University of Waterloo
Department of Electrical & Computer Engineering
E&CE 223 Digital Circuits and Systems
Midterm Examination
Instructor: M. Sachdev
October 30th, 2006

Total Time = 90 Minutes, Total Marks = 100

Student Name:			**Student ID:**			
1 **/10**	**2** **/20**	**3** **/20**	**4** **/25**	**5** **/15**	**6** **/10**	**Total**

Attempt all problems. Show all work. If information appears to be missing make a reasonable assumption, state it, and proceed. Calculators are not needed and are not allowed.

Problem 1 [10 Marks]

(a) Convert $(853)_{10}$ to BCD and 84-2-1 code. [3 Marks]

$(1000\ 0101\ 0011)_{BCD}$

$(1000\ 1011\ 0101)_{8\ 4\ -2\ -1}$

(b) What is the key feature of Gray Code and when should it be used? [3 Marks]

- *Only one bit in the code group changes when going from one number to the next.*
- *Used when sequence of binary numbers may produce an error or ambiguity during the transition from one number to the next. For example, gray code is utilized in measuring & sensing mechanical movement.*

(c) Convert $(101\ 0110\ 1000\ 0111\ 1001\ 0011.11011)_2$ to base 4 format. [4 Marks]

Grouping two bits at a time starting from either side of the decimal point and adding leading zero to the most significant and least significant bits give us:

$(1112\ 2013\ 2103.312)_4$

Problem 2 [20 Marks]

(a) What are three different ways of representing a signed number? Assume 8 bit numbers and represent (-15) in each of them. [4 Marks]

Signed & Magnitude:	*10001111*
Signed & 1's Complement	*11110000*
Signed & 2's Complement	*11110001*

(b) Calculate B-A using 1's complement subtraction. Both B and A are in unsigned binary format. Express your answer in sign and magnitude form. [8 Marks]

Given A=1101010 B=0110101

Take 1's complement for A: 0010101

0110101
0010101
1001010

The answer is in 1's complement form: 1001010

Signed and magnitude: -1110101 = $(-53)_{10}$

(c) Find $(-17)_{10} - (30)_{10}$ using 8-bit signed 2's complement form. Express you answer in signed 2's complement format. [8 Marks]

-17 = 10010001 (Signed & Magnitude)
= 11101111 (Signed & 2's Complement)

-30 = 10011110 (Signed & Magnitude)
= 11100010 (Signed & 2's Complement)

11101111
11100010
111010001

Ignore the carry bit, the final answer is 11010001

Problem 3 [20 Marks]

$F_1 = xyz' + wx'y' + (x'+z+w)(x'+z+w') + xyz + wx'y$
$F_2 = xy + wx' + x' + z$

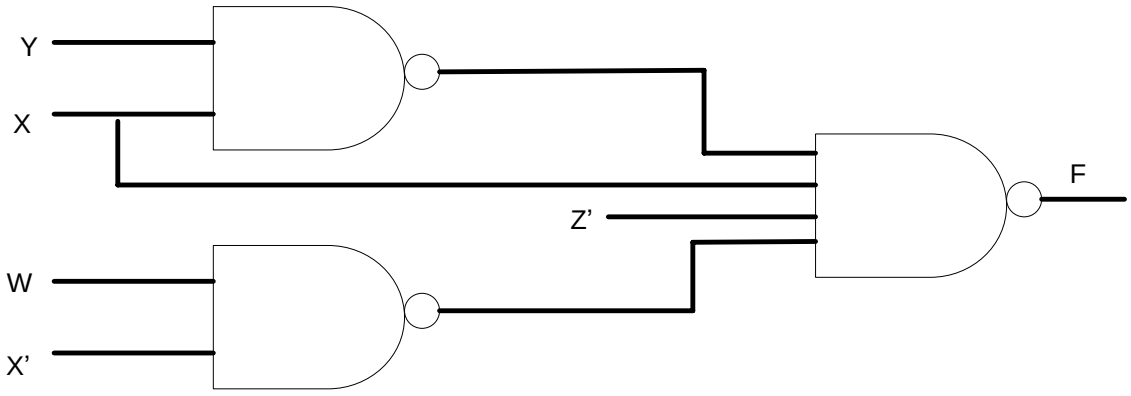
(a) **Do not use K-Map!** Show F_1 can be simplified to F_2 by algebraic means. [8 Marks]

$F_1 = xyz' + wx'y' + (x'+z+w)(x'+z+w') + xyz + wx'y$
$= xy(z + z') + wx'(y+y') + (x' + z) + (ww')$
$= xy + wx' + x' + z$

(b) Implement F_2 using NAND gates only. Assume all variables are available in both true and complement form [6 Marks]

$F_2' = (x'+y')(w'+x)xz'$
$= (xy)'(wx')'xz'$

$F_2 = [(xy)'(wx')'xz']'$



Problem 4 [25 Marks]

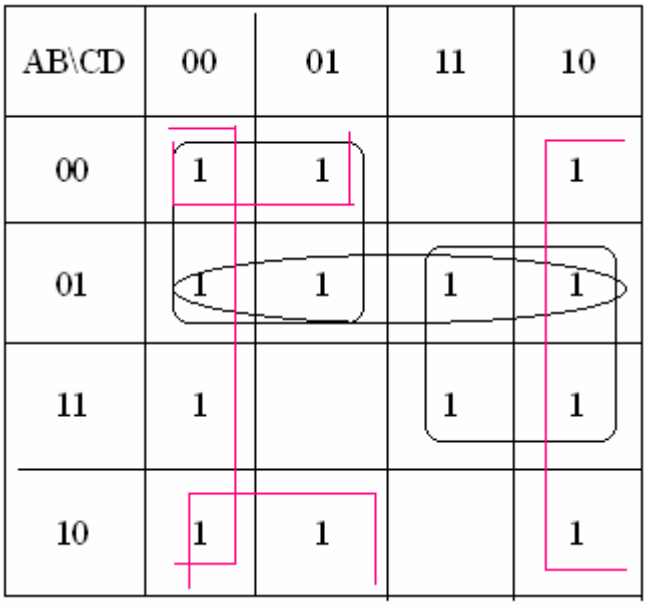
F (A, B, C, D) = ∏ (3, 11, 13)

(a) Express the above expression in canonical *sum of products* form. [4 Marks]

The sum of product expression for F will include the terms not present in the product of sum form. Therefore,

F (A, B, C, D) = ∑(0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15)

(b) List all the prime implicants and essential prime implicants [16 Marks]



F = D′ + A′B + B′C′ + BC or F = D′ + A′C′ + B′C′ + BC

Prime Implicants:

D′, A′B, BC, B′C′, A′C′

or

(0, 2, 4, 6, 8, 10, 12, 14), (4, 5, 6, 7), (6, 7, 14, 15), (0, 1, 8, 9), (0, 1, 4, 5)

Essential Prime Implicants:

D′, BC, B′C′

or

(0, 2, 4, 6, 8, 10, 12, 14), (6, 7, 14, 15), (0, 1, 8, 9)

	0	**1**	**2**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**12**	**14**	**15**
(0,2,4,6 8,10,12,14)	^		^	^		^		^		^	^	^	
(4,5,6,7)				^	^	^	^						
(6,7,14,15)						^	^					^	^
(01,8,9)	^	^						^	^				
(0,1,4,5)	^	^		^	^								

(c) P (A, B, C, D) = $\sum$ (0, 1, 3, 4, 6, 7, 8, 12, 14, 15)

Fill out the K-map provided below for expression P. Assume A is the most significant bit (MSB) and D is the least significant bit (LSB) [5 Marks]

CD\AB	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	1	1	1	0
10	0	1	1	0

Problem 5

Many offices and buildings use combination locks to control entry. As the design engineer of the *Wonderful Door Security Company*, you are asked to implement a door security system by using a card reader. There are four inputs to the card reader: inputs X, Y, and Z are used to validate the correct door code, and input V is used to check if the card reader is still valid. After the card reader is being read by the system, there are three outputs to this system: alarm (A), door open (D), and Error (E). Door (D) will only open when the decimal value of the binary inputs (x, y, z) is odd **AND** the card reader is valid. The Error (E) signal goes on when the code on the card is correct (i.e. decimal value equal to odd) but the card is no longer valid. Finally, the alarm (A) will trigger when the code is incorrect. Show your final design in canonical *product of sum* form.

x	y	z	V	A	D	E
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1
1	1	1	1	0	1	0

$$A(x, y, z, V) = \prod(2, 3, 6, 7, 10, 11, 14, 15)$$

$$D(x, y, z, V) = \prod(0, 1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14)$$

$$E(x, y, z, V) = \prod(0, 1, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15)$$

Problem 6

The critical path in a digital circuit is defined as the path through the logic that determines the ultimate speed of the structure. In the 4-bit ripple carry adder structure shown in **Figure 6-1**, the critical path in this circuit is the time that takes to generate the last carry out bit, $C_{o,3}$. **Figure 6-2** shows the implementation of each full adder used in the ripple carry adder.

If the delay for each logic gate is given in **Table-1**, calculate the total delay in the critical path of the 4-bit ripple carry adder. Assume all the inputs A_0-A_3, B_0-B_3, and $C_{i,0}$ are ready at t=0. **Provide justification for your answer.**

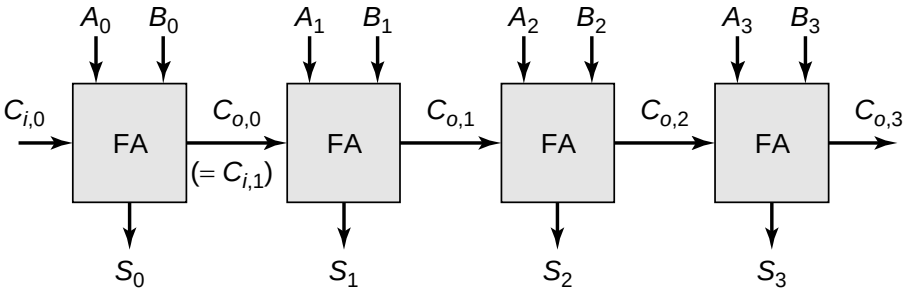


Figure-1: Ripple Carry Adder

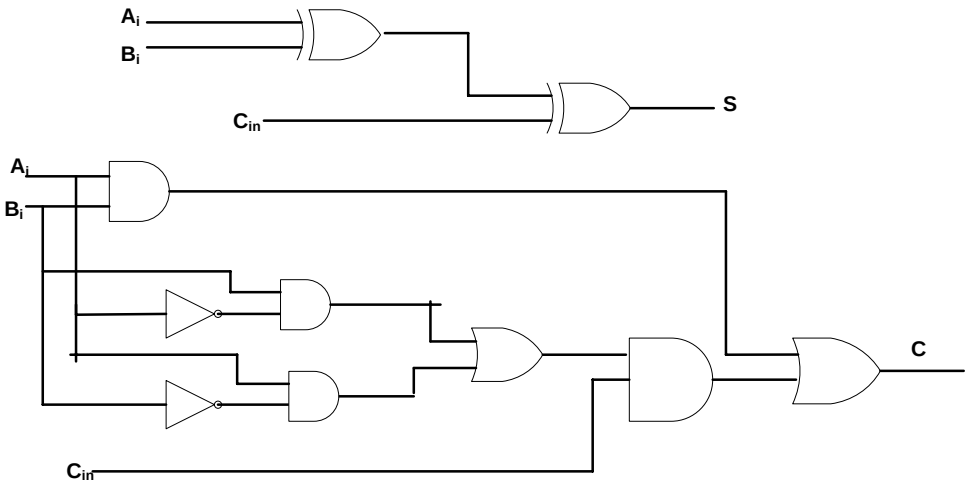


Figure-2: Full Adder Implementation

Table-1: Logic Gate Delays

	Delay (ns)
Inverter	1ns
AND	3ns
OR	5ns
XOR	9ns

1st FA: Delay = 1 + 3 + 5 + 3 + 5 = 17ns

2nd, 3rd, and 4th FA: Delay = 3 + 5 = 8ns

Total delay is 17 + 8 + 8 + 8 = 41ns