Department of Electrical and Computer Engineering
The University of Texas at Austin

EE 360N, Fall 2003
Yale Patt, Instructor
Santhosh Srinath, Danny Lynch, TAs
Exam 2, November 19, 2003

Name:__________________________

Problem 1 (20 points):_______

Problem 2 (20 points):_______

Problem 3 (20 points):_______

Problem 4 (20 points):_______

Problem 5 (20 points):_______

Total (100 points):______________

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is recorded on each sheet of the exam.

GOOD LUCK!

Name:__________________________

Problem 1 (20 points):

Part a (5 points): Is a scoreboard a useful structure in an in-order pipelined processor? Why or why not? Explain in 20 words or fewer, please.

Part b (5 points): A microprocessor manufacturer decides to advertise its newest chip based only on the metric IPC (Instructions per cycle). Is this a good metric? Why or why not?

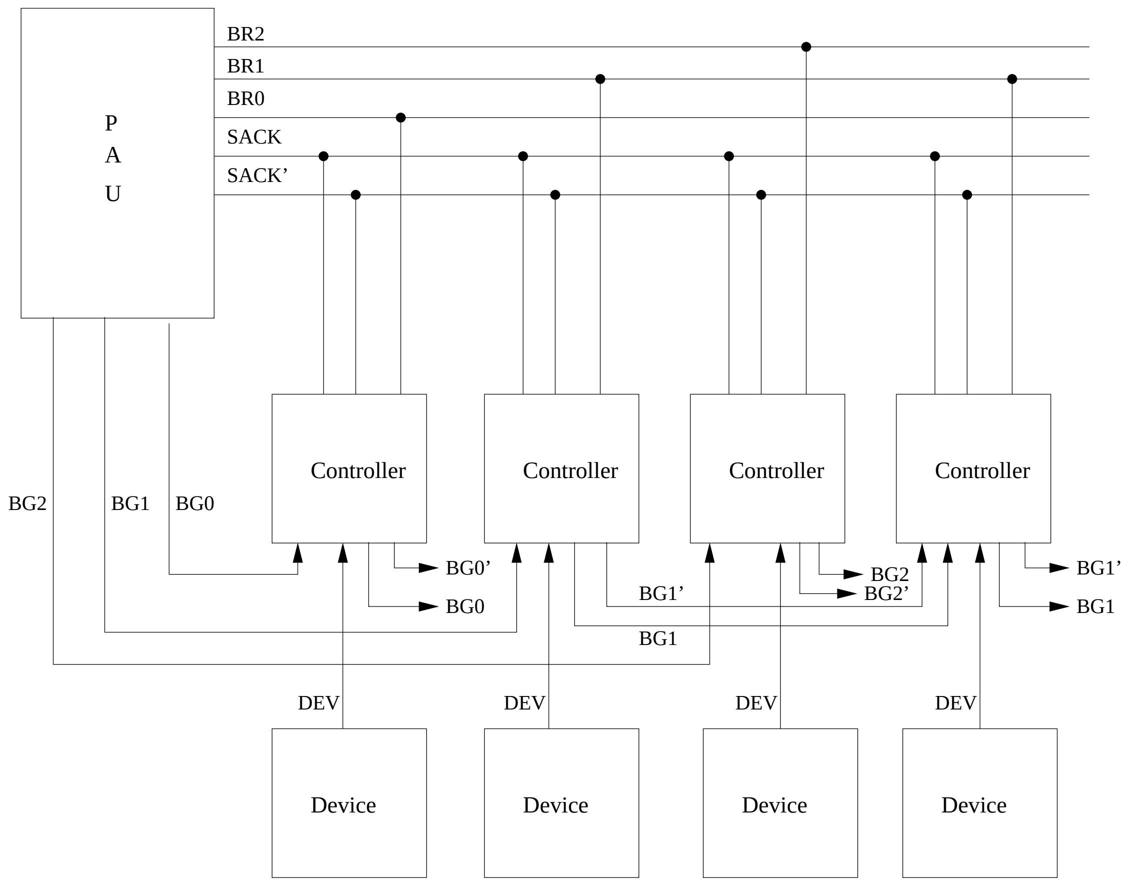
If you were the chief microarchitect for another company and were asked to design a chip to compete based on this metric, what important design decision would you make (in less than 10 words)?

Part c (10 points): A residue number system has been designed with three moduli, p1, p2, and p3. The number 17 is represented as 261. The number 25 is represented as 039. What are the three moduli?

Name:_________________________

Problem 2 (20 points):

Recall the asynchronous bus studied in class. You have been asked to redesign the device controller to allow the PAU (priority arbitration unit) to save time by assigning the bus masters for the next TWO bus cycles (instead of one) at the same time. The connections of device controllers to the PAU and to each other are shown below.



The system will work as follows: If two or more devices request the bus on the same (and highest of those asserted) priority, then the PAU bus grant signal will set up the bus masters for the next two bus cycles. We will need two SACK signals, SACK and SACK'. We will need two bus grant signals for each level of priority, for example BG1, and BG1' shown on the figure above.

If a device controller on priority level i receives BGi and wants the bus, it asserts SACK and passes on BGi'. If it does not want the bus, it passes on BGi. If a device controller receives BGi' and wants the bus, it asserts SACK' and passes on nothing. If it does not want the bus, it passes on BGi'.

When BBSY is negated, the device controller that is asserting SACK asserts BBSY, and negates SACK. When the device controller that is asserting SACK' sees that SACK has been negated, it asserts SACK and negates SACK'.

When SACK and SACK' are both negated, the PAU can again grant the bus.

Your job: On the next page construct the state machine for the device controller on the next page. Use the Moore model, as we did in class. Show only relevant inputs and outputs on all arcs. Our solution requires 8 states.

Name:__________________________

Problem 2 (continued):

Name:__________________________

Problem 3 (20 points):

Six instructions, three adds (ADD Ri,Rj,Rk) and three multiplies (MUL Ri,Rj,Rk) are fetched one at a time, on successive cycles, decoded, and executed when their operands become available (i.e., out-of-order). Rj and Rk are the source registers, Ri is the destination register.

The stages for each instruction are as discussed in class. For the ADD, 7 cycles: F-D-E1-E2-E3-E4-WB. For the MUL, 9 cycles; F-D-E1-E2-E3-E4-E5-E6-WB. Data forwarding is not allowed

Processing starts at cycle 1 with an empty set of reservation stations.

The initial RAT and the RAT at the end of cycle 8 are shown below. The relevant information in the reservation stations at the end of cycle 8 are also shown below.

RAT Initial

	V	TAG	VALUE
R0	1		1
R1	1		2
R2	1		3
R3	1		11
R4	1		19
R5	1		10
R6	1		20
R7	1		30

RAT at the end of Cycle 8

	V	TAG	VALUE
R0	1		1
R1	1		5
R2	1		3
R3	0	σ	11
R4	0	δ	19
R5	1		50
R6	0	π	20
R7	0	ρ	30

Add Reservation Stations at the end of Cycle 8

1		2	1		3	α
1		20	1		30	β
0	π		0	ρ		δ

Multiply Reservation Stations at the end of Cycle 8

1		5	1		50	π
0	π		1		3	ρ
0	δ		1		50	σ

Your job: Fill in the six entries below with the code sequence that produces the above RAT and reservation stations at the end of cycle 8.

	Opcode	DR	SR1	SR2
1				
2				
3				
4				
5				
6				

Name:__________________________

Problem 4 (20 points):

A vector processor with 11 cycle memory latency, two load ports and one store port, 16-way interleaved memory, and supporting full chaining is to be used to execute the compiled program for the following high level code:

```
for (i=0; i<10; i++)
{
    E[i] = (A[i]*B[i])+(C[i]*D[i]);
}
```

Part a (8 points) Write the vector code to accomplish this. You have available the following vector instructions:

```
MOVI VLEN, #n
MOVI VSTR, #n
VADD Vi,Vj,Vk  ; Vj, Vk are sources, Vi is dest
VMUL Vi,Vj,Vk
VSH  Vi,Vj     ; Vi <-- Vj shifted right one bit
VLD  Vi,A      ; Vi gets loaded with contents of memory, starting at A
VST  Vi,A      ; Contents of Vi gets stored in memory, starting at A
```

Part b (12 points) How long will it take for this code to execute? Assume that the latency for multiply is 6 cycles, add is 4 cycles, shift is 1 cycle, and for load or store is 11 cycles.

THE ANSWER

Name:__________________________

Problem 5 (20 points):

We wish to add to the LC-3b the new instruction MIN, which identifies the smallest value in k consecutive memory locations, and stores the result in one of the general purpose registers. Values range from -128 to +127 and occupy one byte of memory each. The condition codes will be set according to whether the result is negative, zero, or positive. We will use the unused opcode 1010 for this purpose. The format of the instruction will be

	15–12	11–9	8–6	5	4	3	2–0
MIN DR, BaseR, LengthR	**1010**	**DR**	**BaseR**	0	0	0	**LengthR**

where the address of the first location is in BaseR, and the number of locations k is in LengthR.

If LengthR is ≤ 0, the instruction is not defined.

There is no requirement that the contents of BaseR or LengthR remain unchanged at the end of this instruction's execution.

Example:
After the instruction MIN R1,R2,R3 is executed, where R2 contains x4000, R3 contains #5, and memory is as shown below:

x4000	**#7**
x4001	**#−44**
x4002	**#82**
x4003	**#−83**
x4004	**#105**

R1 will contain the value #-83, and condition N will be set.

Name:__________________________

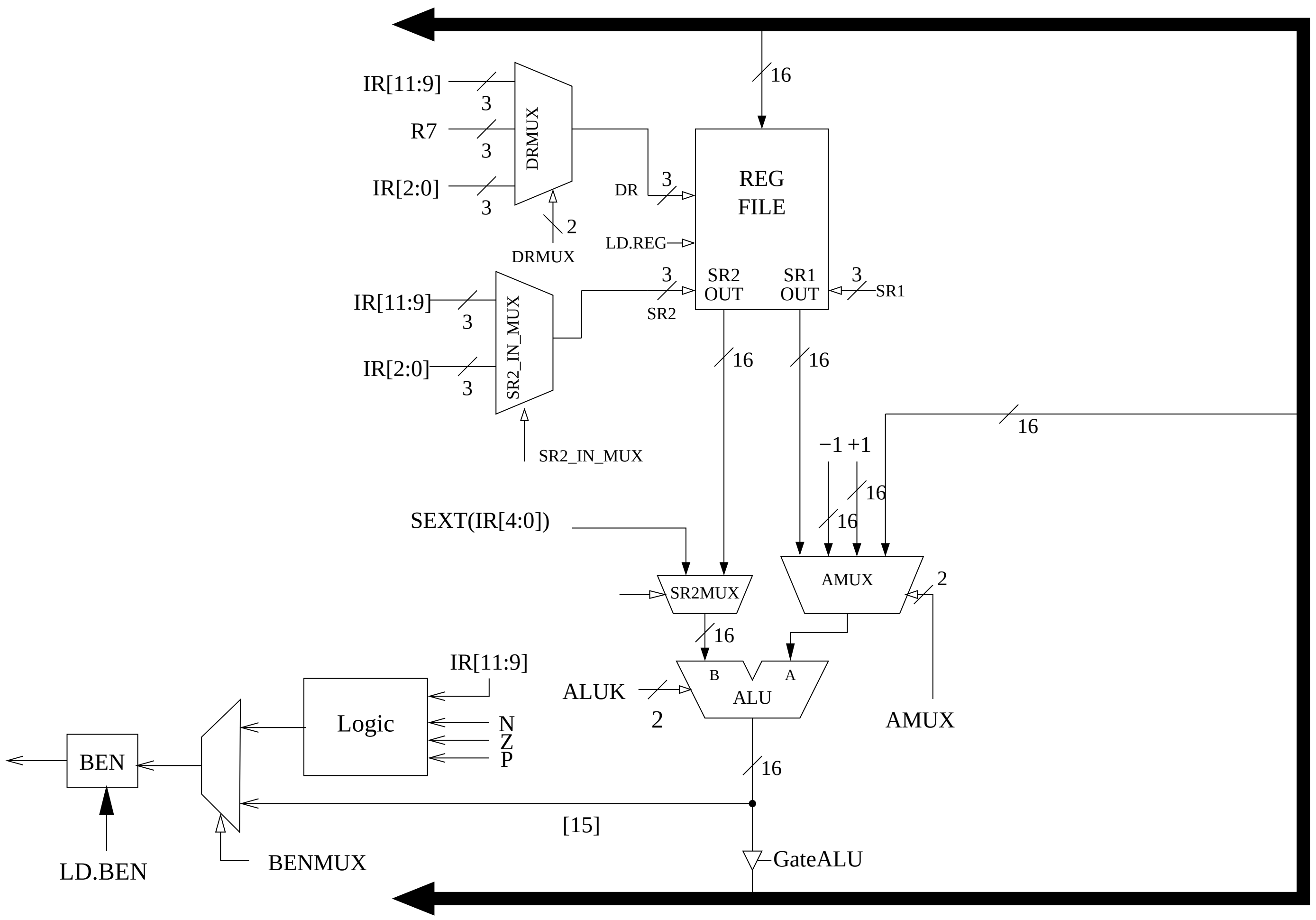
Problem 5 continued:

To implement MIN, we have chosen to add the following to the LC-3b data path:

1. A four-input mux to the A input of the ALU.
This requires a new control field, AMUX, specified as follows:
SR1/00, the SR1 source from the original data path,
MINUS1/01, a constant -1
PLUS/10, a constant 1
BUS/11, the contents on the bus
2. We have also augmented the set of control signals that can control SR2, and DR of the register file.

3. We have also added a BENMUX at the input of the BEN register, as shown. The additional source for the BEN flag is bit 15 of the output of the ALU.

This requires a new control signal, BENMUX, specified as follows: Logic/0
ALUout/1



Name:_________________________

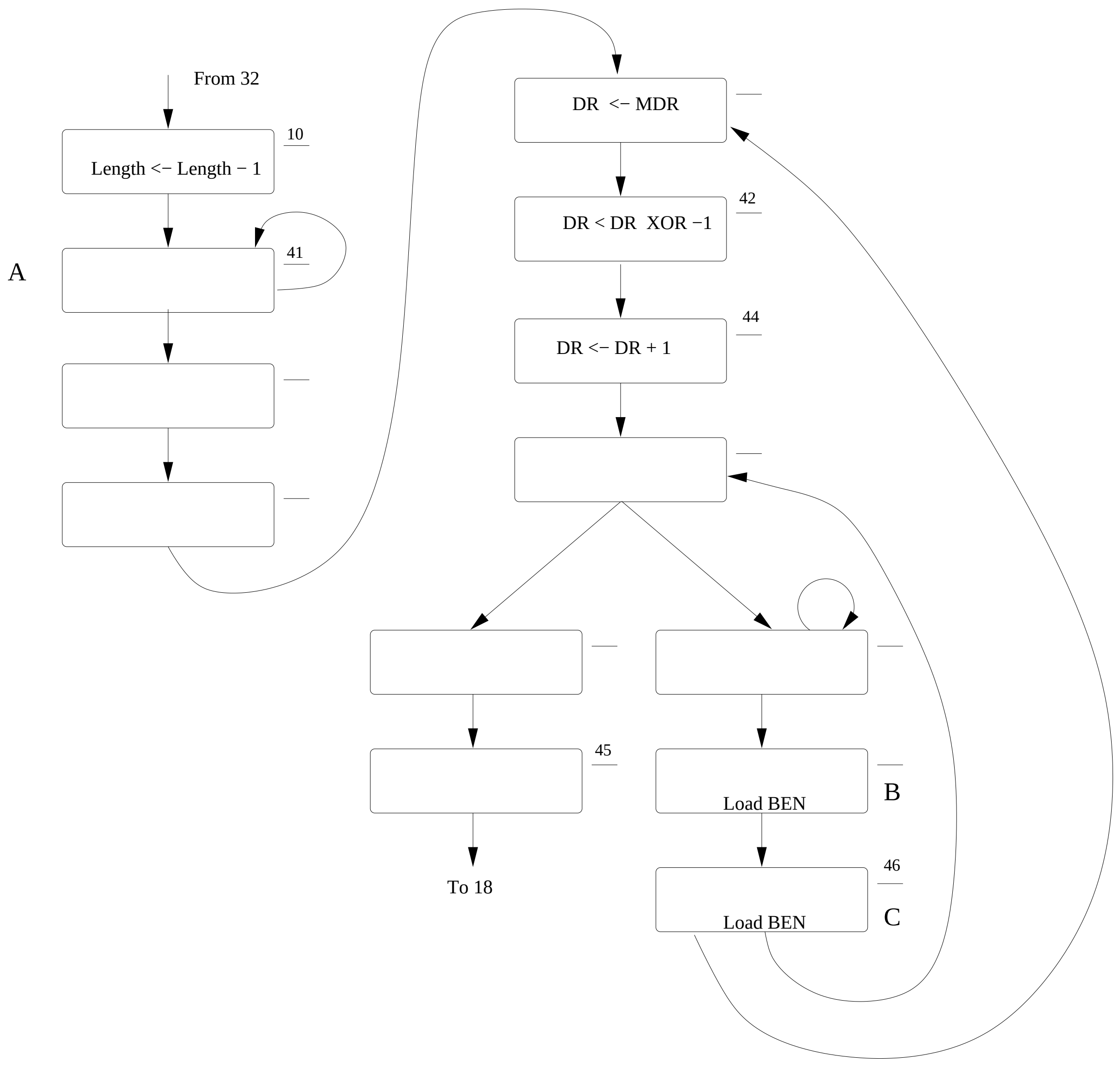
Problem 5 continued:

Part a. (12 points): We have provided the skeleton of the augmented LC-3b state machine for you to implement MIN. Using the notation we have adopted for the LC-3b, show what happens in every state.

We wish to assign state numbers 41 through 52 to the 12 new states. Five states have been assigned for you. Part of your job is to assign the remaining seven states from the set 43,47,48,49,50,51,52. Note that state 10, the entry point for the MIN opcode has also been assigned, in accordance with the 16-way decode from state 32.

Note that in some states we have specified some of the operations required. You are free to specify addtional operations to be performed in each state as you feel necessary.

Note that three of the states have been labeled additionally A,B,C. These labels are for your use in part b on the following page.



Name:__________________________

Problem 5 continued:

Part b. (8 points): Specify the control signals required (for both data path control and microsequencer control) to implement states A, B, C. Note that we have added control signals corresponding to our changes in the data path.

	IRD	Cond	J
A			
B			
C			

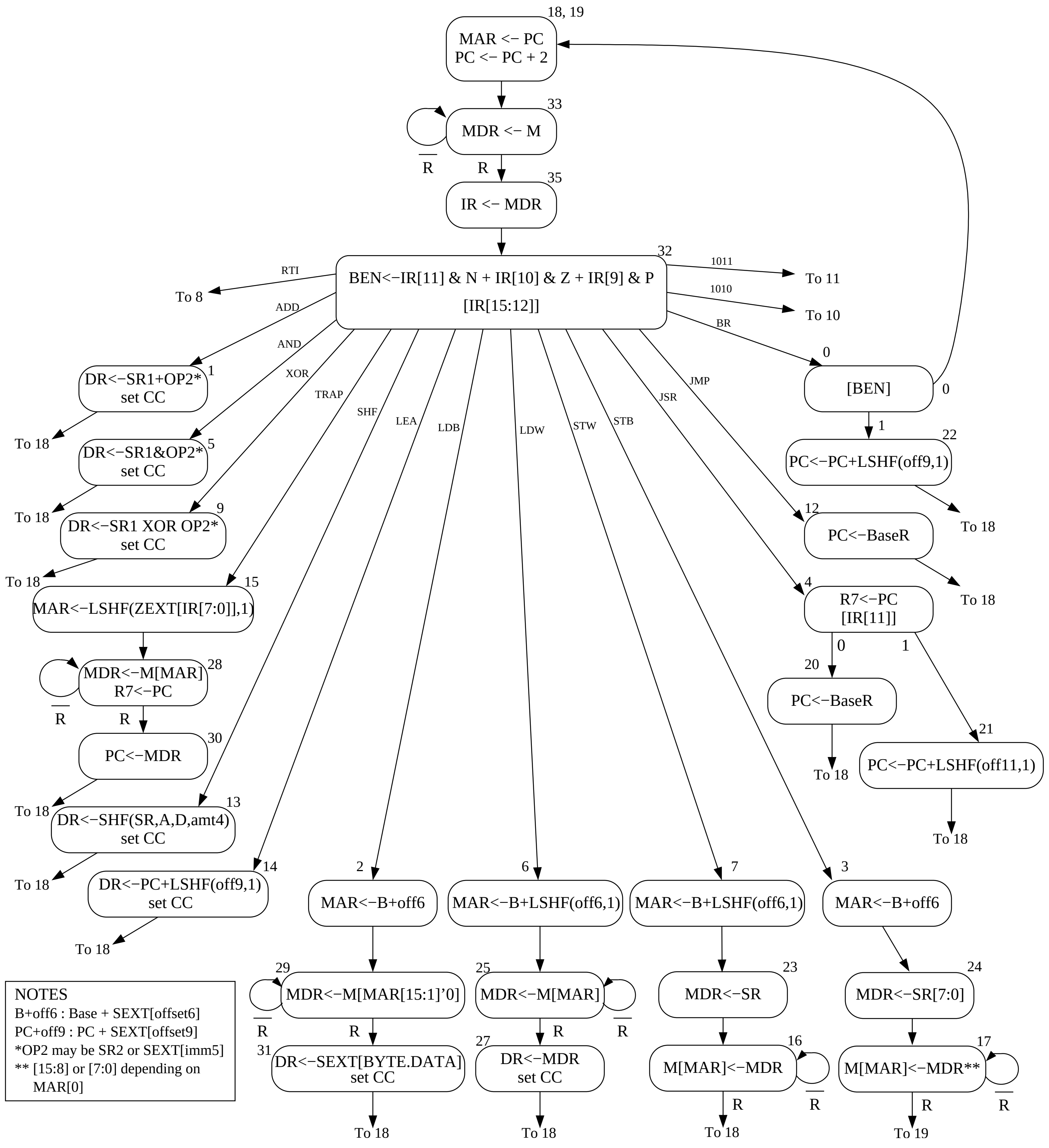
	LD.MAR	LD.MDR	LD.IR	LD.BEN	LD.REG	LD.CC	LD.PC	GatePC	GateMDR	GateALU	GateMARMUX	GateSHF	PCMUX	DRMUX	SR1MUX	ADDR1MUX	ADDR2MUX	MARMUX	ALUK	MIO.EN	R.W	DATA.SIZE	BENMUX	SR2_IN_MUX	AMUX
A																									
B																									
C																									

LC-3b ISA

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD+	0001				DR			SR1			0	00		SR2		
ADD+	0001				DR			SR1			1	imm5				
AND+	0101				DR			SR1			0	00		SR2		
AND+	0101				DR			SR1			1	imm5				
BR	0000				n	z	p	PCoffset9								
JMP	1100				000			BaseR			000000					
JSR	0100				1	PCoffset11										
JSRR	0100				0	00		BaseR			000000					
LDB+	0010				DR			BaseR			boffset6					
LDW+	0110				DR			BaseR			offset6					
LEA+	1110				DR			PCoffset9								
NOT+	1001				DR			SR			1	11111				
RET	1100				000			111			000000					
RTI	1000				000000000000											
LSHF+	1101				DR			SR			0	0	amount4			
RSHFL+	1101				DR			SR			0	1	amount4			
RSHFA+	1101				DR			SR			1	1	amount4			
STB	0011				SR			BaseR			boffset6					
STW	0111				SR			BaseR			offset6					
TRAP	1111				0000				trapvect8							
XOR+	1001				DR			SR1			0	00		SR2		
XOR+	1001				DR			SR			1	imm5				
not used	1010															
not used	1011															

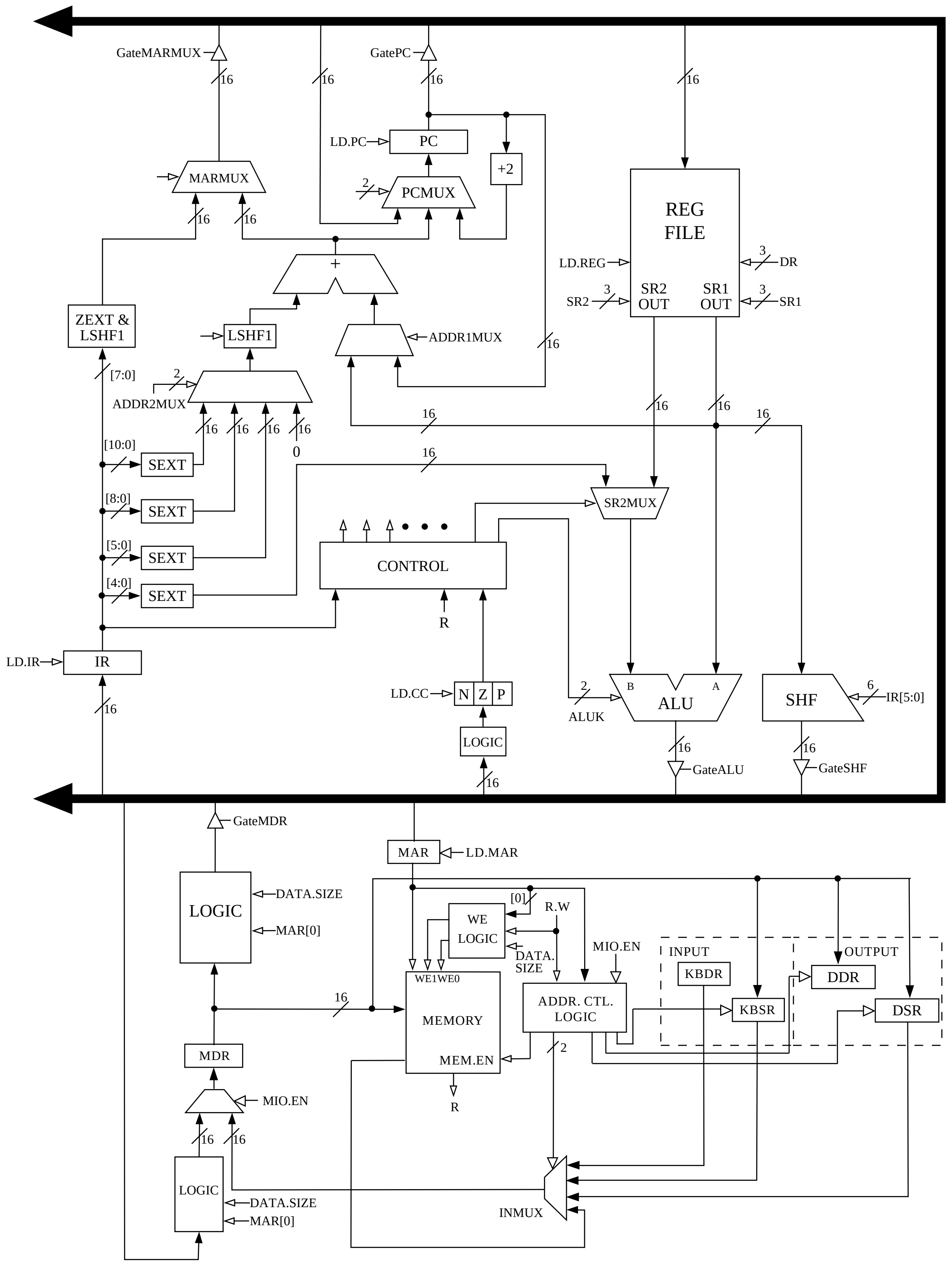
+ indicates instructions that modify condition codes.

A state machine for the LC-3b (from Appendix C)

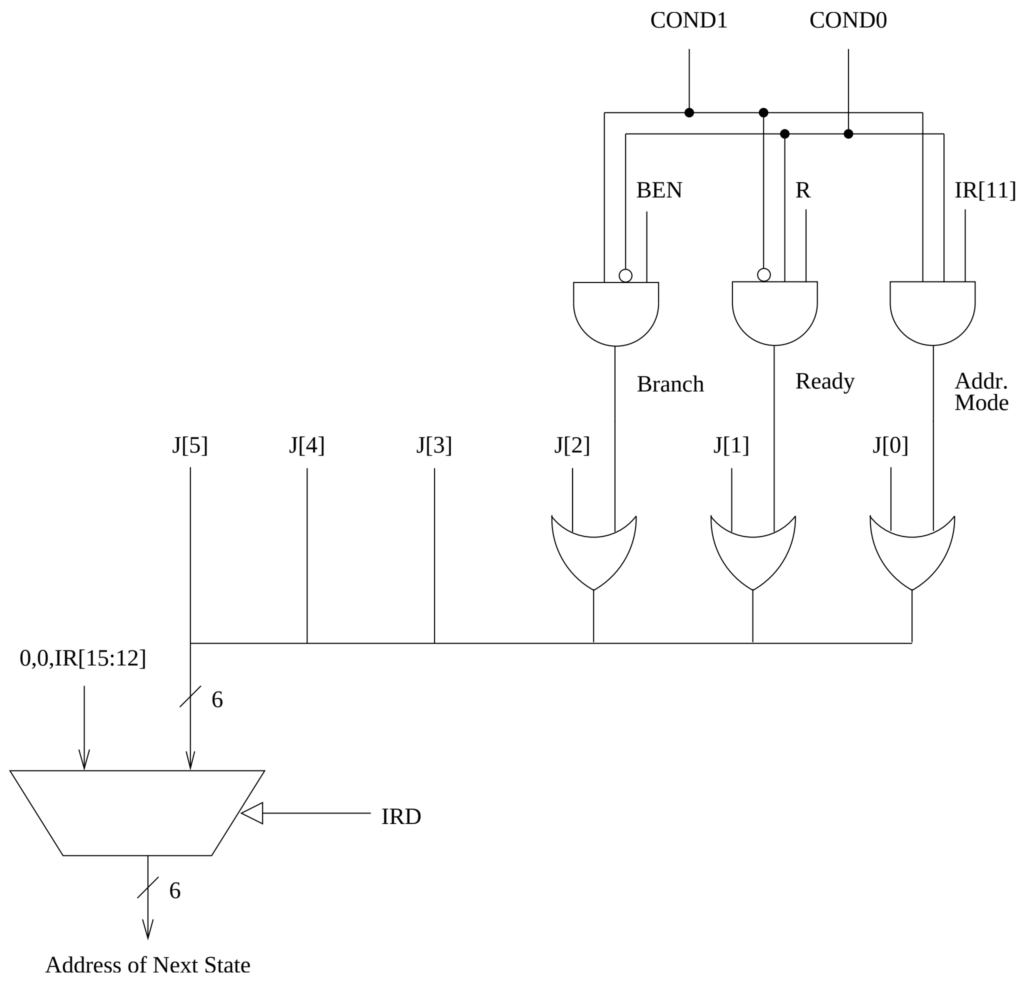


NOTES
B+off6 : Base + SEXT[offset6]
PC+off9 : PC + SEXT[offset9]
*OP2 may be SR2 or SEXT[imm5]
** [15:8] or [7:0] depending on
MAR[0]

The LC-3b datapath (from Appendix C)



The Microsequencer of the LC-3b base machine (from Appendix C)


COND1
COND0
BEN
R
IR[11]
Branch
Ready
Addr.
Mode
J[5]
J[4]
J[3]
J[2]
J[1]
J[0]
0,0,IR[15:12]
6
IRD
6
Address of Next State