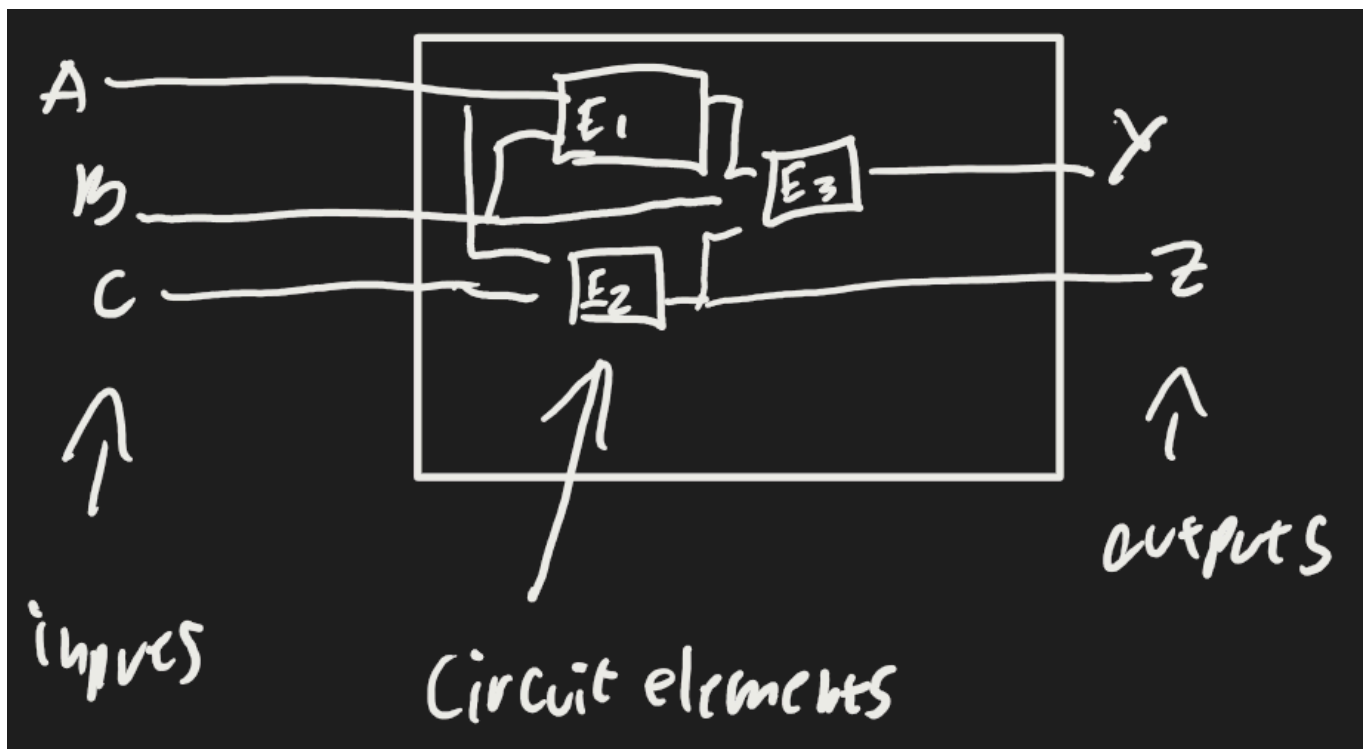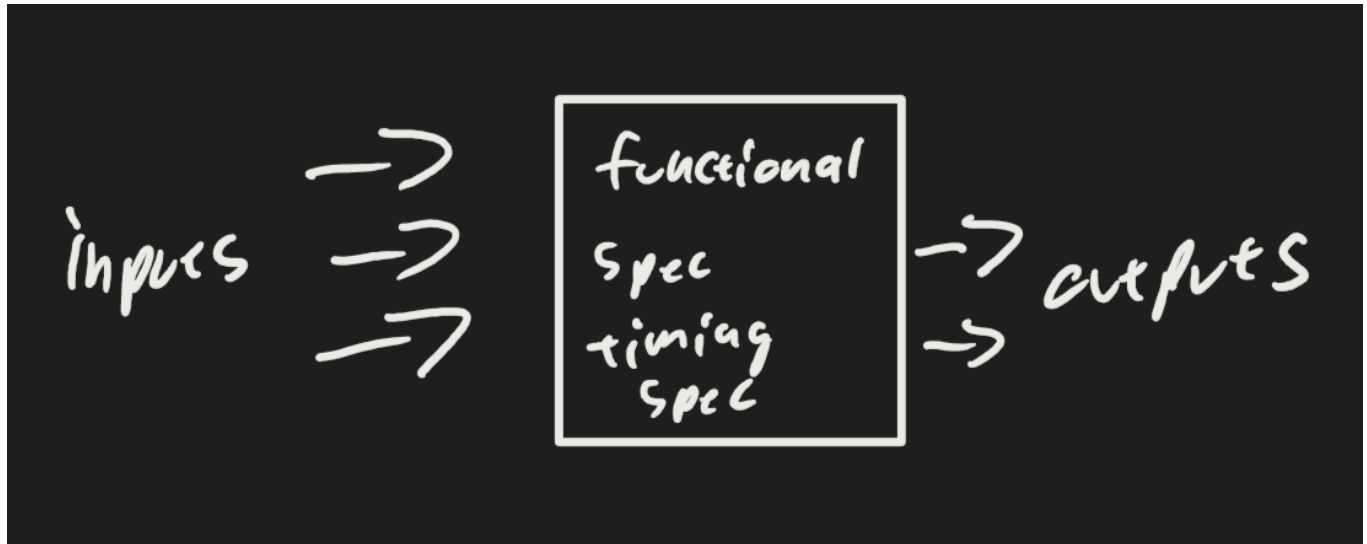# 3 Boolean Logic

Combinatorial Circuits

Combinational circuits don't have any memory, so they only depend on the current inputs.

There are two types of logic circuits: Combinatorial and Sequential.

- No memory
- Current output is a function of current input

To build a combinational circuit, any sub circuit also has to be combinational.

Every node is either an input or connects to exactly one output
There are no cyclic paths (no output that goes back into an element).

- Has memory
- Outputs are a function of previous and current inputs
-

Boolean Equations

This is the description language for our functional specifications. We can use Boolean algebra to simplify our circuits. There are often tradeoffs between speed of the circuit and number of chips.

$$S = F(A, B, C_{in}) = A \oplus B \oplus C_{in}$$

$$C_{out} = F(A, B, C_{in}) : AB + AC_{in} + BC_{in}$$

(C_t means combinational logic).

$\oplus$ means exclusive or, i.e. one or the other but not both.
$+$ means or, i.e. at least 1
AB means A and B.

We have some terms that are commonly used:

- Compliment:
    - $\bar{A}$ means not A
- Literal
    - $A, \bar{A}, B, \bar{B}$
- Implicant
    - $AB\bar{C}$
- Min term
    - $ABC, A\bar{B}\bar{C}$
    - We take every input variable and 'and' them together
- Max term
    - $(A + \bar{B} + c), (\bar{A} + B + \bar{C})$

- we take every input variable, invert them, and 'or' them together

Let's build a truth table to explore these:

$$\begin{bmatrix} A & B & Y & \text{min term} & \text{max term} \\ 0 & 0 & 0 & \bar{A}\bar{B} & A+B \\ 0 & 1 & 1 & \bar{A}B & A+\bar{B} \\ 1 & 0 & 0 & A\bar{B} & \bar{A}+B \\ 1 & 1 & 1 & AB & A+B \end{bmatrix}$$

Our expression for this is the sum of the terms that output 1, so for tihs we would get the boolean expression $Y = F(A, B) = \bar{A}B + AB$. We can write this as combining the minterm 1 and minterm 3 (we zero index), so this is $\sum(1, 3)$. We just or all the terms together, although this is the least compact way of writing our system.

We could also combine our maxterms, for everywhere that it is false, so here it would be $(A + B)(\bar{A} + B) = \Pi(0, 2)$

However, we can see that there is another valid function for this table: $Y = F(B) = B$.

Let's practice with a new table.

$$\begin{bmatrix} C & M & E \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

For min terms this gets
$E = C\bar{m} = \sum(2)$.

For max terms, this gets

$$E = (C + M)(C + \bar{M})(\bar{C} + \bar{M})$$
$$= \Pi(1, 2, 4)$$

Another example:

You can Eat delicious food (**E** is the output) if you
Make it yourself (**M**) or you have a personal Chef (**C**)
and she/he is talented (**T**) but not eXpensive ($\bar{X}$).

This is described by $E = M + CT\bar{X}$

Sometimes we have terms that don't do anything, i.e.
$E = HS + H$ is equivalent to $E = H$.

We start with a problem, make a Boolean expression, and implement it in gates.

Boolean Algebra

We can use Boolean algebra to simplify expressions.
Everything is either a 0 or 1.

There is a notion of duality in minterms vs maxterms - they swap and with or and swap 0 with 1. Those are duals of each other.

> 📋 **Definition: Boolean Axioms** ⌄
>
> $B = 0 \, if \, B \neq 1$
> $\bar{0} = 1$
> $0 \cdot 0 = 0$
> $1 \cdot 1 = 1$
> $1 \cdot 0 = 0 \cdot 1 = 0$
>
> Each of these has a Dual, so axiom 1 has $B = 1$ if $B \neq 0$.
> We continue writing the duels (in the same order as above):
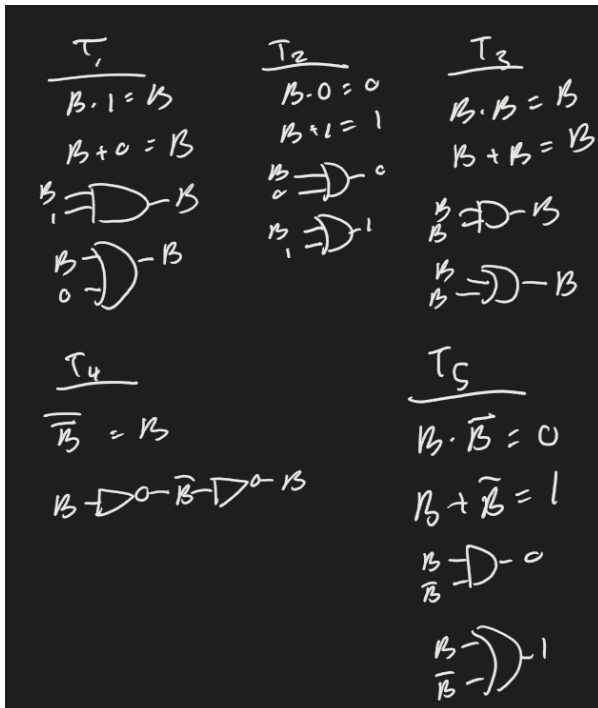
We have Boolean theorems of one variable

$$
\begin{bmatrix}
T_1 & B \cdot 1 = B & \text{identity} \\
T_2 & B \cdot 0 = 0 & \text{Null Element} \\
T_3 & B \cdot B = B \text{ dempotency} \\
T_4 & \bar{\bar{B}} & \text{involution} \\
T_5 & B \cdot \bar{B} = 0 & \text{compliments}
\end{bmatrix}
$$

For these, we have the Dual expressions (in order):

$$
\begin{bmatrix}
T_1 & B + 0 = B \\
T_2 & B + 1 = 1 \\
T_3 & B + B = B \\
T_4 & \bar{\bar{B}} = B \\
T_5 & B + \bar{B} = 1
\end{bmatrix}
$$

We can draw each theorem with gates.

#	Theorem	Dual	Name
T6	B•C = C•B	B+C = C+B	Commutativity
T7	(B•C) • D = B • (C•D)	(B + C) + D = B + (C + D)	Associativity
T8	B • (C + D) = (B•C) + (B•D)	B + (C•D) = (B+C) (B+D)	Distributivity
T9	B • (B+C) = B	B + (B•C) = B	Covering
T10	(B•C) + (B•$\overline{C}$) = B	(B+C) • (B+$\overline{C}$) = B	Combining
T11	(B•C) + ($\overline{B}$•D) + (C•D) = (B•C) + ($\overline{B}$•D)	(B+C) • ($\overline{B}$+D) • (C+D) = (B+C) • ($\overline{B}$+D)	Consensus

Warning: T8' differs from traditional algebra:
OR (+) distributes over AND (•)

We can now start to simplify expressions.

$$B \cdot (B + C)$$

We can write this laboriously and make a truth table and then find an expression and show they are equivalent, or we can use our theorems.

This would simplify to

$$B \cdot (B + C)$$
$$= BB + BC$$
$$= B + BC$$
$$= B \cdot 1 + B \cdot C$$
$$= B(1 + C)$$
$$= B$$

We have multiple methods for simplification. For example, De Morgan's Theorem.

$$\overline{(AB)} = \bar{A} + \bar{B}$$

$$X = \overline{(A + \overline{BD}) \cdot \bar{C}}$$

$$Y = (A + \overline{BD}) + \overline{\cdot C}$$

$$= \bar{A} \; \overline{(\overline{BD})} + C$$

$$= \bar{A} BD + C$$