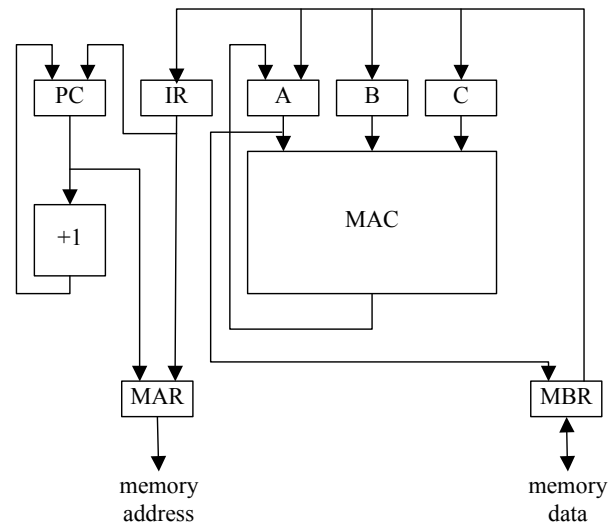


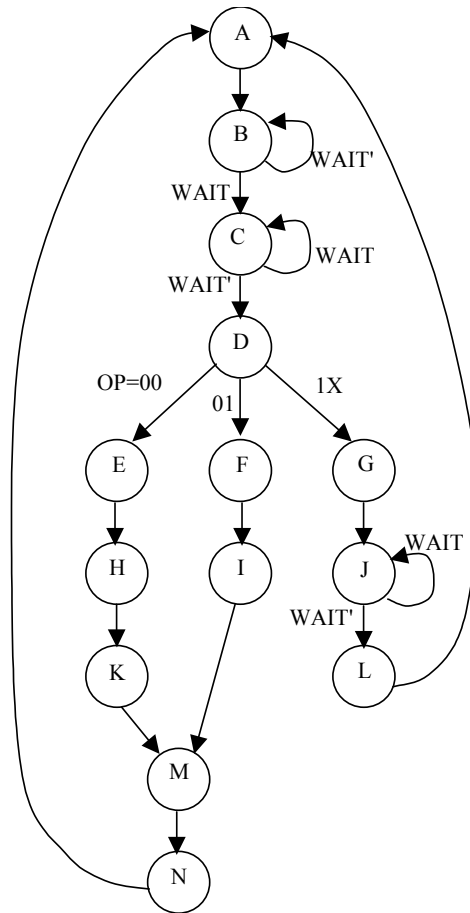
Homework #8

Due: Friday, November 30, 2001

- 1) Problem 11.2 on the book.
- 2) Given four registers (R1, R2, R3 and R4), and an additional signal A, you are required to design the interconnection between them so that the resulting datapath can support the operations described in the following cases:
 - a) transfer the value of any register or A into any register, and being able to do several transfers at the same time.
 - b) transfer the value of any register or A into any register, but only one transfer at a time.
 - c) divide the registers into two groups: group 1 contains R1 and R2, and group 2 contains R3 and R4; you should be able to transfer one register of one group or A into one or both registers of the other group, and being able to do two inter-group transfers at the same time.
- 3) Given the datapath shown in section 9, page 1-20 of the lecture notes, describe in register transfer notation the execution of the following instructions:
 - a) $rt = 55h$ (Hint: the constant 55h is located in the memory right after the opcode)
 - b) $rt = mem(10h)$ (Hint: the constant 10h is located in the memory right after the opcode)
- 4) The figure shows the datapath of a simple Digital Signal Processor. The MAC unit performs the multiply-accumulate operation $(A+B*C)$. The registers have a load control signal (PCld, IRld, Ald, Bld, Cld, MARld) and those with two inputs have a select signal (PCsel, Asel, MARsel and MBRsel). Additionally, MBR can enable a buffer to let the data flow from the memory to the registers by using the signal MBRen.
 - a) Describe in register transfer notation how would you implement the instruction $A=n$, where n is a constant value. (Include the opcode fetch).
 - b) Which of the following instructions you can't implement with this datapath?
 - i) $B=mem(n)$
 - ii) $A=A+B*C$
 - iii) $B=C$
 - iv) jump n
 - v) $A=B+C$
 - vi) jump A
 - c) Give a sequence of valid instructions to perform $Y=X^2$, where X and Y are variables located in $mem(10h)$ and $mem(11h)$ respectively.



- 5) Given the state diagram of the figure, design an implementation of the FSM using the jump counter method. Your design should be complete, including state assignment, minimized Boolean expressions for Clear, Load and Count, jump state logic (if you use a ROM, give the contents) and schematic.



- 6) Using the datapath of Problem 4, identify all the microoperations. Write the opcode fetch implementation using a horizontal microcode format.
- 7) A processor has 3 general purpose registers (A,B,C) and is able to perform the following operations: load/store value from/to memory to/from any register, add A+B and store result in any register, subtract A-B and store result in any register, shift right/left any register, jump to any memory location unconditionally and conditionally on zero or sign of the result.
- Draw a datapath for this processor (you can use an ALU and a Shifter).
 - Select a coding for the instructions, grouping the bits in a convenient way.
 - List the microoperations implied by the datapath.
 - Write a sequence of microoperations to perform a conditional jump on the sign of the result from the ALU. The jump instruction mnemonic is JP N,label and means: if the sign is negative, jump to the address 'label'.