3. Implementing Logic in CMOS

J. A. Abraham

Department of Electrical and Computer Engineering
The University of Texas at Austin

EE 382M.7 – VLSI I
Fall 2011

August 31, 2011

Static CMOS Circuits

N- and P-channel Networks

- N- and P-channel networks implement logic functions
- Each network connected between Output and $V_{DD}$ or $V_{SS}$
- Function defines path between the terminals

Parallel network: "OR" function
Series network: "AND" function
Duality in CMOS Networks

Straightforward way of constructing static CMOS circuits is to implement dual N- and P- networks

- N- and P- networks must implement complementary functions
- Duality sufficient for correct operation (but not necessary)

Constructing Complex Gates

Example: \( F = (A \cdot B) + (C \cdot D) \)

- Take uninverted function \( F = (A \cdot B) + (C \cdot D) \) and derive N-network
- Identify AND, OR components: F is OR of AB, CD
- Make connections of transistors
Construct P-network by taking complement of N-expression \((AB + CD)\), which gives the expression, \((\overline{A + B}) \cdot (\overline{C + D})\).

Combine P and N circuits.

Layout of Complex Gate

AND-OR-INVERT (AOI) gate

Note: Arbitrary shapes are not allowed in some nanoscale design rules.
Example of Compound Gate

\[ F = (A + B + C) \cdot D \]

Note:

N- and P- graphs are duals of each other.

In this case, the function is the complement of the switching function between \( F \) and GND.

Question: Does it make any difference to the function if the transistor with input \( D \) is connected between the parallel \( A, B, C \), transistors and GND?

What about the electrical behavior?

Example of More Complex Gate

\[ OUT = (A + B) \cdot (C + D) \cdot (E + F + (G \cdot H)) \]
Exclusive-NOR Gate in CMOS

Note: designs such as these should be checked very carefully for correct behavior using circuit simulation.

Pseudo nMOS Logic

Based on the old NMOS technology where a “depletion” transistor was used as a pullup resistor.

What happens when there is no path from Z to ground (i.e., Z = 1)?

What happens when there is a path from Z to ground (i.e., Z = 0)?
**Duality is Not Necessary for a CMOS Structure**

Functions realized by N and P networks must be **complementary**, and one of them must conduct for every input combination

\[ F = a \cdot b + \overline{a} \cdot \overline{b} + a \cdot c + c \cdot d + \overline{a} \cdot \overline{b} \]

The N and P networks are NOT duals, but the switching functions they implement are complementary.

**Example of Complex CMOS Gate**

\[ F = \quad \text{and} \quad G = \]

Department of Electrical and Computer Engineering, The University of Texas at Austin

J. A. Abraham, August 31, 2011
Example of Complex CMOS Gate, Cont'd

$F = x \cdot y \cdot z + x \cdot y \cdot z + x \cdot y \cdot z + x \cdot y \cdot z$

$G = \overline{x} \cdot y \cdot z + x \cdot \overline{y} \cdot z + x \cdot y \cdot \overline{z}$

Example of Complex CMOS Gate

$F = x \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot z + x \cdot \overline{y} \cdot z + x \cdot y$

$G = \overline{x} \cdot y \cdot \overline{z} + x \cdot \overline{y} \cdot \overline{z} + \overline{x} \cdot \overline{y}$
### Signal Strength

Voltages represent digital logic values:

- **Strength** of signal:
  - How close it approximates ideal voltage
- $V_{DD}$ and $GND$ rails are strongest 1 and 0
- nMOS transistors pass a strong 0
  - But degraded or weak 1
- pMOS transistors pass a strong 1
  - But degraded or weak 0

Therefore, nMOS transistors are best for the “pull-down” network.

---

### Pass Transistors and Transmission Gates

Transistors can be used as switches; however, they could produce degraded outputs.

#### Transmission gates pass both 0 and 1 well
Pass Transistor Logic

“Pull-Up” Circuit

- Used to restore degraded logic 1 from output of nMOS pass transistor

Pass Transistor Logic – Better Layout

Group similar transistors, so they can be in the same well
Tristates

**Tristate Buffer** produces $Z$ (high impedance) when not enabled

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

Non-Restoring Tristate

- Transmission gate acts as a tristate buffer
- Only two transistors, but nonrestoring
- Noise on $A$ is passed to $Y$

Tristate Inverter

Tristate inverter produces restored output, but complements signal

\[ EN = 0 \quad Y = Z' \]
\[ EN = 1 \quad Y = A \]
How many transistors are needed? (The better design uses 3 NAND gates and 1 inverter)

Transmission Gate MUX

Nonrestoring MUX
- Uses two transmission gates $\Rightarrow$ only 4 transistors

Inverting MUX – adds an inverter
- Uses compound gate AOI22
- Alternatively, a pair of tristate inverters (same thing)
### 4:1 Multiplexer

A 4:1 MUX chooses one of 4 inputs using two selects

- Two levels of 2:1 MUXes
- Alternatively, four tristates

![4:1 Multiplexer Diagram](image)

### D Latch

**Basic Memory Element**

- When CLK = 1, latch is transparent
  - D flows through to Q like a buffer
- When CLK = 0, the latch is opaque
  - Q holds its old value independent of D
- a.k.a., transparent latch or level-sensitive latch

![D Latch Timing](image)
D Latch, Cont’d

D Latch Design:
MUX chooses
between D and
old Q

D Latch
Operation

D Flip-Flop (D-Flop)

Another common storage element

- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- positive edge-triggered flip-flop or master-slave flip-flop
- Built from “master” and “slave” D latches
D Flip-Flop Operation

Race Condition – Hold Time Failure

Back-to-back flops can malfunction from clock skew
- Second flip-flop fires late
- Sees first flip-flop change and captures its result
- Called hold-time failure or race condition
Non-Overlapping Clocks

A simple way to prevent races
- This works as long as non-overlap exceeds clock skew
- Used in safe (conservative) designs
- Industry does not generally use this approach – managing skew more carefully instead

Gate Layout

Building a library of standard cells
- Layout can be time consuming
- One solution is to have layouts of commonly used functions (Inverter, NAND, OR, MUX, etc.), designed to fit together very well

Standard cell design methodology
- $V_{DD}$ and $GND$ should abut (standard height)
- Adjacent gates should satisfy design rules
- nMOS at bottom and pMOS at top
- All gates include well and substrate contacts
- One of the large industry suppliers is ARM (which purchased ARTISAN)
Examples of Standard Cell Layout

Inverter

NAND3

Horizontal N-diffusion and P-diffusion strips
Vertical Polysilicon gates
Metal1 $V_{DD}$ rail at top, Metal1 $GND$ rail at bottom
32$\lambda$ by 40$\lambda$

Wiring Tracks and Well Spacing

**Wiring Track** is the space required for a wire
Example, $4\lambda$ width, $4\lambda$ spacing from neighbor = $8\lambda$ pitch
Transistors also consume one wiring track

Example, **well spacing**: wells must surround transistors by $6\lambda$
Implies $12\lambda$ between opposite transistor flavors
Leaves room for one wire track
Example of Area Estimation

Estimate area by counting wiring tracks
Multiply by 8 to express in \( \lambda \)

Example Circuit 1

Fill in the Karnaugh map to represent the Boolean function implemented by the pass-transistor circuit.
**Example Circuit 1**

Find the function, $F$, implemented by the following circuit

$F(a,b,c,d,e):$

```
0 0 0 1 1
1 1 0 1 1
```

**Example Circuit 2**

Find the function, $F$, implemented by the following circuit

$A + BC + B\overline{C}$
Example Circuit 2

Find the function, \( F \), implemented by the following circuit

\[
\overline{A} + BC + \overline{B} \overline{C}
\]

Example Circuit 3

Find the functions \( X \) and \( Y \) implemented by the following circuit
Example Circuit 3

Find the functions X and Y implemented by the following circuit.

\[
X = A + B \\
Y = A \cdot B
\]