CSE3015 Homework 1

이주헌 (20191629)

March 20, 2024

1. What is different between synchronous sequential systems and asynchronous sequential systems? Give examples how the circuits operate.

Sequential systems are dependent not only on current set of inputs, but also on previous history of inputs. Thus, creating a sequential system usually entails implementation of *states*. The type of a sequential system—whether it is synchronous or asynchronous—is determined by the timing of state changes:

- if the system's state changes *only* at discrete times, then it is a **synchronous sequential system**;
- and if the state can be updated at *any time*, then it is a **asynchronous sequential system**.

Thus, synchronous sequential systems depend on a centralized clock to receive inputs and process them in a timely fashion. One example of such systems is Central Processing Units in computer systems. On the other hand, asynchronous sequential systems do not require a centralized clock, and are capable of receiving inputs at any point in time. One example of asynchronous systems would be some two-way syncronizing communication system, where the sender can send signals at any time and the receiver responds with acknowledgement signal.

Question 2.

A. Convert the following unsigned binary integers to decimal. If the following numbers are decimal integers, convert them to binary.

   (a) $1101011_2 \Rightarrow 107_{10}$
   (b) $110101101_2 \Rightarrow 429_{10}$
   (c) $1110010_2 \Rightarrow 114_{10}$
   (d) $825_{10} \Rightarrow 1100111001_2$
   (e) $514_{10} \Rightarrow 1000000010_2$

B. Convert the following to hexadecimal. If the following numbers are hexadecimal, convert them to decimal.

   (a) $101110010001011_2 \Rightarrow 5C8B_{16}$
   (b) $111001111110111_2 \Rightarrow 73F7_{16}$
   (c) $1110011010110_2 \Rightarrow 1CD6_{16}$
   (d) $3B2A_{16} \Rightarrow 15146$
   (e) $FEA9_{16} \Rightarrow 65193$

3. Compute the sum of the following pairs of 6-bit unsigned integers. If the answer is to be stored in a 6-bit location, indicate which of the sums produce overflow. Also, show the decimal equivalent of both operands and the result.

a. $000111_2 + 010111_2 = 011110_2 \Leftrightarrow 7_{10} + 23_{10} = 30_{10}$; This operation **would not** cause overflow.

b. $110000_2 + 010010_2 = 1000010_2 \Leftrightarrow 48_{10} + 18_{10} = 66_{10}$; This operation **would** cause overflow.

c. $100011_2 + 100111_2 = 1001010_2 \Leftrightarrow 35_{10} + 39_{10} = 74_{10}$; This operation **would** cause overflow.

d. $010011_2 + 101111_2 = 1000010_2 \Leftrightarrow 19_{10} + 47_{10} = 66_{10}$; This operation **would** cause overflow.

e. $010011_2 + 011100_2 = 101111_2 \Leftrightarrow 19_{10} + 28_{10} = 47_{10}$; This operation **would not** cause overflow.

4. The following decimal integers are to be stored in a 6-bit two' s complement format. Show how they are stored.

a. $+15_{10} \Rightarrow 001111_2$

b. $-13_{10} \Rightarrow 110011_2$

c. $0_{10} \Rightarrow 000000_2$

d. $-32_{10} \Rightarrow 100000_2$

e. $32_{10} \not\Rightarrow 0100000_2$; 32 cannot be represented with signed 6-bit two's compliment binary format.

5. The following 6-bit two' s complement integers were found in a computer. What decimal number do they represent?

a. $111001_2 \Rightarrow -7_{10}$

b. $100110_2 \Rightarrow -26_{10}$

c. $111111_2 \Rightarrow -1_{10}$

d. $011011_2 \Rightarrow +27_{10}$

e. $010110_2 \Rightarrow +22_{10}$

6. Each of the following pairs of signed (two' s complement) integers are stored in computer words (6 bits). Compute the sum as it is stored in a 6-bit computer word. Show the decimal equivalents of each operand and the sum. Indicate if there is overflow.

a. $111010_2 + 000111_2 = 1000001_2 \Leftrightarrow -6_{10} + 7_{10} = 1_{10}$; This operation **would not** cause overflow.

b. $101010_2 + 100110_2 = 1010000_2 \Leftrightarrow -22_{10} + -26_{10} = 16_{10}$; This operation **would** cause overflow.

c. $111001_2 + 110001_2 = 1101010_2 \Leftrightarrow -7_{10} + -15_{10} = -22_{10}$; This operation **would not** cause overflow.

d. $101100_2 + 101100_2 = 1011000_2 \Leftrightarrow -20_{10} + -20_{10} = 24_{10}$; This operation **would** cause overflow.

e. $100110_2 + 001100_2 = 110010_2 \Leftrightarrow -26_{10} + 12_{10} = -14_{10}$; This operation **would not** cause overflow.

7. Determine, using truth tables, which expressions in each of the groups are equal.

a. $f = bc' + b'c + ac$
$g = (a + c)(a' + b + c')$

a	b	c	f	g
T	T	T	T	T
T	T	F	T	T
T	F	T	T	F
T	F	F	F	T
F	T	T	F	T
F	T	F	T	F
F	F	T	T	T
F	F	F	F	F

Since f and g produce different truth table, $f \neq g$.

b. $f = a'b + ac' + a'bd'$
$g = ad' + a'bc + a'bd'$

a	b	c	d	f	g
T	T	T	T	F	F
T	T	T	F	F	T
T	T	F	T	T	F
T	T	F	F	T	T
T	F	T	T	F	F
T	F	T	F	F	T
T	F	F	T	T	F
T	F	F	F	T	T
F	T	T	T	T	T
F	T	T	F	T	T
F	T	F	T	T	F
F	T	F	F	T	T
F	F	T	T	F	F
F	F	T	F	F	F
F	F	F	T	F	F
F	F	F	F	F	F

Since f and g produce different truth table, $f \neq g$.

8. Simplify to contain the smallest number of literals. Show each step and justification.

a. $x'z + xy'z + xyz$

Steps	Justifications
$x'z + xy'z + xyz = x'z + (xy')z + (xy)z$	Associative law
$= (x' + xy')z + (xy)z$	Distributive law
$= (x' + xy' + xy)z$	Distributive law
$= (x' + (xy' + xy))z$	Associative law
$= (x' + x)z$	Adjacency law
$= 1 \cdot z$	Complement law
$= z$	Identity of logical AND

b. $x'y'z' + x'y'z + xy'z + xyz'$

Steps	Justifications
$x'y'z' + x'y'z + xy'z + xyz' = (x'y')z' + (x'y')z + xy'z + xyz'$	Associative law
$= x'y' + xy'z + xyz'$	Adjacency law
$= x'y' + xzy' + xyz'$	Commutative law
$= (x' + xz)y' + xyz'$	Distributive law
$= (x' + z)y' + xyz'$	Simplification law
$= x'y' + y'z + xyz'$	Distributive law

c. $(x + y + z)(x + y + z')(x + y' + z)(x + y' + z')$

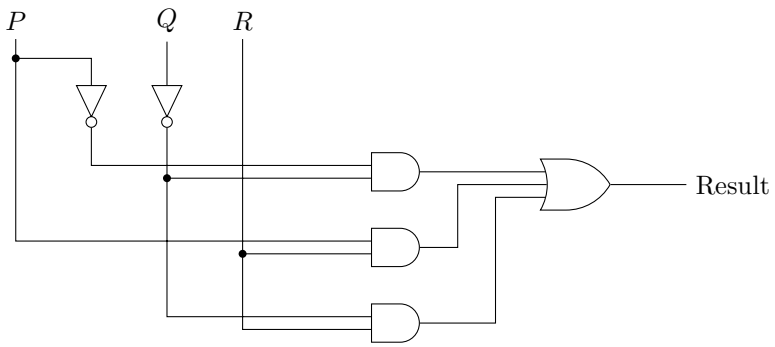
Steps	Justifications
$(x + y + z)(x + y + z')(x + y' + z)(x + y' + z')$	Given
$= ((x + y) + z)((x + y) + z')(((x + y') + z)((x + y') + z'))$	Associative law
$= (x + y)(x + y')$	Adjacency law
$= x$	Adjacency law

d. $(a + b + c)(a + b' + c)(a + b' + c')(a' + b' + c')$

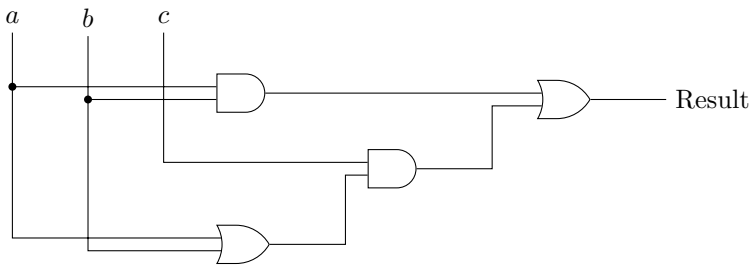
Steps	Justifications
$(a + b + c)(a + b' + c)(a + b' + c')(a' + b' + c')$	Given
$= (a + c + b)(a + c + b')(b' + c' + a)(b' + c' + a')$	Commutative law
$= ((a + c) + b)((a + c) + b')(((b' + c') + a)((b' + c') + a'))$	Associative law
$= (a + c)(b' + c')$	Adjacency law
$= (a + c)b' + (a + c)c'$	Distributive law
$= ab' + b'c + ac' + cc'$	Distributive law
$= ab' + b'c + ac' + 0$	Null of logical AND
$= ab' + b'c + ac'$	Identity of logical OR
$= b'c + ac' + ab'$	Commutative law
$= b'c + ac'$	Consensus law

9. Show a block diagram of a system using AND, OR, and NOT gates to implement the following functions. Assume that variables are available only uncomplemented. Do not manipulate the algebra.

a. $P'Q' + PR + Q'R$

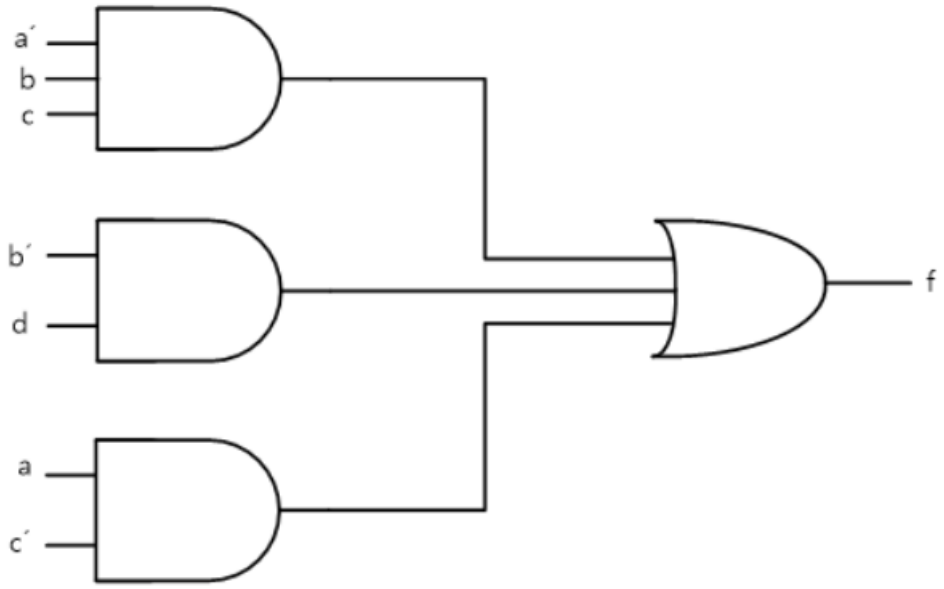


b. $ab + c(a + b)$



10. For each of the following circuits, (i) find an algebraic expression, (ii) put it in sum of product form.

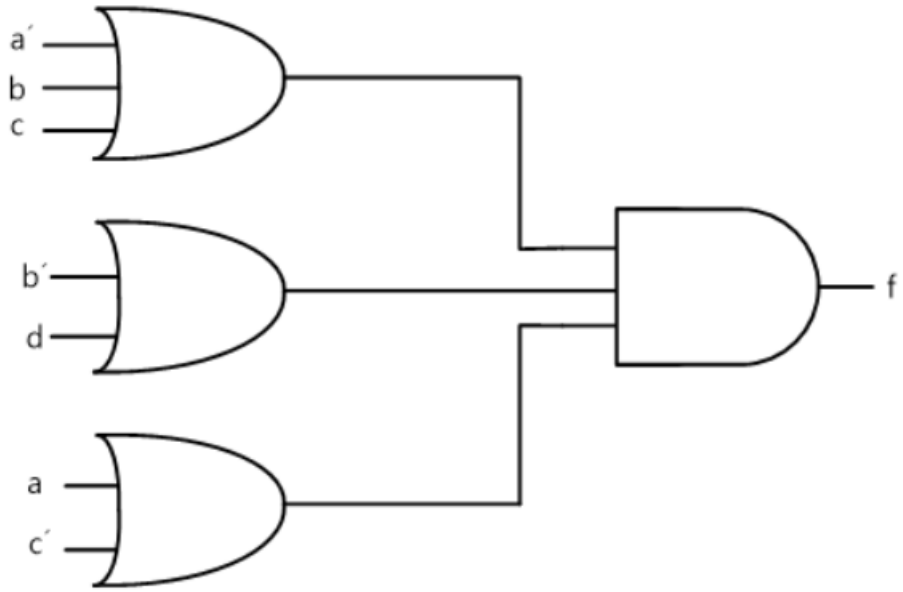
a.



(i). $a'bc + b'd + ac'$

(ii). $a'bc + b'd + ac'$; this expression is already in *sum of product* form.

b.



(i). $(a' + b + c)(b' + d)(a + c')$

(ii). $(a' + b + c)(b' + d)(a + c')$

$= (a'b' + a'd + bb' + bd + b'c + cd)(a + c')$
$= (a'b' + a'd + bd + b'c + cd)(a + c')$
$= aa'b' + a'b'c' + aa'd + a'c'd + abd + bc'd + ab'c + b'cc' + acd + cc'd$
$= a'b'c' + a'c'd + abd + bc'd + ab'c + acd$
$= a'b'c' + ab'c + abd + bc'd$