School of Computer Science

60-265-01 Computer Architecture and Digital Design

Fall 2008

Midterm Examination # 1 B

Wednesday, November 5, 2008

Sample Marking Scheme

Duration of examination: 75 minutes

1. Answer all questions on this examination paper in the space provided.
2. This is a closed-book examination – no notes or books or electronic computing or storage devices may be used.
3. Do not copy from other students or communicate in any way. All questions will be answered only by the attending proctors.
4. All students must remain seated during the last 5 minutes of the examination.
5. The examination must be surrendered immediately when the instructor announces the end of the test period.
6. The last page is blank. It may be detached and used as a worksheet. It does not have to be handed in. You may use the back of any page of the exam to continue an answer – indicate the question (and Part) in all cases.

7. Each student must sign the examination list before leaving the classroom.

Total mark obtained: _____

Maximum mark: **56**

Question 1. [9 marks]
Answer each of the following questions **briefly**. The marks for each question are indicated in square brackets.

(a) State the six (6) fundamental logic gates (not including the inverter) and, for each gate, provide a complete truth table that fully defines the logic of the gate. **[6 marks]**

AND

X/Y	0	1
0	0	0
1	0	1

OR

X/Y	0	1
0	0	1
1	1	1

NAND

X/Y	0	1
0	1	1
1	1	0

NOR

X/Y	0	1
0	1	0
1	0	0

XOR

X/Y	0	1
0	0	1
1	1	0

NXOR (or XNOR)

X/Y	0	1
0	1	0
1	0	1

(b) State the five (5) principal components of the von Neuman architecture for a stored program digital computer. **[3 marks]**

RAM – volatile memory
CPU – central processing unit
Bus – interconnection network (connects all other components)
Mass storage – non-volatile storage
I/O – peripheral devices for user input/output

NOTE: some allowance is made for using alternative terminology, but it has been emphasized that accepted terminology is important. Here it is vital to recognize the 5 principal components because each represents a distinctly different set of requirements reflected in architecture and design.

Question 2. [9 marks]
Answer all parts of this question.

A. Show the value of all bits of a 12-bit, unsigned binary representation for the number equivalent to decimal 251. **[1 mark]**

251 converts to: 128 + 64 + 32 + 16 + 8 + 2 + 1

In 12-bit binary notation: 000011111011

Alternatively, divide 251 (and its quotients) by 2 repeatedly, collecting the remainders (0 or 1) from low order to high order to form the bit string above.

B. Show the value of all bits of a 12-bit, unsigned binary coded octal representation for the number equivalent to decimal 215. **Also**, express the value in octal form. **[2 marks]**

215 converts to octal form: 327 : 000 011 010 111

To convert 215, divide by 8 to get a quotient of 26 and remainder of 7 (the low order octal digit). Then divide 26 by 8 to get a quotient of 3 with remainder 2. The remaining 3 is the high order digit, hence the answer in base-8 (octal) is 327 or 3x64 + 2x8 + 7x1 = 192+16+7= 215.

C. State the largest possible value, in decimal notation, that can be represented using 12 bits in unsigned binary coded octal. **[1 mark]**

NOTE: 12 bits can be subdivided into 4 x 3-bit subfields, one sub-field for each octal digit 0..7

The largest value is represented by the digits being 7 in each octal position, hence:

7*8^3 + 7*8^2 + 7*8^1 + 7*8^0 = 4095

D. Show the value of all bits of a 12-bit representation for the number equivalent to decimal 251 in unsigned binary coded decimal (BCD). **[2 marks]**

In 12-bit binary coded decimal notation, each decimal digit requires 4 bits to represent the decimal digits 0..9, hence 251 is stated as: **001001010001**

Question 2 - continued

E. State the largest possible value that can be represented using 12 bits in unsigned binary coded decimal (BCD). **[1 mark]**

NOTE: Each BCD digit requires 4 bits. Thus, 12 bits can be subdivided into 3 4-bit subfields, one sub-field for each BCD digit 0..9

9*10^2 + 9*10^1 + 9*10^0 = 999

F. What is the **radix (base)** of the integer numbers (coefficients) below if the solution to the quadratic equation shown is **x=5**? **[2 marks]**

$$\mathbf{x^2 - 12\,x + 39 = 0}$$

Recall that we use *positional representation*, where a number is represented in an arbitrary base (or radix) using:

$$N = \sum_{k=0}^{L} d_k \; R^k$$

Hence, since x=5, it follows that (using decimal, or base-10 radix):

$$\mathbf{5^2 - (1*R+2)*5 + (3*R+9) = 25 - 5R - 10 + 3R + 9 = 24 - 2R = 0}$$

It follows that **R = 12**.

Question 3. [12 marks]
Answer all parts of this question.

Part A. Simplify the following Boolean SOP expression using Boolean algebra (Do NOT use Karnaugh maps or truth tables). To earn full marks, you are required to justify your result (ie. show your work!). **[4 marks]**

$$F(W,X,Y,Z) = WXYZ + WXY'Z + WX'Y'Z + WX'YZ + XYZ + W'XY'Z + XYZ' + WXY'Z' + W'XY'Z'$$

Simplify:
Gather similar terms and apply distribution. Note that the second term WXY'Z is used twice (idempotency: A+A=A).

$$= WXZ(Y+Y') + WX'Z(Y'+Y) + XY'Z(W+W') + XY'Z'(W+W') + XY(Z+Z')$$

Now apply complementation postulate to reduce A+A'=1 terms.

$$= WXZ + WX'Z + XY'Z + XY'Z' + XY$$

Gather similar terms and apply distribution.

$$= WZ(X+X') + XY'(Z+Z') + XY$$

Gather similar terms and apply distribution.

$$= WZ + XY' + XY = WZ + X$$ This is best one can do!!

Part B. Simplify the following Boolean SOP expression using the Karnaugh map technique. **[4marks]**

$$F(W,X,Y,Z) = SUM\ m(0,1,2,5,7,8,9,10,12,14,15)$$

Karnaugh Map is shown for the three 4-cubes (using wrap-around!!)

X'Z'	00	01	11	10
00	1	1		1
01		1	1	
11	1		1	1
10	1	1		1

WZ'	00	01	11	10
00	1	1		1
01		1	1	
11	1		1	1
10	1	1		1

X'Y'	00	01	11	10
00	1	1		1
01		1	1	
11	1		1	1
10	1	1		1

We identify the following N-cubes and consequent minterm expressions:
16-cube: None 8-cube: None
4-cube: 4 corners (X'Z'), bottom left/right pairs (WZ'), upper/lower left pairs (X'Y')
2-cube: This leaves only two in the centre, second row (W'XZ) and third column (XYZ)

Which yields the expression: $F(W,X,Y,Z) = X'Z' + WZ' + X'Y' + W'XZ + XYZ$

There may be other expressions, but those ones should all reduce to the one above or be equivalent. Students should note that the use of wraparound is easily proven to be useful and to arrive at greater reduction more quickly. I was deliberately evasive in the lecture about this issue and challenged the class to work through it. However, since I did state that one should not do this, many students did not try it and I have been pretty forgiving. It is still important to understand that even if one arrives at a reduced SOP expression using K-map reduction, it is necessary to inspect the resulting expression for further reduction – this point was made in the lecture.

Question 3 - continued

Part C. [4 marks]
Simplify the following Boolean SOP expression using the Karnaugh map technique. Don't care conditions are indicated by **dc()**.

F(W,X,Y,Z) = SUM m(1,2,5,8,9,12,14)

+ SUM dc(3,6,10,13,15)

Karnaugh Map

	00	01	11	10
00		1	dc	1
01		1		dc
11	1	dc	dc	1
10	1	1		dc

Which yields the expression:

= WZ′ + WY′ + WX + Y′Z + YZ′

There may be other expressions, but those ones should all reduce to the one above or be equivalent. Check these carefully. Note that using wraparound N-cubes, there are only 4-cubes in the K-map.

Question 4. [6 marks]
For this question, start by assuming two 4-bit inputs A and B (ie. each of A and B consists of 4 bits). Both A and B represent signed binary integers (2's complement form).

Design a combinational circuit that accepts inputs from A and B bits, and produces a single output bit called V. The output V = 1, if and only if the result of adding A+B produces an overflow (in the context of signed binary addition arithmetic); otherwise, V = 0. You are permitted to use the concepts of Half-Adder and Full-Adder, and also Comparator, as defined in the lecture notes, if you think these are relevant to this question.

Your answer must consist of an algebraic expression for V, and **also** you must justify your answer.

First, it is vital to correctly define what constitutes overflow and how to recognize it. We note that adding any valid positive number to any valid negative number results in a number that is a valid 2's complement representation – that is, such additions do not produce overflow.

This leaves two cases: adding two positive numbers or adding two negative numbers. By examining such cases, it is clear that overflow situations are recognizable by an absurd change of the sign (high order) bit. For instance, in a 4-bit representation, the largest positive number is 7 (0111). Now, add 1 (0001) to this to get 7+1=8. However, 8 in binary is just 1000. In a 2-s complement representation, this is the number -8, which is illogically absurd.

Thus, a circuit to detect overflow only needs to check the value of the sign bit of the inputs against the sign bit of the result. Denoting the high order bits of A and B inputs as A_H and B_H respectively, and of the sum high order bit as S_H, it follows that the overflow output V is defined by the circuit:

$$\mathbf{V = A_H' \, B_H' \, S_H + A_H \, B_H \, S_H'}$$

In order to produce the output $\mathbf{S_H}$, it is only necessary to use (or construct) a 4-bit signed binary adder, but we already did this in class and the high-order sum output is just $\mathbf{S_H}$. We just couple together a HA (low order bit inputs) and 3 x FA. Note that the adder Carry-Out is not the same as the Overflow V.

Question 5. [10 marks]
For this question, start by assuming two 4-bit inputs A and B (ie. each of A and B consists of 4 bits). Both A and B represent unsigned (ie. non-negative) binary integers.

The purpose of a Comparator unit is to numerically compare two inputs, A and B. The comparator unit has three outputs: E, L and G. If A > B, then G=1 while L and E are both 0. If A < B, then L=1 while G and E are both 0. Finally, if A==B, then E = 1 and L=G=0.

Part A. Design a Half-Comparator circuit that handles the initial comparison of the low-order bits of A and B inputs. Justify your answer. Express the circuit algebraically**. [3 marks]**

The purpose of this circuit is to initialize the values of G_0, L_0 and E_0. The truth table is just:

A_0	B_0	E_0	G_0	L_0
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Thus, the half-comparator circuit, or HC, is a 2-input, 3-output circuit with outputs defined by the expressions:

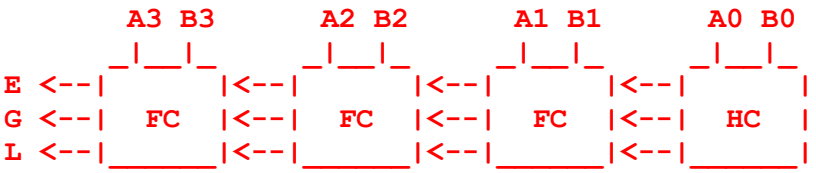
$G_0 = A_0B_0'$; $L_0 = A_0'B_0$; $E_0 = A_0B_0 + A_0'B_0'$ This defines the HC

Part B. Design a Full-Comparator circuit that handles the comparison of the K'th bits of A and B inputs. State all assumptions. Justify your answer. Express the circuit algebraically**. [5 marks]**

This was fully answered in the lecture notes. Refer to those for the full answer to the algebraic expressions for the circuit.

Part C. Using the results from **Part A** and **Part B** of this question, design a 4-bit-Comparator circuit that handles the comparison of the A and B 4-bit inputs. Express the circuit diagrammatically**. [2 marks]**

By combining the diagrams for HC and 3xFC comparators from Parts A and B above one obtains the full answer. On the high order bit side of the circuit there must be three outputs for G, L and E with only one of these allowed to be positive (1) for an unsigned binary input A and B, as described.



Question 6. [10 marks]
Express the six Axioms of Boolean Algebra. It is not required that you name each axiom, but you must be able to illustrate how each axiom is applicable in establishing Boolean Calculus. You must state all forms of each axiom, in cases where dual forms may apply.

Axioms/Postulates: (Review the lecture notes on this material) I have provided additional NOTEs along with each postulate. Students were required to provide either a basic statement or set of formulas, or a clear example to illustrate the postulate. Each postulate is very unique and students often do not appreciate the differences and therefore combine two or more and create confusion. Two good questions to ask when confronting new ideas are: (i) what does it mean/imply? AND, (ii) what does it NOT mean/imply? By the way, this is a dualistic approach.

P0: Existence - There exist at least two elements x,y in B such that $x \neq y$

NOTE: If you don't postulate the existence of values/elements, then you have nothing to work with operationally. This does not imply anything else about operations, actual values, etc.

P1: Closure: there exist x,y and operators (+,.) in B such that

$x + y$ in B $x \cdot y$ in B

NOTE: Like the existence of values, there must also be the existence of operators that *act* on the values to transform them. This does not imply what the operators do, only that there are only two of them. This is a necessary condition for the specification for an *algebra*.

P2: Identity: There exist identity elements 0,1 in B relative to the operations + and . , such that for every x in B:

$0 + x = x + 0 = x$ $1 \cdot x = x \cdot 1 = x$

NOTE: The existence of identity elements is meaningful only with respect to their associated operators; hence we must refer to 0 (1) as the identity element with respect to the operator + (.). The choice of symbol (0 or 1 character shapes) is not relevant, nor do these symbols represent a specific quantity of "something" – keep in mind that in practical circuits, both 0 and 1 "values" have a voltage value that is non-zero (zero voltage means the absence of electrical energy, where circuits do not work).

P3: Commutativity: The operations + and . are commutative for all x,y in B:

$x + y = y + x$ $x \cdot y = y \cdot x$

NOTE: It is intriguing that some algebras have a directionality to their operators – these are called *non-commutative algebras*, where X.Y is NOT equal to Y.X. The distinction to permit commutation on both + and . produces a very specialized algebra. Clearly this is not unique to Boolean algebra since we use it all the time with addition and multiplication of decimal and other numbers.

P4: Distributivity: Each operation + and . is distributive over the other; that is, for all x,y,z in B:

$x \cdot (y + z) = x \cdot y + x \cdot z$

$x + (y \cdot z) = (x + y) \cdot (x + z)$

NOTE: These dual forms are very powerful generalizations of the Distribution Law used in ordinary arithmetic and algebra of real numbers where only the first form applies. The extension to the dual form dramatically alters and limits the nature of algebraic expressions and results in significant simplification of otherwise complicated multinomial expressions.

P5: Complementation: For every element x in B there exists an element ~x, called the complement of x, satisfying:

$x + \sim x = 1$ $\qquad\qquad$ $x \cdot \sim x = 0$

NOTE: The notion of complementation (a generalization of *negation*) is a fundamentally significant idea in Computer Science. As noted here, it may be somewhat easier to state these forms using the ordinary language of sets. Thus, the first form states that, given a portion of a set (x), then creating the *union* of everything else that is not included in x, must yield the complete set with everything in it. This does assume, however, that both x and ~x are included in the full set (a point that is covered above in regard to uniqueness and the specification of the identity relations). The second form states that the *intersection* of a portion x of the set S, together with everything that is not contained in x, namely ~x, is the *empty set*. Strangely enough, this is where the language of set theory is not fully adequate to expose the subtleties of Boolean Set Theory and Algebra because the symbol 0, as the identity element defined over the operator +, is not actually a ***null*** value (ie. empty set). Note also, that both identity values 0 and 1 are associated with their dual operator on the left hand side.