Fall 2003 – CSE 207 Digital Design Final Exam Review

Numbers, Representation, and Arithmetic.

(a) Represent the following decimal numbers as four-bit 2's-compliment numbers:

i. 4 = 0100

ii. –3 = 1101

iii. –5 = 1011

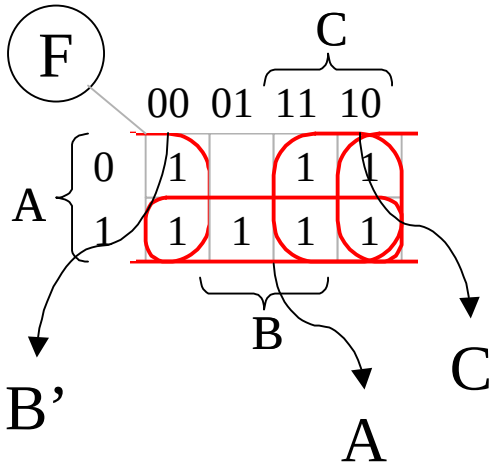
(b) (3) Subtract 1110_2 from 0010_2 by adding the 2's complement of 1110_2 to 0010_2

```
2's compliment of 1110 is 0001+1 = 0010

    0  1  0  0
   0  0  1  0
 + 0  0  1  0
--------------
0  0  1  0  0
```

(c) (8) Using any technique generate the minimal sum of products. Box your final answer. You must show work for full credit.

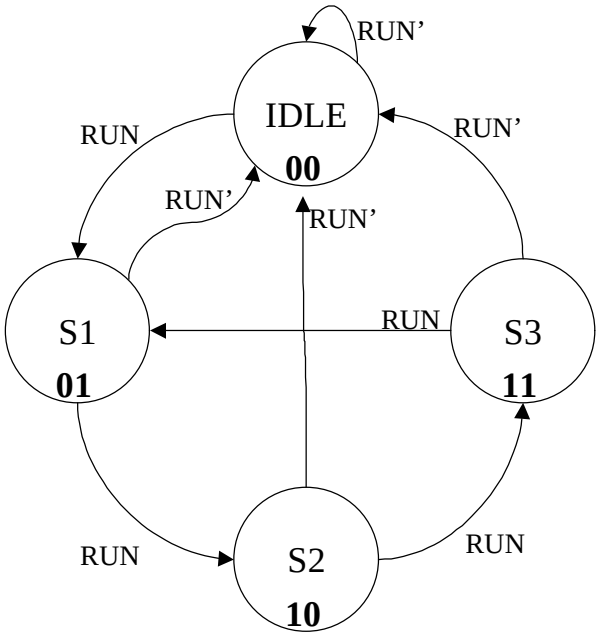
C	B	A	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



$$F = C + B' + A$$

Sequential Circuit Design

You are to design a circuit to implement the state transition diagram shown to the right.



This will be a Moore machine (outputs a function of the current state). It functions as follows: when RUN is true, the machine cycles through the states S1, S2, S3. As soon as RUN is false, the machine transfers to the idle state and waits there indefinitely. The system has the following outputs A, B, C. A it output when in state IDLE, S1, and S2. B is output when in state S2 and S3. C is output when in state S3 only.

a) Generate the truth table and excitation equations for the NextState Q_1^* Q_0^* given the CurrentState Q_1 Q_0 and the input RUN.

RUN	Q_1	Q_0	Q_1^*	Q_0^*
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

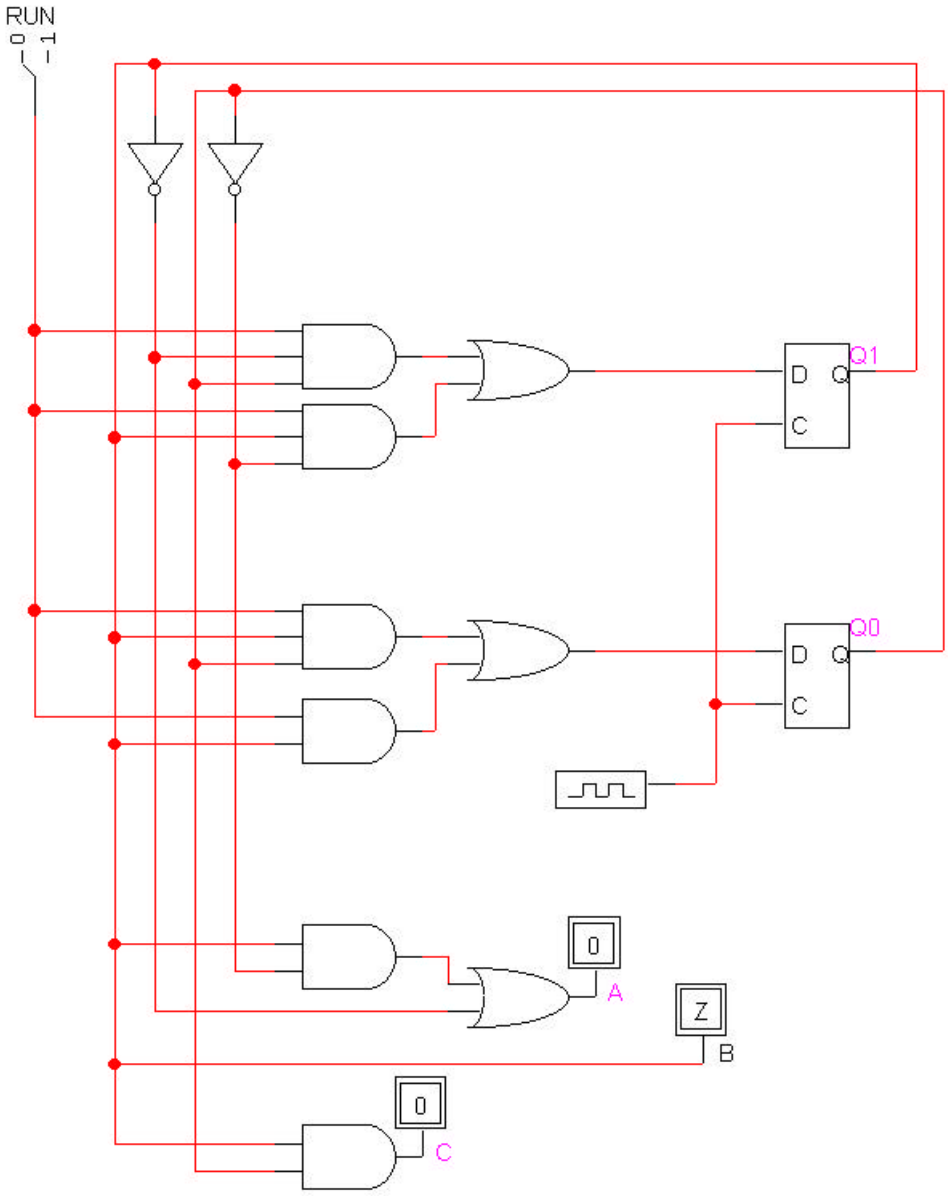
$Q_1^* = (RUN * Q_1' * Q_0) + (RUN * Q_1 * Q_0')$
$Q_0^* = (RUN * Q_1) + (RUN * Q_1' * Q_0')$

b) Generate the truth table and output equations A, B, and C given the CurrentState.

StateName	Q_1	Q_0	A	B	C
IDLE	0	0	1	0	0
S1	0	1	1	0	0
S2	1	0	1	1	0
S3	1	1	0	1	1

$A = Q_1' + (Q_1 * Q_0')$
$B = Q_1$
$C = (Q_1 * Q_0)$

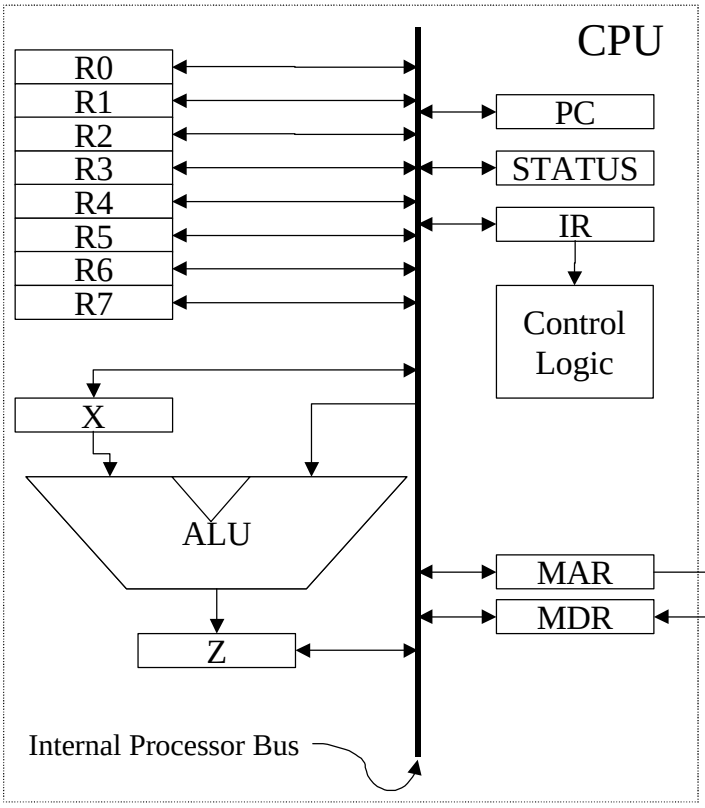
c) Draw the circuit showing all inputs and outputs.



Microcode.
Management has decided that we need to implement the INC mem instruction as described in project #4. Generate the register transfers and list all control signals necessary to implement this instruction. You will not need to worry about active-high or active-low – just indicate which control lines are asserted. All valid control signals are listed on the following sheet. You will start your microcode at the completion of Fetch (i.e. the instruction is in the IR – what additional steps do you need to complete the execution of this instruction).

```
MAR ← dst, RAM_OE
MDR ← RAM, RAM_OE
X ← MDR, ALU=0000, M=0, Cin=0 (A PLUS 1)
Z ← ALUResult
MDR ← Z
RAM ← MDR, RAM_WE
```

Information:



Control Signals:
PC_INC, PC_CLR, IR_CLR, ALU_S3..0, ALU_M, ALU_Cin, C_EN, Z_EN, Reset
R0..7_OE, RX_OE, RZ_OE, MAR_OE, MDR_OE, MDR_MOE, IR_OE, PC_OE, RAM_OE
R0..7_IE, RX_IE, RZ_IE, MAR_IE, MDR_IE, MDR_MIE, IR_IE, PC_IE, RAM_WE

Registers:
R0, R1, R2, R3, R4, R5, R6, R7, RX, RZ, MAR, MDR, PC, STATUS, IR

Other Signals:
IRdst, IRsrc, IRbits, RelAddr, ALUResult, RAM

ALU:

Selection	Active-High Data		
	M=H	M=L; ARITHMETIC FUNCTIONS	
S3 S2 S1 S0	LOGIC FUNCTIONS	Cn′ = H (no carry)	Cn′ = L (with carry)
L L L L	F = A′	F = A	F = A PLUS 1
L L L H	F = (A+B)′	F = A + B	F = (A+B) PLUS 1
L L H L	F = A′B	F = A+B′	F = (A+B′) PLUS 1
L L H H	F = 0	F = MINUS 1 (2′s Cmp)	F = ZERO
L H L L	F = (AB)′	F = A PLUS AB′	F = A PLUS AB′ PLUS 1
L H L H	F = B′	F = (A+B) PLUS AB′	F = (A+B) PUS AB′ PLUS 1
L H H L	F = A⊕B	F = A MINUS B MINUS 1	F = A MINUS B
L H H H	F = AB′	F = AB′ MINUS 1	F = AB′
H L L L	F = A′+B	F = A PLUS AB	F = A PLUS AB PLUS 1
H L L H	F = (A⊕B)′	F = A PLUS B	F = A PLUS B PLUS 1
H L H L	F = B	F = (A+B′) PLUS AB	F = (A+B′) PLUS AB PLUS 1
H L H H	F = AB	F = AB MINUS 1	F = AB
H H L L	F = 1	F = A	F = A PLUS A PLUS 1
H H L H	F = A+B′	F = (A+B) PLUS A	F = (A+B) PLUS A PLUS 1
H H H L	F = A+B	F = (A+B′) PLUS A	F = (A+B′) PLUS A PLUS 1
H H H H	F = A	F = A MINUS 1	F = A