CMOS Logic

MOS Transistors - 2 Types

N莫斯

Source  Gate  Drain

S

G

D

P+  P+

Pmos

VDD  Closed Switch

Logic 1

VDD  Open Switch

Logic 0
CMOS INVERTER

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

OR

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VDD</td>
</tr>
<tr>
<td>VDD</td>
<td>0</td>
</tr>
</tbody>
</table>

CMOS NAND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
CMOS NDR

\[ \begin{array}{c}
A \\
B \\
0 \quad 0 \quad 1 \\
\hline
A \\
B \\
1 \\
\end{array} \]

\[ \begin{array}{c}
A \quad B \\
0 \quad 0 \quad 1 \\
0 \quad 1 \quad 0 \\
1 \quad 0 \quad 0 \\
1 \quad 1 \quad 0 \\
\end{array} \]
IN GENERAL

AVDD

PULL UP NETWORK (PUN)

PMOS TRANSISTOR

VA
VB
VC

PUN

DOUT

PULL DOWN NETWORK (PDN)

NMOS TRANSISTORS

PDN

DESIGN PDN (NMOS)

PUN IS DUAL OF PDN (IF NOT 2 OUTPUT)

⇒ REPLACE SERIES SUBNETS WITH PARALLEL

"PARALLEL" "PARALLEL" "SERIES"

4 POSSIBLE OUTPUTS

+ PULL UP + PULL UP

PULL DOWN OFF 0

PULL DOWN ON 1 CROWBAR (X)

2⇒) HIGH IMPEDANCE OUTPUT.
Example Design: \( y = (a+b)c \)
Example \[ f = a(b+c) + d \]
SOME BOOLEAN ALGEBRA LAWS

\[ A(B+C) = AB + AC \]

\[ A = \overline{A} \]

\[ ABC = \overline{A} + \overline{B} + \overline{C} \]

\[ A + B + C = \overline{ABC} \]

\[ A + 0 = A \]
\[ A + 1 = 1 \]
\[ A \cdot 0 = 0 \]
\[ A \cdot 1 = A \]
\[ A + A = A \]
\[ A + \overline{A} = 1 \]
\[ A \cdot A = A \]
PASS TRANSISTOR T GATE

**NMOS**

\[ g = 0 \implies \text{Blocks both 0 and 1 well} \]
\[ g = 1 \implies \text{Passes 0 well} \]
\[ g = 1 \implies \text{Degrades 1 signal} \]

\[ g = 1 \implies \text{Blocks both 1 and 0 well} \]
\[ g = 0 \implies \text{Passes '1' well} \]
\[ g = 0 \implies \text{Degrades '0' signal} \]

**PMOS**

\[ g = 1 \implies \text{Blocks both 1 and 0 well} \]

\[ g = 0 \implies \text{Pass both well} \]
\[ g = 0 \implies \text{Blocks both well} \]

T-GATE

\[ g = 1 \text{ (} \bar{g} = 0 \text{)} \implies \text{Pass both well} \]
\[ g = 0 \text{ (} \bar{g} = 1 \text{)} \implies \text{Blocks both well} \]
WHY AN NMOS DOES NOT PASS "1" WELL

RECALL

\[ V_{GS} \leq V_T \quad I_D = 0 \] (CUTTOFF)

\[ V_{GS} > V_T \quad \text{ACTIVE (SATURATION)} \]

OR TRIODE

\[ V_y = \frac{1}{C_L} \]

INITIALLY \( V_y = 0 \)

\[ V_y \]

\[ V_{DD} \]

\[ 0 \]

\[ V_{DD} - V_T \]

\[ V_y \]

\[ 0 \]

\[ V_y \]

\[ V_T \]

\[ t \]

\[ t_0 \]

\[ t_1 \]

NMOS GOES INTO CUTTOFF AT \( t = t_1 \)

SO \( I_D = 0 \) AND \( V_y \) REMAINS AT \( V_{DD} - V_T \)

\[ \Rightarrow \text{SIMILAR FOR PMOS} \]
RESTORING VS NON-RESTORING LOGIC

CONSIDER INVERTER

\[ \text{If } V_A = 0.1 \text{ V } \Rightarrow V_B = V_{DD} \]
\[ \text{If } V_A = V_{DD} - 0.1 \text{ V } \Rightarrow V_B = 0 \text{ V} \]

ERRORS ON \( V_A \) STILL RESULT IN CORRECT LOGIC LEVELS

RESTORING BUFFER

\[ V_A \rightarrow V_C \rightarrow V_B \leftrightarrow V_A \rightarrow V_B \]

\[ V_{DD} \]

\[ 0 \rightarrow V_{DD} \rightarrow V_A \]
NON-RESTORING BUFFER

EX 1

FOR \( V_A = V_{DD} - 0.1V \) \( \Rightarrow \) \( V_B = V_{DD} - 0.1 - V_t \)

EX 2

FOR \( V_A = 0.1V \) \( \Rightarrow \) \( V_B = 0.1V \)

\( V_A = V_{DD} - 0.1V \) \( \Rightarrow \) \( V_B = V_{DD} - 0.1V \)
RESTORED LOGIC REQUIRES INVERSIONS!

A → Y

BUFFER

Slight Hysteresis Degraded
Inputs

RESTORED TO
"1" [LO] [HO]"
RESTORING TRI STATE

\[ EN \rightarrow A \rightarrow \bar{EN} \rightarrow Y \]

\[
\begin{array}{c|cc}
EN/\bar{EN} & A & Y \\
\hline
0/1 & 0 & 0 \\
0/1 & 1 & 1 \\
1/0 & 0 & 1 \\
1/0 & 1 & 0 \\
\end{array}
\]

MULTIPLEXER

\[ S \]

\[ A \rightarrow Y \quad B \rightarrow Y \]

MORE POPULAR
LATCHES AND FLIP-FLOPS (REGISTERS)

LATCHE

\[ \begin{array}{c}
D & D & Q \\
\text{LATCH} & \text{CLK}
\end{array} \]

NOTE WORD "LATCH" ON SYMBOL

WHEN \( \text{CLK} = 1 \) \( \Rightarrow \) \( Q = D \) (TRACK MODE)
WHEN \( \text{CLK} = 0 \) \( \Rightarrow \) \( Q \) HELD (HOLD MODE)

HOW TO HOLD 2 INVERTER MEMORY CELL

IF \( A = 1 \) \( \Rightarrow \) \( B = 0 \) \( \Rightarrow \) \( A = 1 \) \( \Rightarrow \) \( B \) HELD AT \( B = 0 \)

IF \( A = 0 \) \( \Rightarrow \) \( B = 1 \) \( \Rightarrow \) \( A = 0 \) \( \Rightarrow \) \( B \) HELD AT \( B = 1 \)
To create a latch using transmission gate multiplexer:

1. Connect the input signal (D) to the latch.
2. Connect the clock (CLK) to the latch.
3. Connect the output (Q) to the next stage.
4. Use transmission gate multiplexer to control the flow of signals.
IF $CLK = 1$ (TRACK mode) $\Rightarrow Q = 0$

$D$ ——— $Q$

IF $CLK = 0$ (HOLD mode) $\Rightarrow Q$ HELD

$D$ ——— $Q$

**TIMING DIAGRAM**

CLK

D

Q
FLIP-FLOP (ALSO CALLED REGISTER)

NOTE NO WORD "LATCH" ON SYMBOL

when \( \text{CLK} = 0 \)  \( Q \) held

when \( \text{CLK} = 1 \)  \( Q \) held

when \( \text{CLK} \) changes from 0 \( \rightarrow \) 1  \( Q = D \)

when \( \text{CLK} \) changes from 1 \( \rightarrow \) 0  \( Q \) held

TIMING DIAGRAM
**Build a Register using 2 Latches**

**Diagram:**

```
D ----> L1 ----> Q1 ----> L2 ----> Q ----> D
       |          |          |          |          |
     LATCH    LATCH    LATCH    LATCH    LATCH
       |          |          |          |          |
     CLK      CLK      CLK      CLK      CLK
```

**When CLK = 0:**

- **L1** in TRACK mode
  - \( Q_1 = D \)

- **L2** in HOLD mode
  - So \( Q \) held

**When CLK = 1:**

- **L1** in HOLD mode
  - \( \implies Q_1 \) held

- **L2** in TRACK mode
  - \( \implies Q = Q_1 (\iff Q_1 \text{ held}) \)

**Q REMAINS UNCHANGED**
When CLK \( \uparrow \) (RISING EDGE)

L1 changes to HOLD mode and samples D at that instant.
L2 changes to TRACK mode so sampled D appears at output Q.

When CLK \( \downarrow \) (FALLING EDGE)

L2 changes to HOLD mode so Q remains unchanged.
L1 changes to TRACK mode so \( Q_1 = D \).
REGISTER IMPLEMENTATION

D

CLK

CLK

CLK

Q

L1

L2

However, here Q = D but non-restoring logic, instead, use 2 inverting latches so restoring logic for Q = D

D

CLK

Q

LATCH

LATCH