FINAL EXAMINATION
COMPUTER ARCHITECTURE AND DIGITAL DESIGN
03-60-265-01

UNIVERSITY OF WINDSOR
SCHOOL OF COMPUTER SCIENCE
Fall 2012

Last Name: MARKER COPY

First Name: .....

Student ID: .....

PLEASE READ CAREFULLY BEFORE YOU START

1. This is a CLOSED book test; no notes, textbooks, calculators or computer aids are allowed.
2. No cellular phones or any other electronic communications devices may be used – all such devices must be turned off during the examination.
3. PRINT your name legibly and clearly with your Student ID in the space indicated above.
4. You will be asked to sign your name, once during the exam (sign-in) and once before leaving the exam room (sign-out).
5. Answer all the questions in the space provided. DO NOT REMOVE any pages or attach any papers to this test (except the last page). If you need more space please request an additional exam booklet which MUST be returned with this exam paper with your name and ID clearly written on it.
6. You are not allowed to give or receive unauthorized help with your test. Any misconduct, as outlined by the Senate bylaw 31 article I, will be reported accordingly.
7. You have 3 hours to complete this test.
8. Final examination papers are not returned to students.

Good Luck!

I AGREE TO THE ABOVE TERMS AND WILL NEITHER RECEIVE NOR GIVE UNAUTHORIZED HELP ON THIS EXAM

....

SIGNATURE **DATE**

SIGN HERE

MARKS:

Q1	/11
Q2	/16
Q3	/12
Q4	/15
Q5	/08
TOTAL:	/62

Question 1. [11 marks]

Part A. [3 marks]
Simplify the following Boolean function using a four-variable Karnaugh map, expressing your answer as an algebraic expression in **SOP** form with minimal number of literals.

F(A,B,C,D) = Σ m(0,1,2,3,4,7,11,15)

AB\CD	00	01	11	10
00	1	1	1	1
01	1		1	
11			1	
10			1	

SOP form: F = A'B' + C D + A'C'D' 6 literals

Check for equivalent answers that are algebraically equal – note that the answers must be expressible in SOP format. Deduct marks for not having a minimal number of literals.

Part B. [3 marks]
Simplify the following Boolean function using a four-variable Karnaugh map, expressing your answer as an algebraic expression in **POS** form with minimal number of literals.

F(A,B,C,D) = Π M(0,1,2,4,6,8,9,10,12,14)

AB\CD	00	01	11	10
00	0	0		0
01	0			0
11	0			0
10	0	0		0

POS form: F = (D)(B+C) 3 literals

Check for equivalent answers that are algebraically equal – note that the answers must be expressible in POS format. Deduct marks for not having a minimal number of literals.

Question 1 – continued.

Part C. [3 marks]
Simplify the following Boolean function, **F**, together with don't care conditions, **dc**, using a four-variable Karnaugh map, expressing your answer as an algebraic expression in **SOP** form with minimal number of literals.

F(A,B,C,D) = Σ m(0,5,6,15) ; dc(A,B,C,D) = Σ m(1,4,7,10,13)

AB\CD	00	01	11	10
00	1	-		
01	-	1	-	1
11		-	1	
10				-

SOP form: F = A'C' + B D + A'B 5 literals

Check for equivalent answers that are algebraically equal – note that the answers must be expressible in SOP format. Deduct marks for not having a minimal number of literals.

Part D. [2 marks]
Assume that X is a Boolean variable. Using **only the Postulates of Boolean Algebra**, prove the following Boolean relation: (use of truth tables is not allowed for this question)

X + 1 = 1

ANSWER:

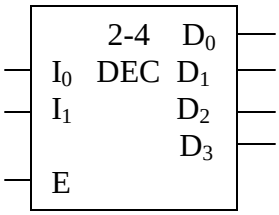
X + 1 = 1 . (X + 1)
= (X + X') . (X + 1)
= X + (X'. 1)
= X + X'
= 1 QED!

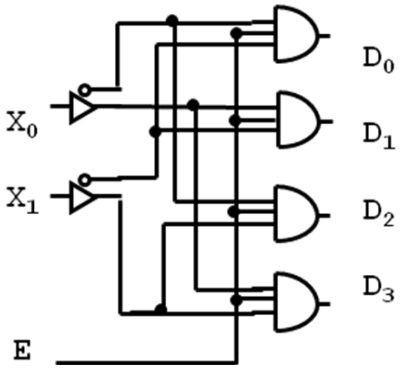
Deduct marks for not using Postulates alone.

Question 2. [16 marks]

Part A. [4 marks]
Draw the logic gate circuit diagram for a 2-to-4 line decoder with Enable (whose high-level block diagram is shown below).

The 2-to-4 line Decoder

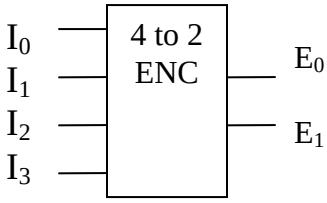




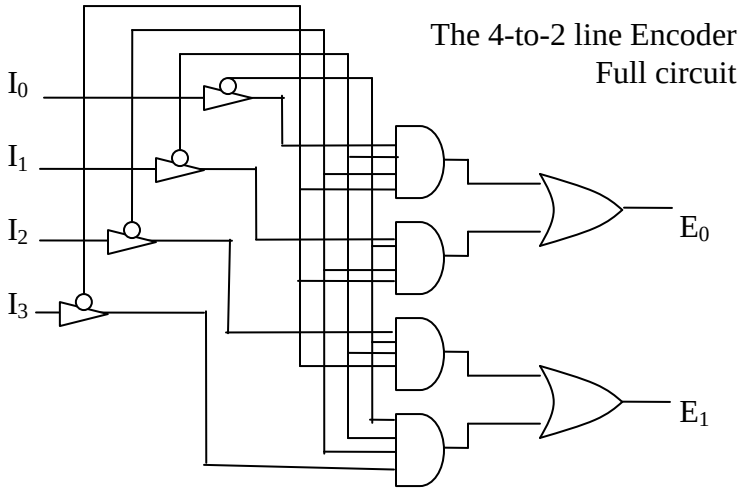
Note that the inputs have been labeled X_0, X_1 above – this should not be considered important so long as the inputs are labeled correctly (ie. logic is correct). The Enable input is simply connected to each AND gate. Also, note that I have used a special symbol (triangle with open circle) above to separate the input line into both input and its complement (from the open circle).

Part B. [4 marks]
Draw the logic gate circuit diagram for a 4-to-2 line encoder (whose block diagram is shown below).

The 4-to-2 line Encoder



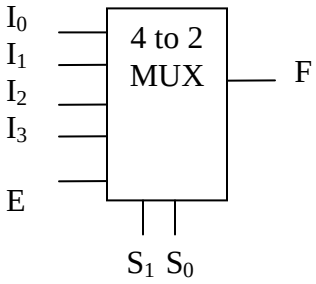
The 4-to-2 line Encoder
Full circuit

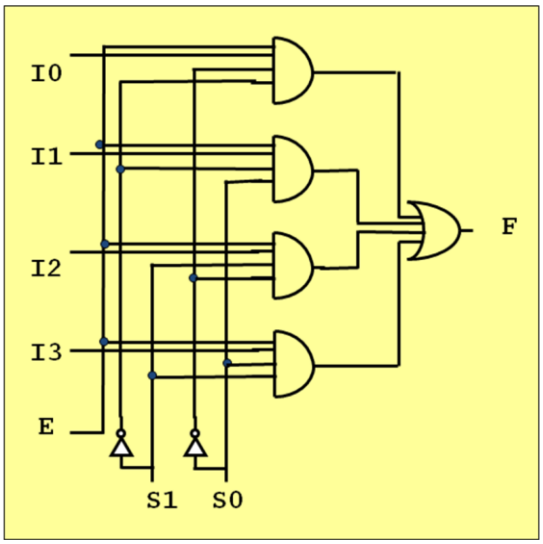


Marker: Check the diagram of the full circuit carefully – students may approach it somewhat differently, yet still arrive at a correct diagram. Note that I did not ask for an Enable input. There is no need to show a truth table, but if they answer with a correct truth table but do not provide a diagram, then they may still obtain full marks. If they provide a correct truth table but make mistakes in the diagram then deduct from 1-2 marks.

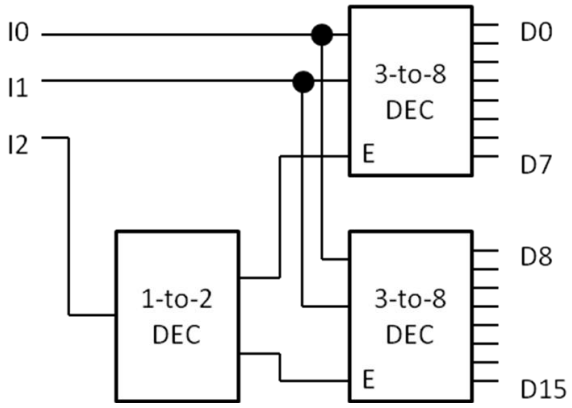
Part C. [4 marks]
Draw the logic gate circuit diagram for a 4-to-1 line multiplexer with Enable input (E) and output (F) (whose block diagram is shown below).

The 4-to-1 multiplexer with Enable





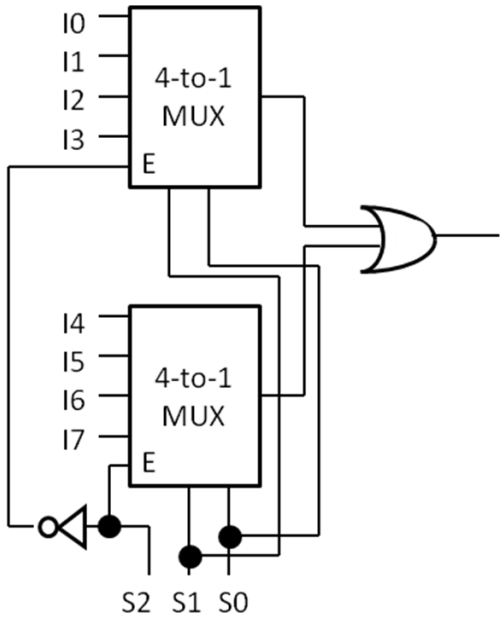
Part D. [2 marks]
Draw the logic block diagram for a 4-to-16 line decoder using only two 3-to-8 line decoders plus a single 1-to-2 line decoder. Label your diagram completely.



Marker: Give most marks if the decoders are correctly identified and connected as above. For labeling, deduct marks only if it is not fully clear as to how the inputs and outputs are arranged (eg. the student may reverse the numeric subscripts as shown above, or not provide complete labels and still get full marks – but if the labels are confused then deduct marks).

Part E. [2 marks]
Draw the logic block diagram for an 8-to-1 line multiplexer using only two 4-to-1 line multiplexers with Enable inputs, plus additional logic gates. Label your diagram completely.

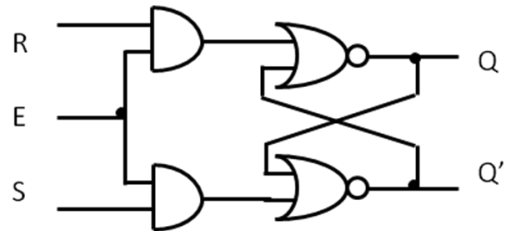
Marker: This question may be answered in different ways, so check the logic for correct equivalent answers. In my answer I have used an inverter on the S2 selector input to distinguish the top/bottom multiplexers, and an OR gate to combine the two multiplexer outputs.



Question 3. [12 marks]

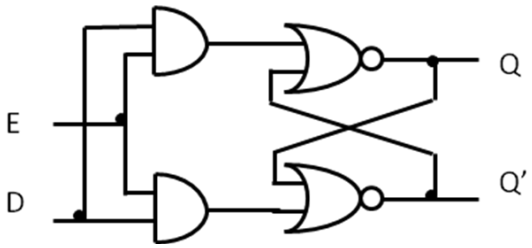
Part A. [4 marks]
Draw the logic gate circuit diagram for an S-R latch with Enable input. Also, provide the State Table for the S-R latch in the case when Enable=1.

S	R	Q^+
0	0	Q
0	1	0
1	0	1
1	1	Forbidden



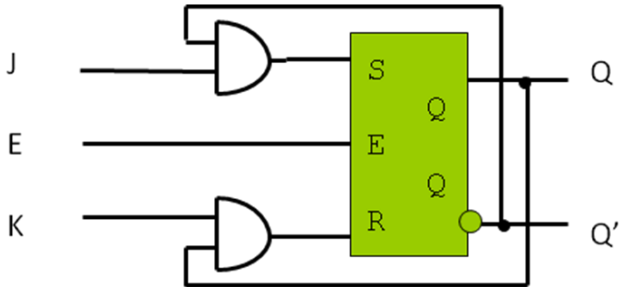
Part B. [2 marks]
Draw the logic gate circuit diagram for an D-flipflop with Enable input. Also, provide the State Table for the D-flipflop in the case when Enable=1.

D	Q^+
0	0
1	1



Part C. [4 marks]
Draw the logic gate circuit diagram for a JK-flipflop with Enable input. Also, provide the State Table for the JK-flipflop in the case when Enable=1.

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	Q'



Marker: Note that I have used a block diagram for the internal SR latch – this is acceptable for student answers if they do not specify the complete SR logic gate details.

Part D. [2 marks]
Express the Characteristic Equation for the JK-flipflop.

$Q^+ = JQ' + K'Q$

Question 4. [15 marks]
Both parts of this question relate to the issue of instruction cycles, assuming the approach used in the lectures. In each part you must provide the complete register transfer language (RTL) statements that specify the sequence of micro-operations that occur during the instruction cycle. In each part, the first three steps in the sequence are shown (with the control conditions separated from the micro-operations in different columns). Also, it is necessary to treat the sequence counter (SC) explicitly in each RTL statement. Finally, interrupts are not considered in this question. Note that all machine language opcodes are provided on the last page of the exam.

Part A. [3 marks]
Provide all details of the complete timing and logic control conditions and micro-operations performed at each step required to execute the machine language instruction: **`7020 = INC`**

Control Conditions	Micro-operation(s)
T0	AR = PC, SC = SC+1
T1	IR = M[AR] , PC = PC + 1, SC = SC+1
T2	I = IR(15) , {D0,…,D7} = DEC(IR(12..14)), AR = IR(0..11) , SC = SC+1
IR(5).I'.D7.T3	AC = AC + 1, SC = 0

Marker: Students may show more or fewer steps. What is important here is that they perform all the micro-operations as shown and do not show conflicting micro-operations. Control logic must also be complete but deduct only ½ marks and do not doubly penalize for repeated errors. Some students may include use of the DR register as an intermediary for bus transfers between CPU and RAM.

Part B. [3 marks]
Provide all details of the complete timing and logic control conditions and micro-operations performed at each step required to execute the machine language instruction: **`7010 = SPA`**

Control Conditions	Micro-operation(s)
T0	AR = PC, SC = SC+1
T1	IR = M[AR] , PC = PC + 1, SC = SC+1
T2	I = IR(15) , {D0,…,D7} = DEC(IR(12..14)), AR = IR(0..11) , SC = SC+1
IR(4).(AC>0). I'.D7.T3	PC = PC + 1, SC = 0

Marker: Control logic must be complete and all micro-operations must be properly stated. Deduct ½ marks for small errors.

Part C. [3 marks]
Provide all details of the complete timing and logic control conditions and micro-operations performed at each step required to execute the machine language instruction: **`4400 = BUN 400`**

Control Conditions	Micro-operation(s)
T0	AR = PC, SC = SC+1
T1	IR = M[AR] , PC = PC + 1, SC = SC+1
T2	I = IR(15) , {D0,…,D7} = DEC(IR(12..14)), AR = IR(0..11) , SC = SC+1
I'.D4.T3	DR = M[AR], SC = SC+1
I'.D4.T4	PC = DR(0..11), SC = 0

Marker: Students may show more or fewer steps than above. What is important here is that they perform all the micro-operations as shown and do not show conflicting micro-operations. Control logic expressions must be complete. Deduct ½ marks for small errors.

Part D. [6 marks]
Provide all details of the complete timing and logic control conditions and micro-operations performed at each step required to execute the machine language instruction: **`E800 = ISZ 800 I`**

Control Conditions	Micro-operation(s)
T0	AR = PC, SC = SC+1
T1	IR = M[AR] , PC = PC + 1, SC = SC+1
T2	I = IR(15) , {D0,…,D7} = DEC(IR(12..14)), AR = IR(0..11) , SC = SC+1
I.D6.T3	DR = M[AR], SC = SC+1
I.D6.T4	AR = DR(0..11), SC = SC+1
I.D6.T5	AC = M[AR], SC = SC+1
I.D6.T6	AC = AC + 1, SC = SC+1
I.D6.T7	M[AR] = AC, SC = SC+1
I.D6.T8.AC'	PC = PC + 1 // IF AC contains 0 then increment PC
I.D6.T9	SC = 0

Marker: Students may show more or fewer steps than above, such as using DR for bus transfers with subsequent transfer to AC. What is important here is that they perform the micro-operations as shown and do not show conflicting micro-operations. The final step above emphasizes that resetting SC must be done for either AC=0 or not = 0. This can be done in different ways (including fewer steps) so check this carefully.

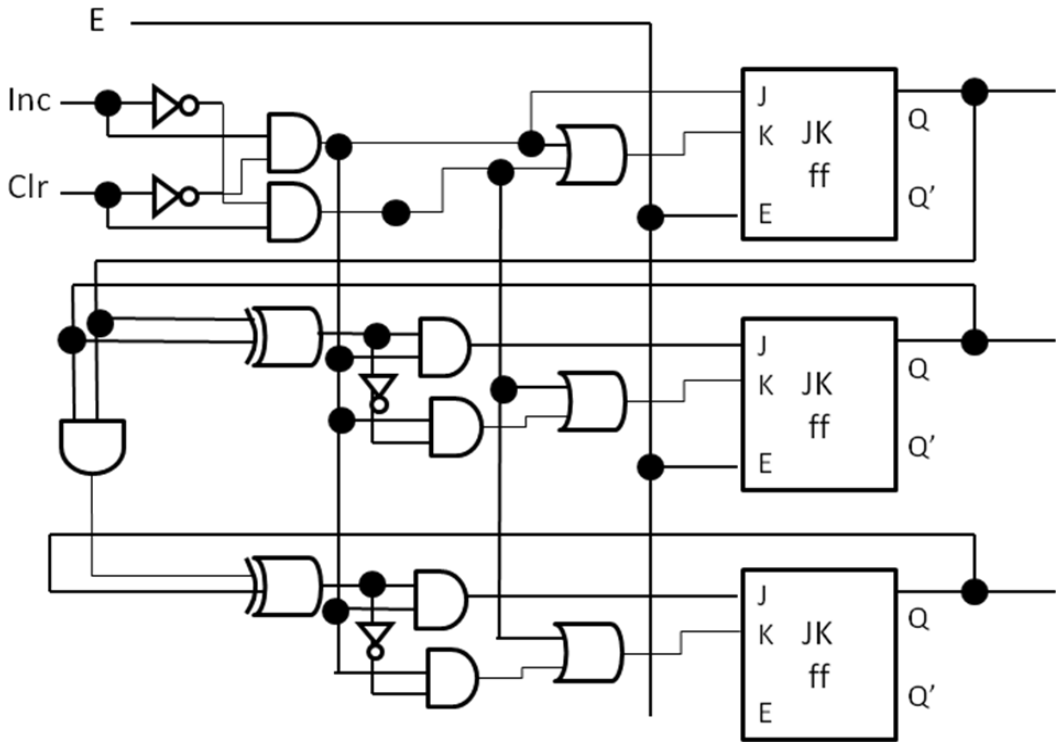
Question 5. [8 marks]

Design and draw the logic diagram for a 3-bit counter register consisting of three JK-flipflops (use block diagrams for these) and the control interface logic (expressed using logic gates). The inputs to the counter must include (a) Enable, (b) Clear to zero, and (c) Increment. Note that Clear and Increment are performed only if Enable is 1. If Enable=1 and both Clear and Increment are 0 or 1, the JK-flipflops simply refresh themselves.

My approach to this problem is described below:

a. I have used JK-flipflops with Enable. By connecting the E input directly to the Enable inputs of all three JK's, when E=0 the flipflops simply refresh themselves.
b. When E=1, I distinguish into three cases, I=C (either 0 or 1), for which a refresh must happen; then I=0 and C=1 which clears all three flipflops; finally, I=1 and C=0 for which an increment must happen.
c. Construct a truth table for the 3-bit incrementer. Combining this step with the step above, leads to the expressions for each JK input for each flipflop (labeled 0, 1 and 2), assuming E=1 in all cases:
   J0 = IC' K0 = I'C + IC'
   J1 = IC'(Q0 xor Q1) K1 = I'C + IC'(Q0 xor Q1)'
   J2 = IC'(Q2 xor (Q0.Q1)) K2 = I'C + IC'(Q2 xor (Q0.Q1))'

This results in the diagram shown below.



Marker: This is a challenging problem for an examination. Approach marking by looking to see if the student has understood what is asked and has developed an approach that should produce good results – this should be quickly apparent. If the student clearly does not have much idea but produces somewhat useful partial results, give between 0-4. If the student seems to have a workable strategy, but makes errors, give between 4-6. Only if the student has a solid strategy and makes a few errors should you give between 6-8 marks. If the approach is clearly solid and some simple mistake is found you may still give 8/8, but use good judgement.

This page contains various definitions and information provided freely for each student to use for the examination, if required.

Full Register set for Mano's machine:

PC, AR, AC, DR, IR, TR, I, E, R, FGI, FGO, SC, IEN, INPR, OUTR

Operation Codes and Mnemonics for Mano's machine:

0 (8)	AND	1 (9)	ADD	2 (A)	LDA	3 (B)	STA				
4 (C)	BUN	5 (D)	BSA	6 (E)	ISZ						
7800	CLA	7400	CLE	7200	CMA	7100	CME	7080	CIR		
7040	CIL	7020	INC	7010	SPA	7008	SNA	7004	SZA		
7002	SZE	7001	HLT								
F800	INP	F400	OUT	F200	SKI	F100	SKO				
F080	ION	F040	IOF								

Boolean Postulates:

P1: Closure: For every x,y in B there exist two combinational operators + and . where x+y is in B and x.y is in B
P2: Identity: There exist identity elements 0,1 in B relative to the operations + and ., such that for every x in B: 0+x = x+0 = x and 1.x = x.1 = x.
P3: Commutativity: The operations + and . are commutative for all x,y in B: x+y = y+x and x.y = y.x.
P4: Distributivity: Each operation + and . is distributive over the other; that is, for all x,y,z in B: x.(y+z) = x.y+x.Z and x+(y.z) = (x+y).(x+z)
P5: Complementation: For every element x in B there exists an element ~x, called the complement of x, satisfying: x+~x = 1 and x.~x = 0
P6: Existence: There exist at least two elements x,y in B such that x ≠ y

Fundamental Logic Micro-operations:

F = 0	F = AB	F = AB′	F = A
F = A′B	F = B	F = AB′+A′B	F = A+B
F = A′B′	F = AB+A′B′	F = B′	F = A+B′
F = A′	F = A′+B	F = A′+B′	F = 1

