Digital Logic Design (CSNB163)

Module 5

Combinations of Binary Variables

- In Module 3, you have learned the concept of Boolean Algebra which consists of binary variables and binary operator.
- A binary variable *x*, can either appear in its:
  - normal form ***(x)*** or
  - complement form ***(x')***
- If there are two binary variables (e.g. *x* and *y*) beings the inputs to a binary operator (e.g. AND), then there are basically 2^2 number of possible combinations (e.g. *xy, x'y, xy', x'y'*),
- Thus, $\mathbf{2^n}$ possible combinations where $\boldsymbol{n}$ is the total number of input variable.

Boolean Algebra Representation

- A Boolean Algebraic expression can appear in **any of the following forms**:
  - **standard** form
    - Sum of Product (SOP)
    - Product of Sum (POS)
  - **canonical** form
    - Sum of Minterms
    - Product of Maxterms

Standard Vs Canonical Form of Boolean Algebra

- The major difference between standard and canonical form of Boolean Algebra expression is in terms of the **input binary variables between terms**.

Criteria	Standard Form	Canonical Form
input binary variables between terms	Not necessarily consistent	Must be consistent
Example	F(a, b, c) = a + b'c	F(a, b, c) = a'b'c + ab'c + ab'c' + a'b'c

Standard Boolean Algebra (Sum of Product)

- Sum of Product (SOP) is a Boolean expression **containing AND terms** (also known as 'product term') whereby the AND terms may contain **one or more input binary variables** and are **combined by OR** operations
- Example:

contain AND terms with one or more input binary variables

$$F(x, y, z) = xy' + y'z' + xyz$$

AND terms are combined through OR operations

Standard Boolean Algebra (Product of Sum)

- Product of Sum (POS) is a Boolean expression **containing OR terms** (also known as 'sum term') whereby the OR terms may contain **one or or more input binary variables** and are **combined by AND** operations
- Example:

contain OR terms with one or more input binary variables

$$F(x, y, z) = (x+y') \,.\, (y'+z') \,.\, (x+y+z)$$

AND terms are combined through AND operations

Canonical Boolean Algebra (Sum of Minterms)

- Sum of Minterms is a Boolean expression **containing AND terms** (also known as '**minterms**') whereby the minterms **must** contain **similar input binary variables** and are **combined by OR** operations
- Example:

contain minterms with similar input binary variables

$$F(x, y, z) = xy'z' + xy'z + xyz$$

minterms are combined through OR operations

Canonical Boolean Algebra (Sum of Minterms cont.)

- Each input binary variable in a minterm can be associated with a binary value, whereby:
  - **normal form = 1** (e.g. x = 1)
  - **complement form = 0** (e.g. x’ = 0)
- E.g. If given F1:

  $F1 = xy'z' + xy'z + xyz$

  the equivalence

  $F1 = m4 + m5 + m7$
  $= \sum (m4, m5, m7)$

- Thus, **output** is only associated with binary **1**

x	y	z	minterm	designation	F1
0	0	0	*x’y’z’*	*m0*	0
0	0	1	*x’y’z*	*m1*	0
0	1	0	*x’yz’*	*m2*	0
0	1	1	*x’yz*	*m3*	0
1	0	0	*xy’z’*	*m4*	1
1	0	1	*xy’z*	*m5*	1
1	1	0	*xyz’*	*m6*	0
1	1	1	*xyz*	*m7*	1

Canonical Boolean Algebra (Sum of Minterms) (Example1)

- Given the following truth table:

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

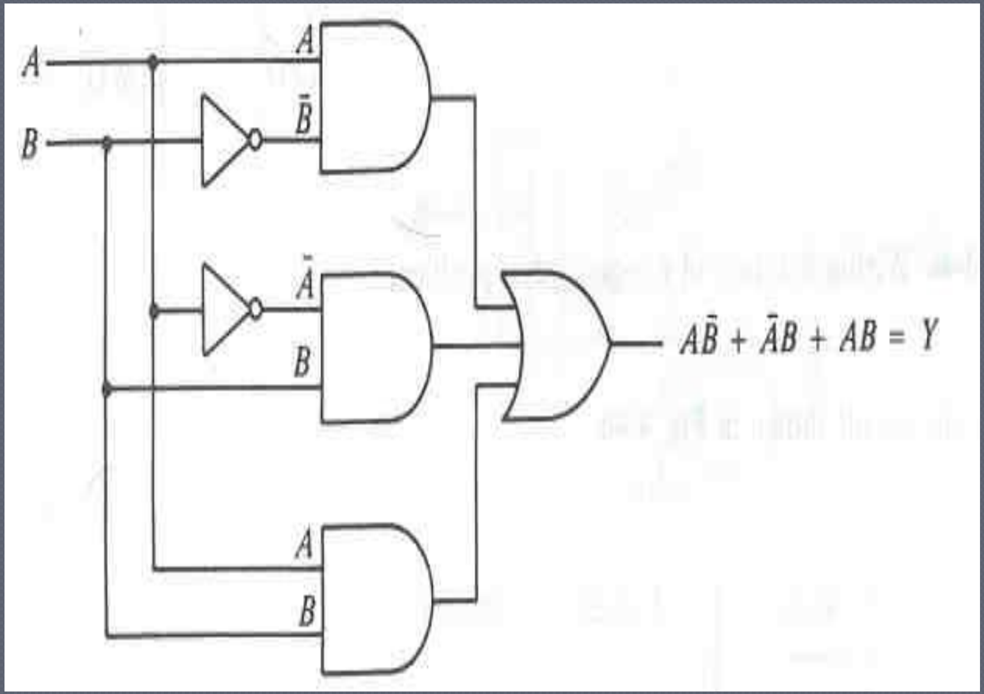
  - Derive the Sum of Minterms Boolean expression for Y.
  - Draw the circuit diagram.

Canonical Boolean Algebra (Sum of Minterms) (Example1)

- Answer =

$$Y = A'B + AB' + AB$$

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



Standard Boolean Algebra (Product of Maxterms)

- Product of Maxterms is a Boolean expression **containing OR terms** (also known as **maxterms**) whereby the **maxterms** must contain **similar input binary variables** and are **combined by AND** operations
- Example:

contain maxterms with similar input binary variables

$$F2 = (x+y+z')\,(x+y'+z')\,(x'+y'+z)$$

maxterms are combined through AND operations

Canonical Boolean Algebra (Product of Maxterms cont.)

- Each input binary variable in a maxterm can be associated with a binary value, whereby:
  - **normal form = 0** (e.g. x = 0)
  - **complement form = 1** (e.g. x' = 1)
- If given F2:

$F2 = (x+y+z')\,(x+y'+z')\,(x'+y'+z)$

the equivalence

$F2 = M1 \,.\, M3 \,.\, M6$
$\quad = \prod(M1, M3, M6)$

- Thus, **output** is only associated with binary **0**

x	y	z	maxterm	designation	F2
0	0	0	$x+y+z$	$M0$	1
0	0	1	$x+y+z'$	$M1$	0
0	1	0	$x+y'+z$	$M2$	1
0	1	1	$x+y'+z'$	$M3$	0
1	0	0	$x'+y+z$	$M4$	1
1	0	1	$x'+y+z'$	$M5$	1
1	1	0	$x'+y'+z$	$M6$	0
1	1	1	$x'+y'+z'$	$M7$	1

Canonical Boolean Algebra (Product of Maxterms) (Example1)

- Given the following truth table:

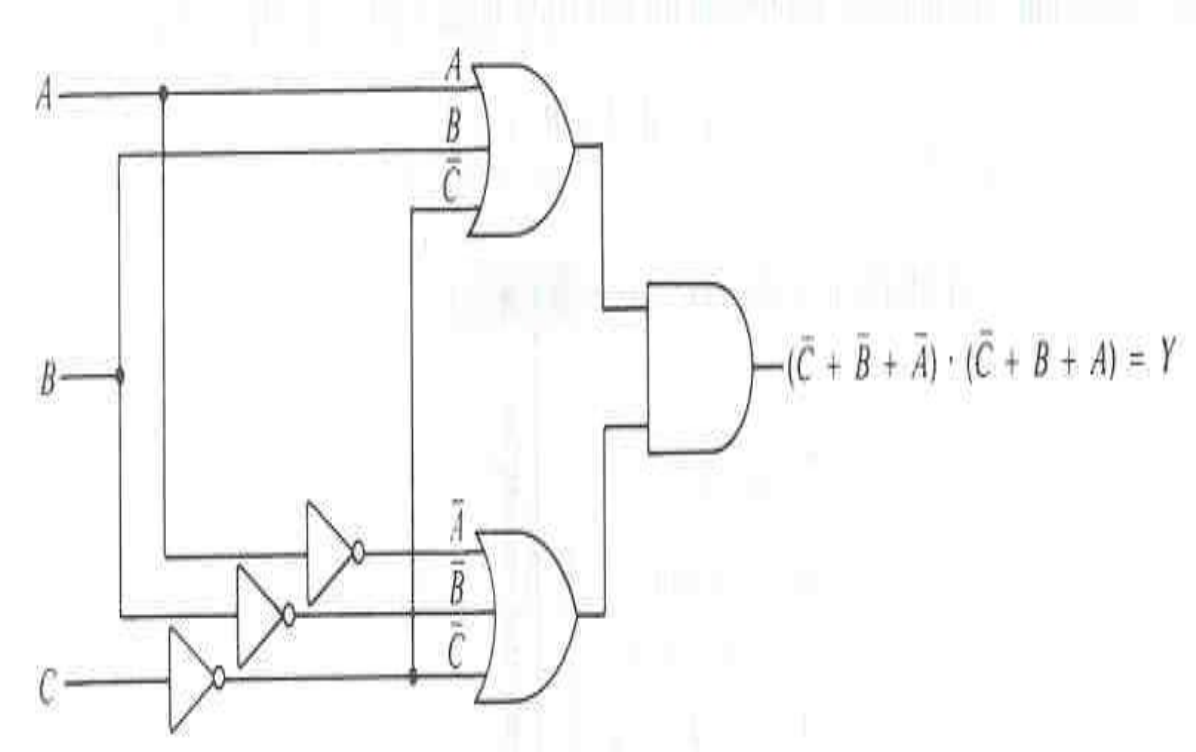
C	*B*	*A*	*Y*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

  - Derive the Product of Maxterms Boolean expression for Y.
  - Draw the circuit diagram.

Canonical Boolean Algebra (Product of Maxterms) (Example1)

- Answer =

$Y = (C' + B + A)\,(C' + B' + A')$



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

Sum of Minterms – Product of Maxterms

- An Algebraic Boolean expression in Sum of Minterms can be converted into its Product of Maxterms (and vice versa)

Sum of Minterms ⟷ Product of Maxterms

This can be proven through Theorems of Boolean Algebra!

Sum of Minterms – Product of Maxterms (Example1)

- Given the following truth table:

x	*y*	*Z*	*Y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

  - Derive the Sum of Minterms Boolean expression for Y.
  - Derive the Product of Maxterms Boolean expression for Y.
  - Prove that both expressions are equal.

Sum of Minterms – Product of Maxterms (Example1)

- Answer:

Sum of Minterms

$Y1 = m3 + m5 + m6 + m7$

$= x'yz + xy'z + xyz' + xyz$

Product of Maxterms

$Y2 = M0.M1.M2.M4$

$= (x+y+z)(x+y+z') (x+y'+z) (x'+y+z)$

x	y	z	Y	m	M
0	0	0	0	m0	M0
0	0	1	0	m1	M1
0	1	0	0	m2	M2
0	1	1	1	m3	M3
1	0	0	0	m4	M4
1	0	1	1	m5	M5
1	1	0	1	m6	M6
1	1	1	1	m7	M7

Sum of Minterms – Product of Maxterms (Example1)

- Answer:

x	y	z	Y	m	M
0	0	0	0	m0	M0
0	0	1	0	m1	M1
0	1	0	0	m2	M2
0	1	1	1	m3	M3
1	0	0	0	m4	M4
1	0	1	1	m5	M5
1	1	0	1	m6	M6
1	1	1	1	m7	M7

Proving:
Sum of Minterms =
Product of Maxterms

Y1' = m0+m1+m2+m4

Y1 = (Y1')' = (m0+m1+m2+m4)'

Y1 = M0.M1.M2.M4 = Y2

Y1' = m0+m1+m2 +m4 = x'y'z'+x'y'z+x'yz'+xy'z'

Y1=(Y1')'= (x'y'z'+x'y'z+x'yz'+xy'z')' via Involution

Y1=(x'y'z')' . (x'y'z)' . (x'yz')' . (xy'z')' via DeMorgan's

Y1=(x''+y''+z'').(x''+y''+z').(x''+y'+z'').(x'+y''+z'') via DeMorgan's

Y1=(x+y+z).(x+y+z').(x+y'+z).(x'+y+z)=Y2 via Involution

Conversion between Boolean Algebra Representations

- We can **convert** standard form of Boolean Algebraic expression into canonical form (and vice versa).
- For example:
  - **Convert Sum of Product into sum of minterms** (and vice versa)
  - **Convert Product of Sum into product of maxterms** (and vive versa)
- Conversion is carried out by manipulating the basic postulates, properties and theorems of Boolean Algebra taught in Module 4.

Convert Standard Form into Canonical Form (Example1)

- Convert the following Boolean Function

$$F(a, b, c) = a + b'c$$

to its canonical Sum of Minterms form.

$a + b'c$	
$= a(1) + b'c$	via Identity Element
$= a(b + b') + (b'c)$	via Complement
$= ab + ab' + b'c$	via Distributive
$=(ab + ab')(1) + (b'c)(1)$	via Identity Element
$=(ab + ab')(c+c') + b'c(a + a')$	via Complement
$=abc + abc' + ab'c + ab'c' + ab'c + a'b'c$	via Distributive
$=abc + abc' + ab'c + ab'c' + a'b'c$	via Idempotency

Convert Standard Form into Canonical Form (Example2)

- Convert the following Boolean Function

$$F(x, y, z) = xy + x'z$$

to its canonical Product of Maxterms form.

(1st step – get the Sum of Minterms)

$F(x, y, z) = xy + x'z$	
$= xy(1) + x'z(1)$	via Identity Element
$= xy(z+z') + x'z(y+y')$	via Complement
$= xyz + xyz' + x'yz + x'y'z$	via Distributive
$= m7 + m6 + m3 + m1$	**Sum of Minterms**

Convert Standard Form into Canonical Form (Example2)

- Convert the following Boolean Function

$F(x, y, z) = xy + x'z$

to its canonical Product of Maxterms form.

(2nd step – convert Sum of Minterms into Product of Maxterms)

$F(x,y,z) = m7+m6+m3+m1$ **Sum of Minterms**

Thus $F(x, y, z)' = m5+m4+m2+m0 = xy'z + xy'z' + x'yz' + x'y'z'$

$F(x,y,z) = \{F(x, y, z)'\}'$

$= (xy'z + xy'z' + x'yz' + x'y'z')'$ via Involution

$= (xy'z)' . (xy'z')' . (x'yz')' + (x'y'z')'$ via DeMorgan's

$= (x'+y''+z').(x'+y''+z'').(x''+y'+z'').(x''+y''+z'')$ via DeMorgan's

$= (x'+y+z').(x'+y+z).(x+y'+z).(x+y+z)$ via Involution

$= M5 + M4 + M2 + M0$ **Product of Maxterms**

Digital Logic Design (CSNB163)

End of Module 5