CpE111
Introduction to Computer Engineering

Dr. Minsu Choi

Supplemental CH 1: CMOS Logic Circuits

Introduction to CMOS

- CMOS is an acronym that stands for Complementary Metal-Oxide-Semiconductor and refers to a specific type of electronic integrated circuit (IC).
- CMOS is widely used in practice, since:
  1. Logic functions are very easy to implement.
  2. CMOS allows for very high logic integration density. This means that the logic circuits are very small and can be built in extremely small areas.
  3. The technology used to make silicon CMOS chips is very well known, and the chips can be manufactured and sold for a reasonable price.
- Use low and high voltages to represent logical 0 and 1 values.

Ex) Inverter

NOT Truth table

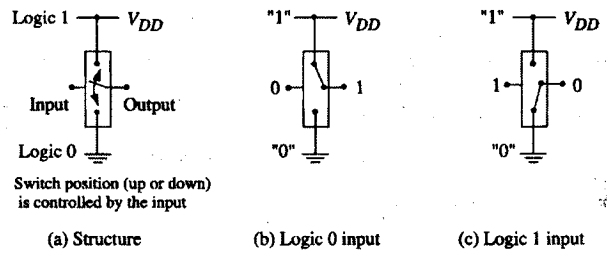
A	$\bar{A}$
0	1
1	0

$\Rightarrow$

electronic equivalent

V_{in}	V_{out}
0V	V_{DD}
V_{DD}	0V

- Electronic switch can be used to implement an inverter.



(a) Structure (b) Logic 0 input (c) Logic 1 input

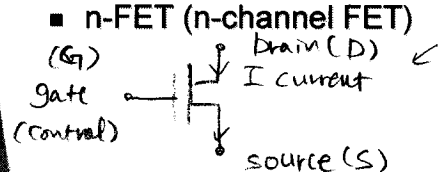
- => in CMOS, the switching action is implemented using electronic switching devices called "transistors".

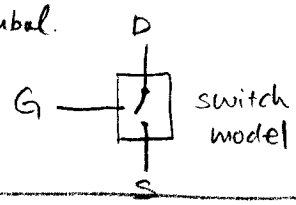
MOSFETs

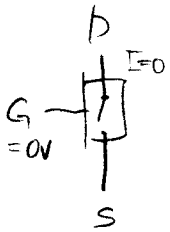
- Metal-Oxide-Semiconductor Field-Effect Transistors.
- Electronic devices that are used as switches.
- n-FET (n-channel FET)

There are two different types. n type & p type.

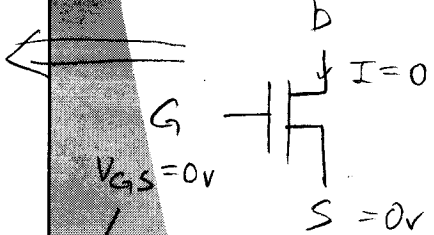
circuit symbol.

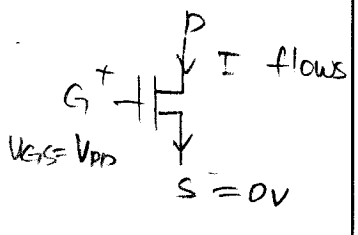


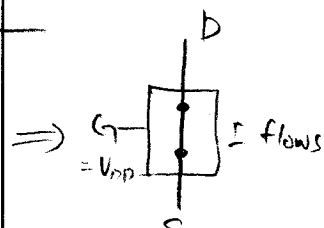




① open switch.







② closed switch

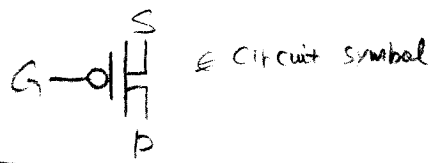
If (Gate − Source) voltage = 0V. then no current flows between D & S. This state is called cut off or off state

If V_{GS} = V_{DD}, the current flows ⇒ active or on state

Continued,

voltage polarities & directions are opposite.

- p-FET (p-channel FET) ≈ logic complement of the n-FET.



(Source - Gate) voltage

$V_{SG} = V_{DD}$, G = 0V, S = V_{DD}, I flows, D

① Active (on) state

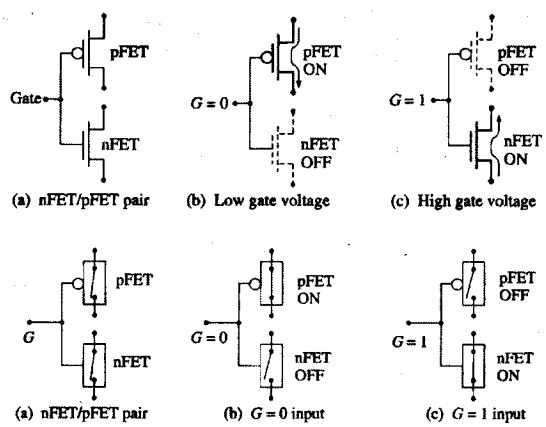
G = V_{DD}, S = V_{DD}, I = 0, D

② Cutoff (off) state

- -> voltage-controlled or logic-controlled devices.

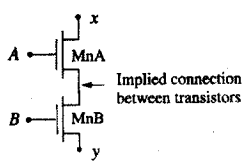
Complementary Pairs

- CMOS logic circuits use nFET and pFET transistors that are arranged as complementary pairs.
- Ex) Inverter implementation



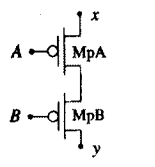
Logic Formulation using MOSFETs

- Series-connected MOSFETs and parallel-connected MOSFETs:



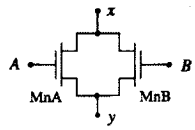
x is connected to y if and only if $A = 1$ AND $B = 1$

(a) nFET



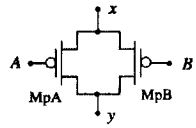
x is connected to y if and only if $A = 0$ AND $B = 0$

(b) pFET



x is connected to y if either $A = 1$ OR $B = 1$ (or both)

(a) nFETs



x is connected to y if either $A = 0$ OR $B = 0$ (or both)

(b) pFETs

A B	path x-y
0 0	open
0 1	open
1 0	open
1 1	closed

A B	path x-y
0 0	closed
0 1	open
1 0	open
1 1	open

A B	x-y
0 0	open
0 1	closed
1 0	closed
1 1	closed

A B	x-y
0 0	closed
0 1	closed
1 0	closed
1 1	open

complementary

complementary

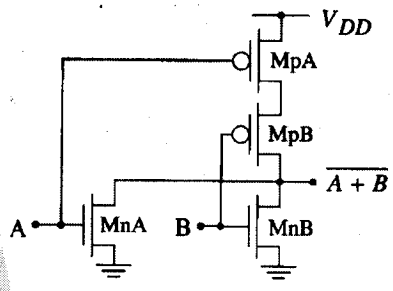
Continued,

1. Series-connected nFETs can be used to create the AND function for logic 0.
2. Parallel-connected nFETs can be used to create the OR function for logic 0.
3. Series-connected pFETs can be used to create the NOR function for logic 1.
4. Parallel-connected pFETs can be used to create the NAND function for logic 1.

- In CMOS, NOR and NAND and NOT gates are used as the basic gates.
- nFET arrays and pFET arrays can be combined accordingly to implement NOR, NAND and NOT gates.

The NOR Gate

two nFETs (MnA, MnB) are in parallel,
two pFETs (MpA, MpB) are in series.

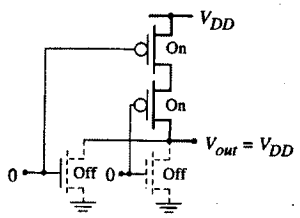


(a) CMOS circuit

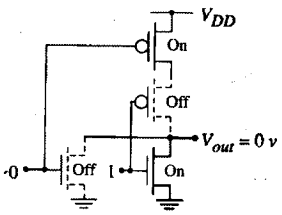
A	B	MnA	MnB	MpA	MpB	Out
0	0	OFF	OFF	**ON**	**ON**	1
0	1	OFF	**ON**	**ON**	OFF	**0**
1	0	**ON**	OFF	OFF	**ON**	**0**
1	1	**ON**	**ON**	OFF	OFF	**0**

(b) Operation summary

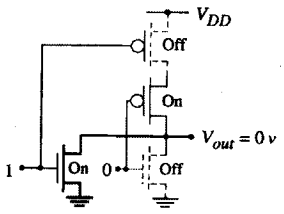
Four Possible NOR2 Circuit States



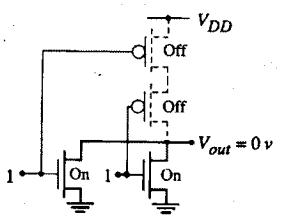
(a) $(AB) = (00)$



(b) $(AB) = (01)$



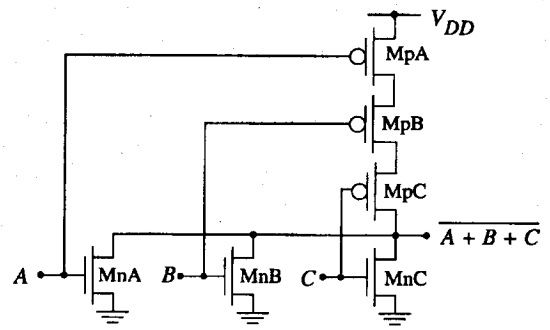
(c) $(AB) = (10)$



(d) $(AB) = (11)$

N-Input NOR gate

- The structure of the NOR2 gate can be easily extended to create NOR gates with more inputs.
- Ex) NOR3

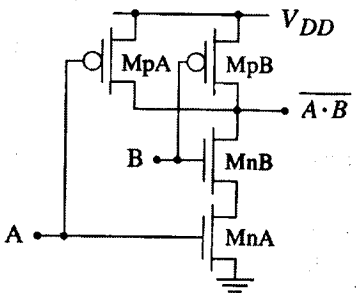


3 nFETs in parallel

3 pFETs in series

NAND2 Gate

two nFETs in series, two PFETs in parallel.



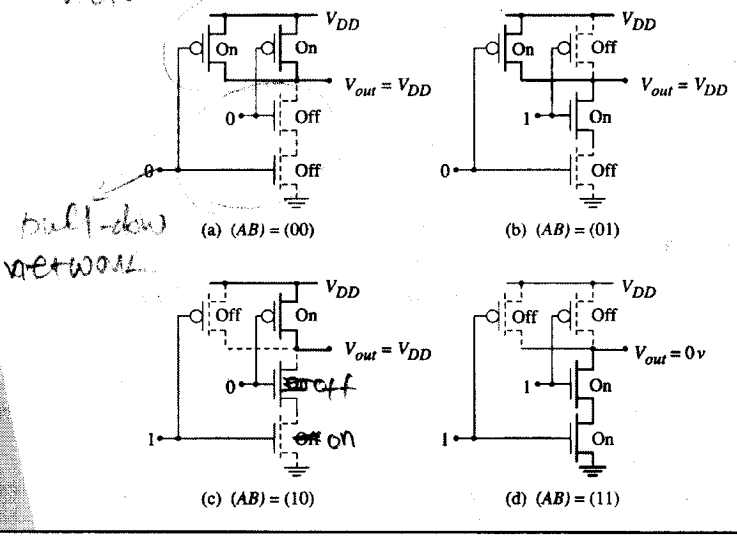
(a) CMOS circuit

A	B	MnA	MnB	MpA	MpB	Out
0	0	OFF	OFF	**ON**	**ON**	1
0	1	OFF	**ON**	**ON**	OFF	1
1	0	**ON**	OFF	OFF	ON	1
1	**1**	**ON**	**ON**	OFF	OFF	**0**

(b) Operation summary

Four Possible NAND2 Circuit States

pull-up network



pull-down network

Complex Logic Gates in CMOS

- We know how to build NOT, NAND and NOR gates.
- Theoretically, any logic function can be constructed.
- Complex logic gate: implements a function that can provide the basic NOT, AND, OR operations but integrate them into a single circuit.
  1. Complementary pairs of MOSFETs are used, so that input is connected to both an nFET and pFET.
  2. Complex logic gates will be constructed using the general nFET and pFET arrays.
  3. Series-parallel combinations of nFETs and pFETs will be used (when nFETs are in parallel, pFETs are in series and vice versa).

since {NOT, NAND, NOR} is complete

AOI and OAI

- We will find that CMOS is ideally suited for creating gates that have logic equations that exhibit…

1. AOI: AND-OR-INVERT (=complemented SOP form).
2. OAI: OR-AND-INVERT (=complemented POS form).

- Ex) AOI example
  1. Find complemented SOP of the given function.
  2. nFETs for AND terms -> serial, then connect in parallel to OR them up.
  3. pFETs for AND terms -> parallel, then connect in series to OR them up.

Continued,

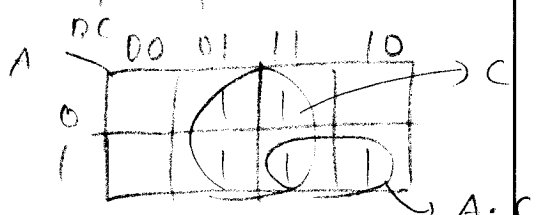
A B C	g	$\bar{g}$
0 0 0	1	0
0 0 1	0	1
0 1 0	1	0
0 1 1	0	1
1 0 0	1	0
1 0 1	0	1
1 1 0	0	1
1 1 1	0	1

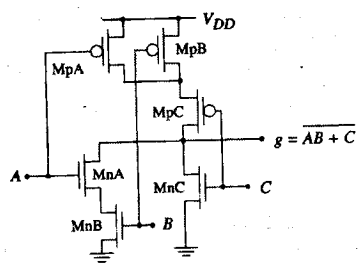
⇒ So, $\bar{g} = A \cdot B + C$

⇒ $g = \overline{A \cdot B + C}$

AOI form.

⇒ k-map simplification of $\bar{g}$

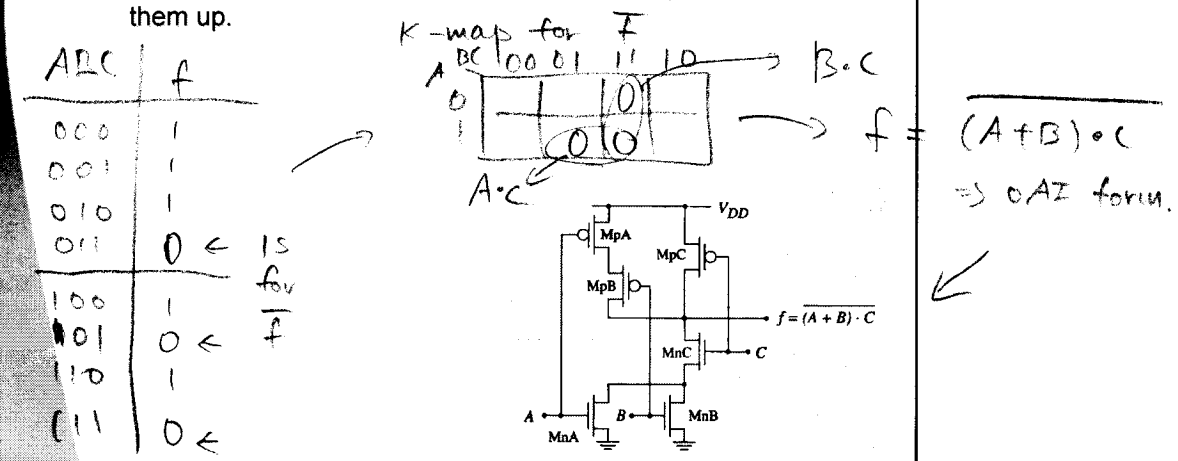




(☆) nFETs for AND terms → serial → parallel
pFETs for AND terms → parallel → serial
to OR them up.

OAI Example

1. nFETs for OR terms -> parallel, then connect in series to AND them up.
2. pFET for OR terms -> series, then connect in paralllel to AND them up.



Program Completed

University of Missouri-Rolla

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.