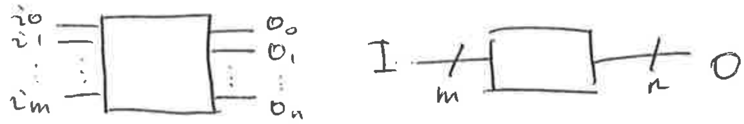
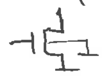
Computation structures 2014

S3
- ☐ recap
- ☐ Boolean algebra and combinational devices
  - two input gates
  - multiple input gates
  - algebra and simplification

• Recap

- we have learned how to represent 1s and 0s in a circuit.
- A combinational device can be written as



- It must satisfy voltage transfer disciplines
- ~~capacitor effect delays~~
- Two simplest devices NFET  PFET  inverter by nature

buffer

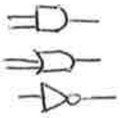
I	O
0	0
1	1

inverter

I	O
0	1
1	0

→ How do we build arbitrary combinational logic devices?
- buffer and inverter ✓
- more inputs
- more outputs

? Combination of gates
- AND
- OR
- NOT



Math first, and then hardware.

• algebra

- British mathematician George Bool

1850s "the mathematics of logic"

[domain] {0, 1} representing {False, True} values.

[Boolean functions] (because of this, binary algebra is also called logic, or Boolean algebra)

- The subject of Boolean algebra
- takes one or more inputs, gives one output.

e.g.

NOT

I	O
0	1
1	0

AND

I_1	I_0	O
0	0	0
0	1	0
1	0	0
1	1	1

NAND, NOR, XOR

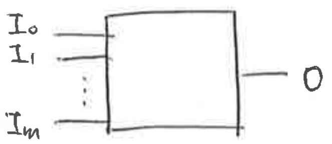
OR

I_1	I_0	O
0	0	0
0	1	1
1	0	1
1	1	1

2^2 rows, 2^{2^2} functions.
why are AND and OR and NOT special?
they are universal.

- The above tables are called <u>Truth tables</u>; they enumerate all possible inputs and define the output. All Boolean functions can be exhaustively described using truth table.

- Boolean functions are naturally connected to M-to-1 device



each defice is such a Boolean function, and can be described using a truth table

- [M-to-N] devices can be treated as N M-to-1 devices in parallel, although there may be better solutions.

- [M-to-1] devices can be simplified as a combination of 2-1 and 1-1 devices.

- Given a truth table, it is easy to find a Boolean function.

Note how enumerated

	~~I_7 I_6 I_5 I_4~~	I_3	I_2	I_1	I_0	O
0	- - - -	0	0	0	0	0
1	- - - -	0	0	0	1	0
2	- - - -	0	0	1	0	0
3	- - - -	0	0	1	1	0
4	- - - -	0	1	0	0	0
5	- - - -	0	1	0	1	0
6	- - - -	0	1	1	0	0
7	- - - -	0	1	1	1	1
8	- - - -	1	0	0	0	0
9	- - - -	1	0	0	1	0
10	- - - -	1	0	1	0	0
11	- - - -	1	0	1	1	0
12	- - - -	1	1	0	0	0
13	- - - -	1	1	0	1	1
14	- - - -	1	1	1	0	0
15	- - - -	1	1	1	1	1

There are three combinations of inputs that makes the output 1.

O ((NOT I_3) AND I_2 AND I_1 AND I_0) OR (I_3 AND I_2 AND (NOT I_1) AND I_0) OR (I_3 AND I_2 AND I_1 AND I_0)

- In Boolean algebra, AND is also expressed as X, OR as +, NOT A as $\bar{A}$

↳ $O = \bar{I_3} I_2 I_1 I_0 + I_3 I_2 \bar{I_1} I_0 + I_3 I_2 I_1 I_0$ (O is 1 only in cases)

(The 16 terms are mutually exclusive, and covers all the cases.)

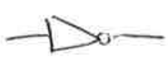
- The above rule is useful for any truth table. natural interpretation of AND, OR, NOT?

↳ Any Boolean function can be ~~used~~ written in AND, OR, NOT

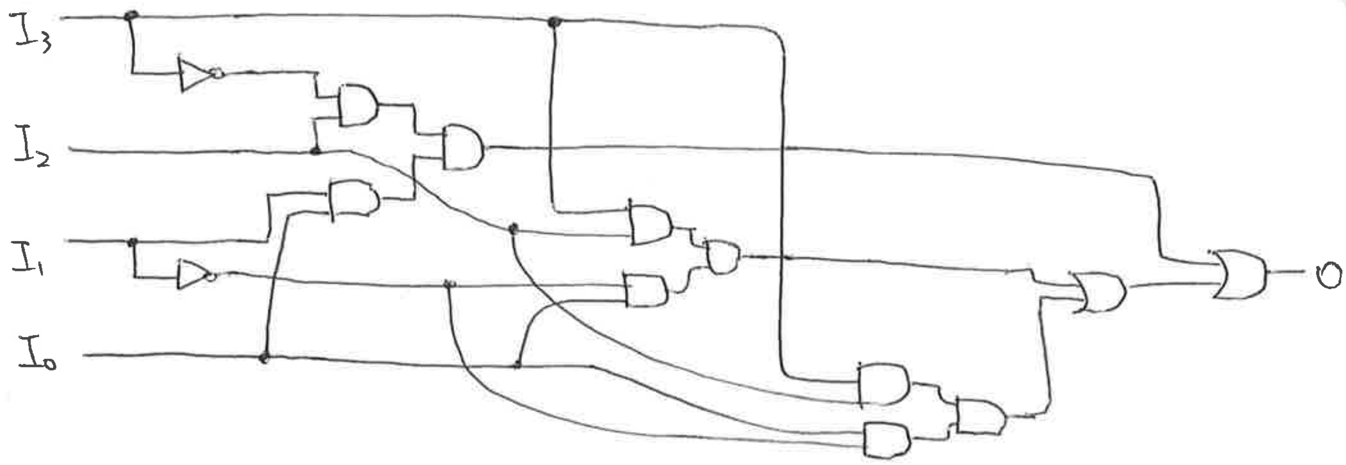
↳ Any combinational device can be made from 

and gate 

or gate 

inverter 

$\bar{I}_3 I_2 I_1 I_0 + I_3 I_2 \bar{I}_1 I_0 + I_3 I_2 I_1 I_0 = O$



- Can it be simpler?

[Boolean algebra] (proof truth table).

T1: Commutative $A+B=B+A$ $AB=BA$

T2: Associative $(A+B)+C = A+(B+C)$ $A(BC)=(AB)C$

T3: Distributive $A(B+C)=AB+AC$ $A+(BC) = (A+B)(A+C)$

Symmetric

T4: Identity $A+A=A$ $AA=A$

T5: $AB + A\bar{B} = A$ $(A+B)\cdot(A+\bar{B}) = A$

$\hookrightarrow A(B+\bar{B})$ $\hookrightarrow A+B\bar{B}$

T6: Redundence $\cancel{A}A+AB=A$ $A(A+B)=A$

T7: $0+A=A$ $0A=0$

T8: $1+A=1$ $1A=A$

T9: $\bar{A}+A=1$ $\bar{A}A=0$

T10: $A+\bar{A}B = A+B$ $A(\bar{A}+B)=AB$

$\hookrightarrow AB+A\cancel{B}+\bar{A}B$ (T6)

T11: De Morgan

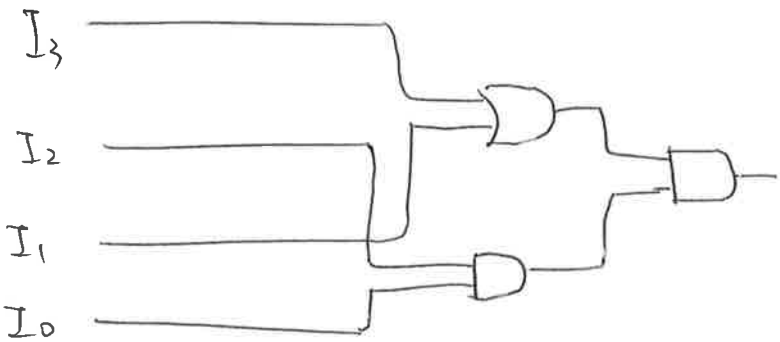
$\overline{A+B} = \bar{A}\bar{B}$ $\overline{AB} = \bar{A}+\bar{B}$

$$O = \bar{I}_3 I_2 I_1 I_0 + I_3 I_2 \bar{I}_1 I_0 + I_3 I_2 I_1 I_0$$

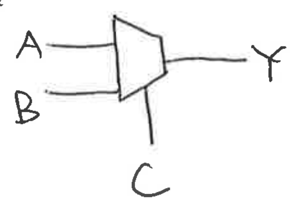
$$= (\bar{I}_3 I_1 + I_3 \bar{I}_1 + I_3 I_1) I_2 I_0$$

$$= (\bar{I}_3 I_1 + I_3) I_2 I_0$$

$$= (I_3 + I_1) I_2 I_0$$



Q



$C=0 \rightarrow Y=A$

$C=1 \rightarrow Y=B$

<data selection>