Your Name: _____ netid:_____ Group #:
Name: _____ netid:_____
Name: _____ netid:_____
Name: _____ netid:_____

ECE 120 Worksheet 15: LC-3 Control Unit

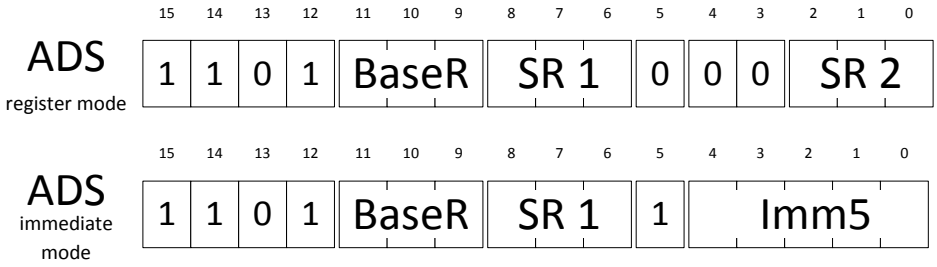
In this discussion, you will work with the LC-3 control unit defined in the appendix of Patt & Patel. In particular, you will extend the LC-3 ISA by introducing a new instruction, and will modify the microsequencer in order to extend the ISA.

The problems presented in this discussion are taken from final exams from prior semesters.

You may detach and keep the last two pages of this discussion booklet.

1. Extending the LC-3 ISA

You are charged with adding a new instruction, called **ADS** (add and store), to the LC-3 instruction set. This instruction adds two values, just like the ADD instruction, and stores the result to **memory** instead of the register file. The destination memory address is provided in the BaseR register specified by IR[11:9] bits (so M[BaseR] ← sum). The binary encoding of this instruction is:



a) In RTL form, give a sequence of (at most four) *microinstructions* that implement the execute phase of the **ADS** instruction. Make sure that your implementation **does not modify any values in the general-purpose register file** and **does not set condition codes**.

_____

_____

_____

_____

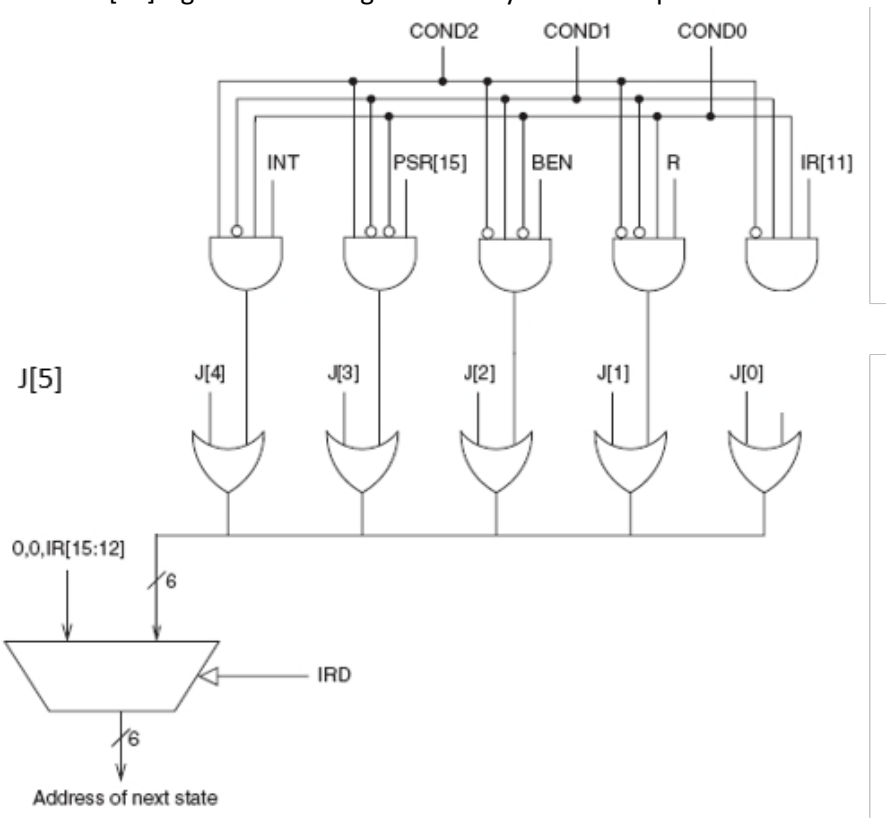
b) Determine the control ROM microinstructions that implement the RTL statements from part (a). Complete the table below by filling in 0, 1, or X as appropriate. Use don't cares wherever possible. Specify ROM addresses in **decimal**. Note that your first state number is implied by the decode strategy used in the P&P microarchitecture. Be sure to reuse the memory write state used by all other store instructions in the design (see the state diagram attached to the back of this document). If you need additional states, state numbers **55**, **56**, **57**, and **58** are available for your use.

ROM address	IRD	COND(3)	J(6)	LD.BEN LD.MAR LD.MDR LD.IR LD.PC LD.REG LC.CC	GateMARMUX GateMDR GateALU GatePC	MARMUX PCMUX(2) ADDR1MUX ADDR2MUX(2) DRMUX(2) SR1MUX(2) ALUK(2)	MIO.EN R.W

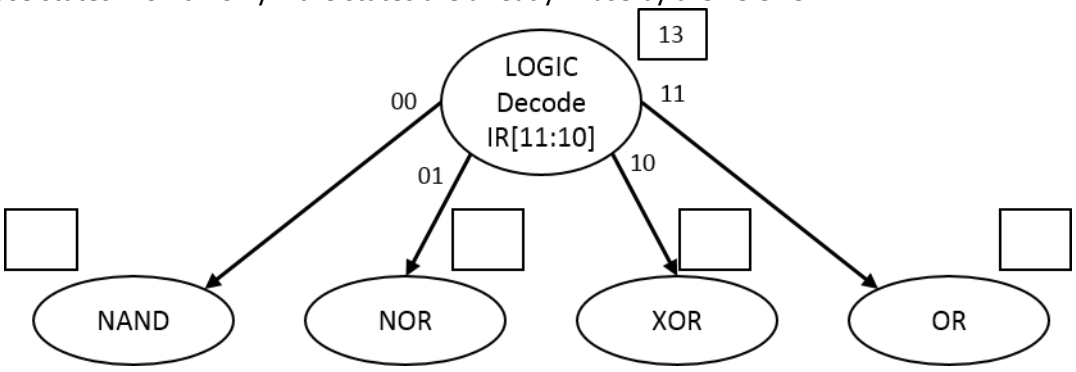
2. Modifying the LC-3 Microsequencer

Add a new instruction, **LOGIC**, with opcode 1101, to the LC-3. The **LOGIC** instruction performs one of four logic operations (NAND, NOR, XOR, and OR) based on the IR[11:10] bits. To implement this new instruction, we need five new states in the LC-3 FSM and need to modify the microsequencer circuit. The first state performs a secondary decode phase before executing the four logic operations.

a) Use COND = 110 to determine the next state during the secondary decode phase of the logic operation. Adding at most four gates, modify the microsequencer circuit so that it can correctly decode the **LOGIC** instruction. A few modifications have already been made to give you a hint. Note that the INT and PSR[15] signals in the diagram are beyond the scope of our course.



b) Based on your microsequencer circuit above, choose a set of *viable* next states for the execute phase states. Don't worry if the states are already in use by the LC-3 FSM.

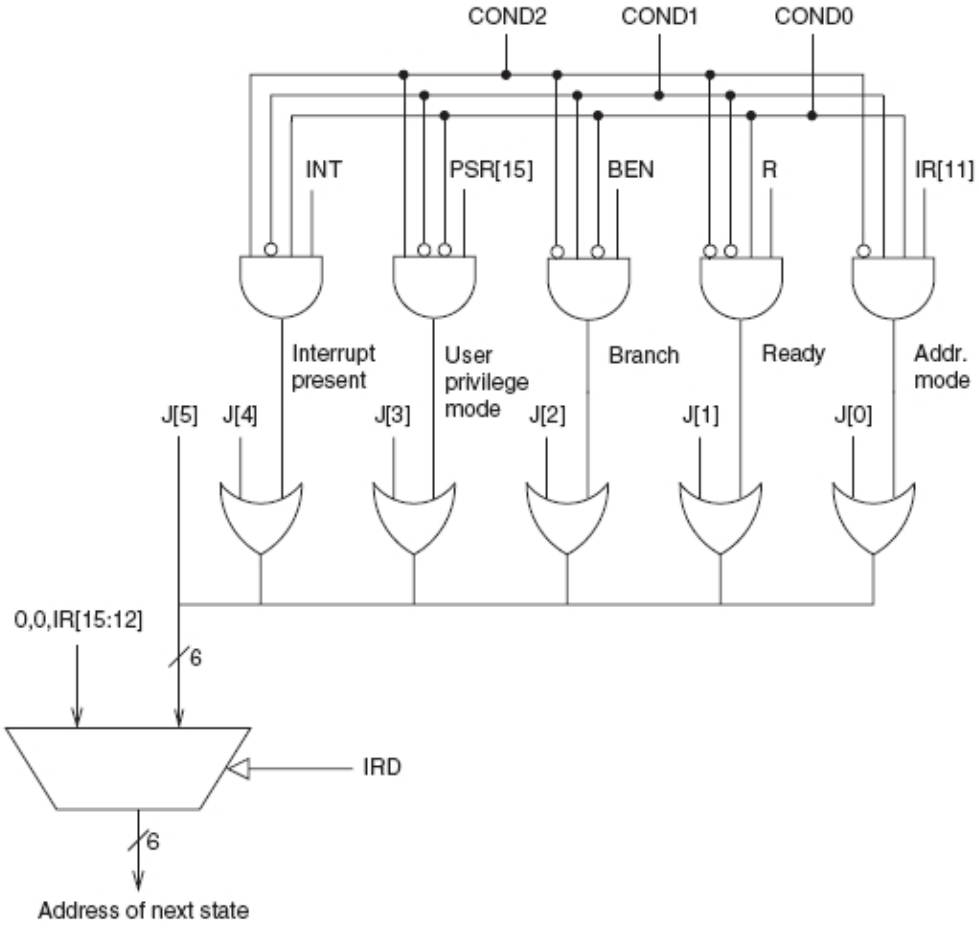


The LC-3 Microsequencer from Patt & Patel

$$\text{CAR} \leftarrow \begin{cases} 00||\text{opcode} & \text{if IRD} = 1 \text{ (during decode)} \\ \text{J OR } 1 & \text{if IRD} = 0 \text{ AND COND} = 011 \text{ AND IR}[11] = 1 \\ \text{J OR } 2 & \text{if IRD} = 0 \text{ AND COND} = 001 \text{ AND R} = 1 \\ \text{J OR } 4 & \text{if IRD} = 0 \text{ and COND} = 010 \text{ AND BEN} = 1 \\ \text{J} & \text{otherwise} \end{cases}$$

where IRD indicates that the current state is the decode state, J is the 6-bit next state address contained in the current state's control word, COND is a condition contained in the current state's control word, R is the memory ready signal, and BEN is the branch enable signal.

Again, INT and PSR[15] are outside the scope of our class, so we have left them out of the equation above.



The LC-3 FSM from Patt & Patel