address. Examine the figures below and on the following page showing the datapath and the finite state machine (FSM) control for the multicycle implementation, respectively. Make the appropriate modifications to both the datapath and the FSM control figures to support the *jal* instruction. (100 points)

GOAL: $[\$ra] = PC + 4$, $PC = \text{JUMP ADDRESS}$



TO GET \$ra (31) INTO THE WRITE DESTINATION REGISTER EXPAND THE MUX AND HARDCODE 31 AS AN INPUT. SET REGDEST TO 10 (2) TO PICK IT. ALSO, EXPAND THE MUX FOR WRITE DATA INPUT TO ACCEPT PC AS AN INPUT AND THEN CONNECT PC TO IT. SET MEMTO REG THAT CONNECTS TO THAT INPUT TO 10 (2) TO PICK IT. PC WRITE AND PC SOURCE REMAIN THE SAME AS J INSTRUCTION TO COMPLETE JUMP. ALSO SET REGWRITE TO 1 SO THAT THE ADDRESS GETS WRITTEN.

0: INSTRUCTION FETCH
 1: INSTRUCTION DECODE
 10: [31] = PC + 4, PC = JUMP ADDRESS

