Digital Design and Computer Organization

July 2020

Unit I Question Bank

Table 1: Unit I Questions

Lecture	Topic	Text book Solved Examples	Exercise Problems
1	Introduction		Exercise 1.8,1.9, 1.10, 1.25 Question 2.1, Question 2.4
2	Boolean Functions	Example 2.1 Combinational circuits Example 2.2 Sum-of-products form Example 2.3 Product-of-sums form	Exercise 2.1, 2.2, 2.3, 1.53, 1.52
3	Boolean algebras	Example 2.4 Derive the product-of-sums form Example 2.5 Proving the consensus theorem Example 2.6 Equation minimization	Exercise 2.7, Exercise 2.9
4	Identities		
5-6		Logic minimization	Exercise 2.15, 2.16, 2.22, 2.23, 2.24
7-8	K maps	Example 2.9 Minimization of a three-variable function using a k-map Example 2.10 Seven-segment display decoder Example 2.11 Seven-segment display decoder with don't cares	Exercise 2.19 Question 2.2
9-12	Adder subtractor, Overflow	Example 1.9 Sign/magnitude numbers Example 1.11 Value of negative two's complement numbers Example 1.12 Adding two's complement numbers Example 1.13 Subtracting two's complement numbers Example 1.14 adding two's complement numbers with overflow Example 5.1 Ripple-Carry adder and Carry-Lookahead adder delay Example 5.2 Prefix adder delay	Exercise 1.33, 1.34, 1.35 Exercise 1.36,1.37,1.38 Exercise 5.1

1 Exercise Problems (Problem statements)

1.1 What is the largest unsigned 32-bit binary number?

1.2 What is the largest 16-bit binary number that can be represented With,a) unsigned numbers? b) two's complement numbers? c) sign/magnitude numbers

1.3 What is the smallest (most negative) 16-bit binary number that can be represented with,a) unsigned numbers?. b) two's complement numbers? c) sign/magnitude numbers?

1.4 How many 5-bit two's complement numbers are greater than 0?. How many are less than 0? How would your answers differ for sign/magnitude numbers?

1.5 Sketch a schematic for the two-input XOR function using only NAND gates. How few can you use?.

1.6 A gate or set of gates is universal if it can be used to construct any Boolean function. For example, the set AND, OR, NOT is universal. (a) Is an AND gate by itself universal? Why or why not? (b) Is the set OR, NOT universal? Why or why not? (c) Is a NAND gate by itself universal? Why or why not?

1.7 Write a Boolean equation in sum-of-products canonical form for each of the truth tables

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

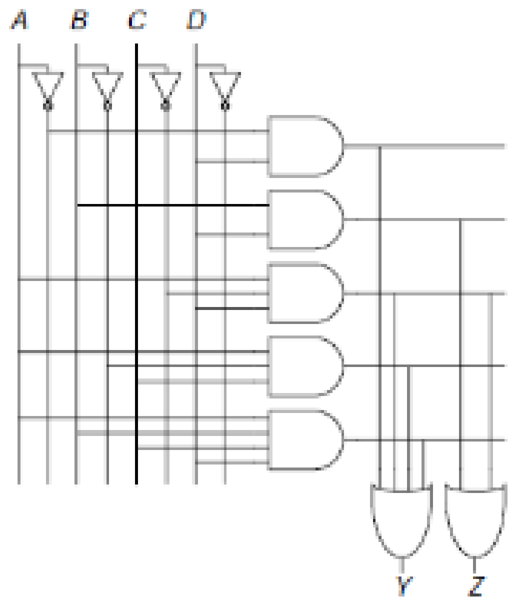
1.8 Write a Boolean equation in product-of-sums canonical form for the truth tables in question 7.

1.9 Minimize each of the Boolean equations from the equations obtained in the question 7.

1.10 Simplify the following Boolean equations using Boolean theorems. Check for correctness using a truth table or K-map. a) Y=AC+A'B'C b)Y= A'B'+A'BC'+ (A+C)' c) Y=A'B'C'D'+AB'C'+ ABD+A'B'CD'+BC'D+A'

1.11 Simplify each of the following Boolean equations. Sketch a reasonably simple combinational circuit implementing the simplified equation. a)Y=BC + (AB)'C' + BC' b) Y=(A + A'B + (AB)')' + (A+B')' c) Y= ABC+ ABD+ABE+ACD+ ACE+(A+D+E)'+B'C'D

1.12 Write Boolean equations for the circuit in Figure given below. You need not minimize the equations



1.13 **Minimize the Boolean equations for the question and sketch an improved circuit with the same function.**

1.14 **Find a minimal Boolean equation for the function in truth table given below. Remember to take advantage of the don't care entries.**

Truth Table

A	B	C	D	Y
0	0	0	0	X
0	0	0	1	X
0	0	1	0	X
0	0	1	1	0
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X
1	1	1	1	1

1.15 **Ben Bitdiddle will enjoy his picnic on sunny days that have no ants. He will also enjoy his picnic any day he sees a hummingbird, as well as on days where there are ants and ladybugs. Write a Boolean equation for his enjoyment (E) in terms of sun (S), ants (A), hummingbirds (H), and ladybugs (L).**

1.16 **Complete the design of the seven-segment decoder segments Sc through Sg . (a) Derive Boolean equations for the outputs Sc through Sg assuming that inputs greater than 9 must produce blank (0) outputs. (b) Derive Boolean equations for the outputs Sc through Sg assuming that inputs greater than 9 are don't cares. (c) Sketch a reasonably simple gate-level implementation of part (b). Multiple outputs can share gates where appropriate.**

1.17 **A circuit has four inputs and two outputs. The inputs, A3:0, represent a number from 0 to 15. Output P should be TRUE if the number is prime (0 and 1 are not prime, but 2, 3, 5, and so on, are prime). Output D should be TRUE if the number is divisible by 3. Give simplified Boolean equations for each output and sketch a circuit**

1.18 **Design a circuit that will tell whether a given month has 31 days in it. The month is specified by a 4-bit input, A3:0. For example, if the inputs are 0001, the month is January, and if the inputs are 1100, the month is December. The circuit output, Y, should be HIGH only when the month specified by the inputs has 31 days in it. Write the simplified equation, and draw the circuit diagram using a minimum number of gates. (Hint: Remember to take advantage of don't cares.)**

1.19 Perform the following additions of unsigned binary numbers. Indicate whether or not the sum overflows a 4-bit result. a) 1001_2 + 0100_2 (b) 1101_2 +1011_2

1.20 Perform the following additions of unsigned binary numbers. Indicate whether or not the sum overflows an 8-bit result. (a) 10011001_2 + 01000100_2 (b) 11010010_2 +10110110_2

1.21 Perform the following additions of unsigned binary numbers. Indicate whether or not the sum overflows an 8-bit result. (a) 10011001_2 + 01000100_2 (b) 11010010_2 + 10110110_2.Note that the binary numbers are in two's complement form.

1.22 Convert the following decimal numbers to 6-bit two's complement binary numbers and add them. Indicate whether or not the sum overflows a 6-bit result. a) 16_{10} + 9_{10} (b) 27_{10} + 31_{10} (c) -4_{10} + 19_{10} d) 3_{10} + -32_{10} e) -16_{10} + -9_{10} f) -27_{10} + -31_{10}

1.23 Perform the following additions of unsigned hexadecimal numbers. Indicate whether or not the sum overflows an 8-bit (two hex digit) result.(a) 7_{16} + 9_{16} (b) 13_{16} +28_{16} (c) AB_{16} + $3E_{16}$ (d) $8F_{16}$ + AD_{16}

1.24 Convert the following decimal numbers to 5-bit two's complement binary numbers and subtract them. Indicate whether or not the difference overflows a 5-bit result. a) 9_{10} - 7_{10} (b) 12_{10} -15_{10} (c) -6_{10} - 11_{10} (d) 4_{10} – (-8_{10})

1.25 How many different truth tables exist for Boolean functions of N variables?

1.26 There are 16 different truth tables for Boolean functions of two variables. List each truth table. Give each one a short descriptive name (such as OR, NAND, and so on).

1.27 **What is the delay for the following types of 64-bit adders? Assume that each two-input gate delay is 150 ps and that a full adder delay is 450 ps. (a) a ripple-carry adder (b) a carry-lookahead adder with 4-bit blocks (c) a prefix adder**

Additional questions (These questions have been taken from Digital Design, Morris Mano and Michael Ciletti)

2 Boolean Functions

2.1 Find the complement of the functions F1 = x'yz'+ x'y'z and F2 = x(y'z' + yz). By applying DeMorgan's theorems as many times as necessary

2.2 Express the Boolean function F = A + B'C as a sum of minterms. The function has three variables: A, B, and C

2.3 Express the Boolean function F = xy + x'z as a product of maxterms.

3 Identities

3.1 Demonstrate the validity of the following identities by means of truth tables: (a) DeMorgan's theorem for three variables: (x + y + z)' = x'y'z' and (xyz)' =x' + y' + z' (b) The distributive law: x + yz = (x + y)(x + z) (c) The distributive law: x(y + z) = xy + xz (d) The associative law: x + (y + z) = (x + y) + z (e) The associative law and x(yz) = (xy)z

4 Boolean Expressions

4.1 Simplify the following Boolean expressions to a minimum number of literals: (a) xy + xy' (b) (x + y) (x + y') (c) xyz + x 'y + xyz' (d) (A + B)'(A'+ B')' (e) (a + b + c')(a' b' + c) (f) a'bc + abc'+ abc + a'bc'

5 K-maps

5.1 Simplify the Boolean function: F (x, y, z) $=\sum(2,3,4,5)$

5.2 Simplify the Boolean function: F (x, y, z) $=\sum(0,2,4,5,6)$

5.3 Simplify the following Boolean function into (a) sum-of-products form and (b) product-of-sums form: F (A, B, C, D) $=\sum(0,1,2,5,8,9,10)$

5.4 Simplify the Boolean function F (w, x, y, z) $=\sum(1,3,7,11,15)$ which has the don't-care conditions d (w, x, y, z) $=\sum(0,2,5)$

5.5 Simplify the Boolean function by finding all its prime implicants and essential prime implicants: F(A, B, C, D) $=\sum(0,2,3,5,7,8,9,10,11,13,15)$

5.6 Simplify the following Boolean expressions, using four-variable maps: (a) A'B'C'D'+ AC'D'+ B'CD'+ A'BCD + BC'D (b) x'z + w'xy'+ w(x'y + xy') (c) w'z + xz + x'y + wx'z (d) AD'+ B'C'D + BCD' + BC'D

5.7 Simplify the following Boolean functions by first finding the prime and essential prime implicants: (a) F(w, x, y, z) $=\sum(0,2,5,7,8,10,12,13,14,15)$ (b) F(A, B, C, D) $=\sum(0,2,3,5,7,8,10,11,14,15)$ (c) F(A, B, C, D) $=\sum(1,3,4,5,10,11,12,13,14,15)$ (d) F(w, x, y, z) $=\sum(0,1,4,5,6,7,9,11,14,15)$

6 Adder subtractor, overflow

6.1 Perform subtraction on the given unsigned binary numbers using the 2's complement of the subtrahend. Where the result should be negative, find its 2's complement and affix a minus sign. (a) 10011 – 10010 (b) 100010 – 100110 (c) 1001 – 110101 (d) 101000 – 10101

6.2 Convert decimal +49 and +29 to binary, using the signed-2's-complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+29) + (-49), (-29) + (+49), and (-29) + (-49). Convert the answers back to decimal and verify that they are correct.

6.3 **Implement the following four Boolean expressions with three half adders: D = A** $\oplus B \oplus C, E = A'BC + AB'C, F = ABC' + (A' + B')C, G = ABC$

6.4 **A majority circuit is a combinational circuit whose output is equal to 1 if the input variables have more 1's than 0's. The output is 0 otherwise. (a)Design a 3-input majority circuit by finding the circuit's truth table, Boolean equation, and a logic diagram.**

6.5 **Using four half-adders (a) Design a full-subtractor circuit incrementer. (A circuit that adds one to a four-bit binary number.) (b) Design a four-bit combinational decrementer (a circuit that subtracts 1 from a four bit binary number**