Circuit Design

behaviour (truth table)

↓

logic gates

(Boolean expression, Minimization)

EE2026 (Part 1)
Tutorial 4 - Questions

Boolean Algebra and Minimization

$A=0 \Leftrightarrow \bar{A}$

A	B	C	minterm
0	0	0	$\bar{A}\cdot\bar{B}\cdot\bar{C}$
0	0	1	$\bar{A}\cdot\bar{B}\cdot C$
0	1	0	$\bar{A}\cdot B\cdot\bar{C}$
0	1	1	$\bar{A}\cdot B\cdot C$
1	0	0	$A\cdot\bar{B}\cdot\bar{C}$
1	0	1	$A\cdot\bar{B}\cdot C$
1	1	0	$A\cdot B\cdot\bar{C}$
1	1	1	$A\cdot B\cdot C$

input value making minterm = 1

One-to-One mapping

$\boxed{A|B|C} \Leftrightarrow$ minterm $=1$

$\boxed{\bar{A}|\bar{B}|\bar{C}} \Leftrightarrow$ maxterm $=0$

$A=1 \Leftrightarrow \bar{A}$

A	B	C	maxterm
0	0	0	$A+B+C$
0	0	1	$A+B+\bar{C}$
0	1	0	$A+\bar{B}+C$
0	1	1	$A+\bar{B}+\bar{C}$
1	0	0	$\bar{A}+B+C$
1	0	1	$\bar{A}+B+\bar{C}$
1	1	0	$\bar{A}+\bar{B}+C$
1	1	1	$\bar{A}+\bar{B}+\bar{C}$

input value making maxterm = 0

Convert Truth Table → Canonical Form: SOP (CSOP)

- In n-variable function, there are 2^n different minterms and maxterms
  - Example with n=3: sum of 4 products (CSOP)

A	B	C	F	minterm	maxterm
0	0	0	0	$\bar{A}\cdot\bar{B}\cdot\bar{C}$	$A+B+C$
0	0	1	0	$\bar{A}\cdot\bar{B}\cdot C$	$A+B+\bar{C}$
0	1	0	1	$\boxed{\bar{A}\cdot B\cdot\bar{C}}$	$A+\bar{B}+C$
0	1	1	0	$\bar{A}\cdot B\cdot C$	$A+\bar{B}+\bar{C}$
1	0	0	1	$\boxed{A\cdot\bar{B}\cdot\bar{C}}$	$\bar{A}+B+C$
1	0	1	1	$\boxed{A\cdot\bar{B}\cdot C}$	$\bar{A}+B+\bar{C}$
1	1	0	0	$A\cdot B\cdot\bar{C}$	$\bar{A}+\bar{B}+C$
1	1	1	1	$\boxed{A\cdot B\cdot C}$	$\bar{A}+\bar{B}+\bar{C}$

minterm: basic building block. for F=1 in truth table

$$F=\bar{A}\cdot B\cdot\bar{C}+A\cdot\bar{B}\cdot\bar{C}+$$
$$+A\cdot\bar{B}\cdot C+A\cdot B\cdot C$$

any minterm = 1, F = 1.

if no minterm = 1, F = 1

- Truth table is expressed directly with canonical form by
  1. Identifying minterms = 1 for the input values making F = 1
  2. (Logically) add them up

Convert Truth Table → Canonical Form: POS (CPOS)

maxterm: basic building block for F=0.

- Example with n=3: product of 3 sums (CPOS)

A	B	C	F	minterm	maxterm
0	0	0	0	$\bar{A}\cdot\bar{B}\cdot\bar{C}$	$\boxed{A+B+C}$
0	0	1	0	$\bar{A}\cdot\bar{B}\cdot C$	$\boxed{A+B+\bar{C}}$
0	1	0	1	$\bar{A}\cdot B\cdot\bar{C}$	$A+\bar{B}+C$
0	1	1	0	$\bar{A}\cdot B\cdot C$	$\boxed{A+\bar{B}+\bar{C}}$
1	0	0	1	$A\cdot\bar{B}\cdot\bar{C}$	$\bar{A}+B+C$
1	0	1	0	$A\cdot\bar{B}\cdot C$	$\boxed{\bar{A}+B+\bar{C}}$
1	1	0	0	$A\cdot B\cdot\bar{C}$	$\boxed{\bar{A}+\bar{B}+C}$
1	1	1	1	$A\cdot B\cdot C$	$\bar{A}+\bar{B}+\bar{C}$

any maxterm = 0, F = 0

no maxterm = 0, F = 1

$$F=(A+B+C)\cdot(A+B+\bar{C})\cdot$$
$$\cdot(A+\bar{B}+\bar{C})\cdot(\bar{A}+B+\bar{C})\cdot$$
$$\cdot(\bar{A}+\bar{B}+C)$$

each maxterm is 0 only for a specific input (no "interference" with other inputs) → just multiply maxterms independently

- Truth table is expressed directly with canonical form by
  1. Identifying maxterms = 0 for the input values making F = 0
  2. (Logically) multiply them

Two-variable K-map:
(maximum 4 minterms)

$m_0 \to 00 \to \bar A\,\bar B$
$m_1 \to 01 \to \bar A\,B$
$m_2 \to 10 \to A\,\bar B$
$m_3 \to 11 \to AB$

squares

A \ B	$\bar B$	B
$\bar A$	$\bar A\bar B$	$\bar A B$
A	$A\bar B$	AB

A \ B	0	1
0	00	01
1	10	11

"0" → Literal **with** overbar
"1" → Literal **without** overbar

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

Compress
Truth Table ⟶ K-map.

A·B. Cols ⟶ A row
B col.

A	B	F
0	0	0
0	1	0
1	0	1
1	1	1

A \ B	0	1
0	0	0
1	1	1

Three-variable K-map

A \ BC	$\bar B\bar C$	$\bar B C$	BC	$B\bar C$
$\bar A$	$\bar A\bar B\bar C$	$\bar A\bar B C$	$\bar A BC$	$\bar A B\bar C$
A	$A\bar B\bar C$	$A\bar B C$	ABC	$AB\bar C$

A \ BC	00	01	11	10
0	000	001	011	010
1	100	101	111	110

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Four-variable K-map

AB \ CD	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

$$F = \bar A B + AB\bar C + \bar A\bar B C$$

$$\bar A B = \bar A B(C + \bar C) = \bar A BC + \bar A B\bar C$$

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	0	1

Or $\bar A B \to 01$, $C = 0$ *or* 1

$$F = A + \bar A\bar B CD + B\bar C\bar D$$

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

SoP → MSoP : group "1"s

Simplify: $F = \bar A B + AB + \bar A\bar B$

A \ B	0	1
0	1	1
1	0	1

2-cell group

$\bar A B + \bar A\bar B = \bar A(B + \bar B) = \bar A$

common literal: remains

literals by which they differ: eliminated

$AB + \bar A B = B(\bar A + A) = B$

common literal: remains

literals by which they differ: eliminated

MSOP:

$F = \bar A + B$

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

$\bar B$

Assume X = 1

group as big as possible.
can overlap.

Grouping: looking for the unchanged Boolean expression

Don't care X.
place holder.
can be 1.

two variable change value

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

$\bar B\bar C\bar D$ BD CD

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

$\bar B\bar D$ BD

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

only in 2^n

Squares in the group are not in power of two

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

not horizontal or vertical

☆ no boundary.
all connected.

(POS → MPOS : group "0"s . X can be 0.)

1. A switching circuit has four inputs as shown. A and B represent the MSB and LSB bits of a binary number N_1. C and D represent the MSB and LSB bits of a binary number N_2. The output is to be 1 only if the product N_1 x N_2 is less than or equal to 2. decimal.

(a) Write the truth table for the system.

(b) Write the canonical SOP and POS expressions for F.

(c) Draw a Karnaugh-map (K-map) for the function F.

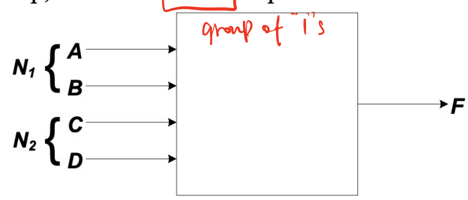
(d) From the K-map, derive the MSOP expression for F.

$N_1 = (AB)_2$

$N_2 = (CD)_2$

if $N_1 \times N_2 \leq 2$: $F = 1$

else: $F = 0$.



Truth Table

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$1 \times 3 > 2$.

$2 \times 2 > 2$

$2 \times 3 > 2$

$3 \times 1 > 2$

$3 \times 2 > 2$

$3 \times 3 > 2$

$X = 0 \Leftrightarrow \bar{X}$
verify minterm

$F=1$

$$Z_{SOP} = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + AB\bar{C}\bar{D}$$

$F=0$

$$Z_{POS} = (A + \bar{B} + \bar{C} + \bar{D}).(\bar{A} + B + \bar{C} + D).(\bar{A} + B + \bar{C} + \bar{D}).(\bar{A} + \bar{B} + C + \bar{D}).(\bar{A} + \bar{B} + \bar{C} + D).(\bar{A} + \bar{B} + \bar{C} + \bar{D})$$

$X = 0 \Leftrightarrow X$.
verify maxterm

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

simplify (save logic gates)

$$Z_{MSOP} = \bar{A}\bar{B} + \bar{C}\bar{D} + \bar{A}\bar{C} + \bar{A}\bar{D} + \bar{B}\bar{C}$$

groups with 2^n "1"s.

2. A bank vault has three locks with a different key for each lock. Each key is owned by a different person. To open the door, at least two people must insert their keys into the assigned locks. The signal lines A, B and C are 1 if there is a key inserted into lock 1, 2 or 3, respectively. Write an equation for the variable Z which is 1 if and only if the door should open.

(a) Write the truth table for the system.
(b) Write the canonical SOP and POS expressions for Z.
(c) Draw a Karnaugh-map (K-map) for the function Z.
(d) From the K-map, derive the MSOP expression for Z.

Truth Table

A	B	C	F
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 ✓

011

F=1 $Z_{SOP} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

F=0. $Z_{POS} = (A + B + C).(A + B + \bar{C}).(A + \bar{B} + C).(\bar{A} + B + C)$

0 0 0

BC \ A	0	1
00	0	0
01	0	1
11	1	1
10	0	1

2^n.

$Z_{MSOP} = AC + BC + AB$

SOP: $X=1 \leftrightarrow X$
$X=0 \leftrightarrow \bar{X}$.

3. Use K-maps to obtain an MSOP and an MPOS for each of the following functions:

(a) $Z = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D}$ with don't care for $ABCD$ = 1010

(b) $Z = (\bar{A} + B + \bar{C})(A + B + \bar{C})$ with don't cares for ABC = 111 and 110

(c) $f(x_1,\ldots,x_4) = \sum m(0, 4, 5, 6, 7) + D(1, 12, 13, 14, 15)$, where $m(\,)$ is the set of minterms for which f = 1 and $D(\,)$ is the set of don't cares. For example, $m(2)$ is the minterm corresponding to $x_1x_2x_3x_4$ = 0010 (this alternate shorthand notation is often used to express SOPs).

(a) $Z = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D}$ with 'X' for $ABCD$ = 1010

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

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

MSOP

$Z = \bar{C}D + A\bar{B}\bar{C}$

group of 4 group of 2

MPOS

$Z = \bar{C}(A + D)\cdot(\bar{B} + D)$

group of 8 group of 4 group of 4

$X = 1 \leftrightarrow \bar{X}$

$X = 0 \leftrightarrow X$.

Verify using C = 1.

(b) $Z = (\bar{A} + B + \bar{C})(A + B + \bar{C})$ with don't cares for ABC = 111 and 110

0 0.

all the rest are 1.

BC \ A	0	1
00	1	1
01	0	0
11	1	X
10	1	X

MSOP

$Z = B + \bar{C}$

BC \ A	0	1
00	1	1
01	0	0
11	1	X
10	1	X

MPOS

$Z = (B + \bar{C})$

to binary!

0000 0100 0101 0110 0111

(c) $f(x_1,\ldots,x_4) = \sum m(0,4,5,6,7) + D(1,12,13,14,15)$, where D is the set of don't cares and m is the set for which f = 1 (this alternate shorthand notation is also used to express min terms).

X_3X_4 \ X_1X_2	00	01	11	10
00	1	1	X	0
01	X	1	X	0
11	0	1	X	0
10	0	1	X	0

X_3X_4 \ X_1X_2	00	01	11	10
00	1	1	X	0
01	X	1	X	0
11	0	1	X	0
10	0	1	X	0

MSOP

$Z = (x_2 + \bar{x}_1\bar{x}_3)$

MPOS

$Z = (x_2 + \bar{x}_3)\cdot(\bar{x}_1)$

4. A combinational circuit has four inputs A, B, C and D and one output Z. The output is asserted whenever three or more of the inputs are asserted, otherwise the output is de-asserted. Find an MSOP expression for Z. Design the combinational circuit using only inverters, 2-input NAND gates and 3-input NAND gates. Assume that A, B and Z are active high signals, while C and D are active low signals. Use alternate gate representations for clarity of circuit diagrams. Calculate the critical path delay.

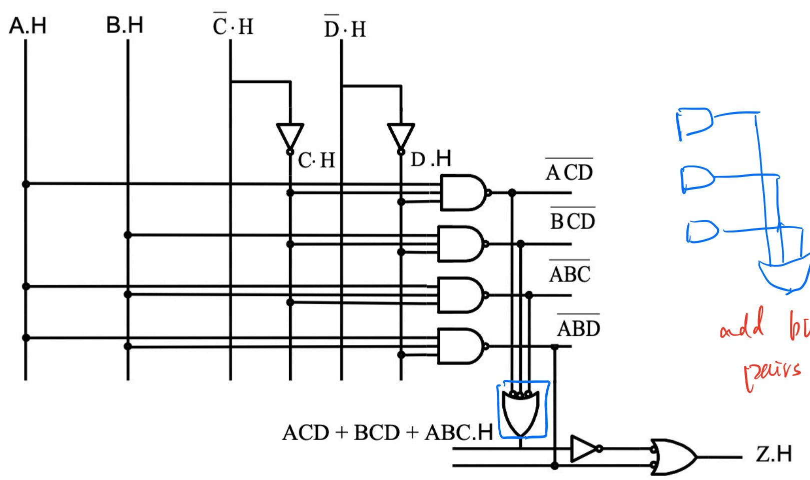
Gate	t_{pd}
NOT	0.3ns
2-input NAND	0.5ns
3-input NAND	0.8ns

ABCD
1110
1101
1011
0111
1111

Z = 1

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

MSOP
Z = ACD+ BCD+ ABC+ ABD



add bubble pairs later

Z.H = (ACD + BCD + ABC + ABD).H

Critical Path Delay
= 0.3 (NOT) + 0.8 (NAND3) + 0.8 (NAND3) + 0.3 (NOT) + 0.5 (NAND2)
= 2.7ns

longest path (pass through the most gates)

$\overline{C}\cdot H \longrightarrow Z\cdot H$