Designing Combinational Logic Circuits
Combinational vs. Sequential Logic

Combinational

Output = \( f(In) \)

Sequential

Output = \( f(In, Previous\ In) \)
Static CMOS Circuit

At every point in time (except during the switching transients) each gate output is connected to either $V_{DD}$ or $V_{ss}$ via a low-resistive path.

The outputs of the gates assume at all times the value of the Boolean function, implemented by the circuit (ignoring, once again, the transient effects during switching periods).

This is in contrast to the dynamic circuit class, which relies on temporary storage of signal values on the capacitance of high impedance circuit nodes.
CMOS Circuit Styles

- **Static complementary CMOS** - except during switching, output connected to either $V_{DD}$ or GND via a low-resistance path
  - high noise margins
  - full rail to rail swing
  - $V_{OH}$ and $V_{OL}$ are at $V_{DD}$ and GND, respectively
  - low output impedance, high input impedance
  - no steady state path between $V_{DD}$ and GND (no static power consumption)
  - delay a function of load capacitance and transistor resistance
  - comparable rise and fall times (under the appropriate transistor sizing conditions)

- **Dynamic CMOS** - relies on temporary storage of signal values on the capacitance of high-impedance circuit nodes
  - simpler, faster gates
  - increased sensitivity to noise
Static Complementary CMOS

- Pull-up network (PUN) and pull-down network (PDN)

PMOS transistors only

pull-up: make a connection from $V_{DD}$ to $F$ when $F(In_1, In_2, \ldots, In_N) = 1$

NMOS transistors only

pull-down: make a connection from $F$ to GND when $F(In_1, In_2, \ldots, In_N) = 0$

PUN and PDN are dual logic networks

One and only one of the networks is conducting in steady state
Threshold Drops

PUN

PDN
Threshold Drops

PUN

\[ V_{DD} \rightarrow 0 \rightarrow V_{DD} \]

\[ V_{GS} \rightarrow V_{DD} - V_{Tn} \]

PDN

\[ V_{DD} \rightarrow 0 \rightarrow V_{DD} \]

\[ V_{GS} \rightarrow |V_{Tp}| \]
NMOS Transistors in Series/Parallel Connection

Transistors can be thought as a switch controlled by its gate signal.

NMOS switch closes when switch control input is high.

Y = X if A and B

Y = X if A OR B

NMOS Transistors pass a “strong” 0 but a “weak” 1.
PMOS Transistors in Series/Parallel Connection

PMOS switch closes when switch control input is low

PMOS Transistors pass a “strong” 1 but a “weak” 0
Construction of PDN

- NMOS devices in **series** implement a NAND function
  
  ![NAND circuit diagram]

- NMOS devices in **parallel** implement a NOR function
  
  ![NOR circuit diagram]
Dual PUN and PDN

- PUN and PDN are dual networks
  - DeMorgan’s theorems
    \[ \overline{A + B} = \overline{A} \cdot \overline{B} \]
    \[ \overline{A \cdot B} = \overline{A} + \overline{B} \]
  - a parallel connection of transistors in the PUN corresponds to a series connection of the PDN
- Complementary gate is naturally inverting (NAND, NOR, XNOR)
- Number of transistors for an N-input logic gate is 2N
Complementary CMOS Logic Style

- PUN is the **DUAL** of PDN
  (can be shown using DeMorgan’s Theorem’s)

\[
\overline{A + B} = \overline{A} \overline{B} \\
\overline{AB} = \overline{A} + \overline{B}
\]

- The complementary gate is inverting

\[\text{AND} = \text{NAND} + \text{INV}\]
CMOS NAND

\[ A \cdot B \]

\[
\begin{array}{ccc}
A & B & F \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]
CMOS NOR

\[ V_{DD} \]

\[ \overline{A + B} \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Complex CMOS Gate

F = D + A \cdot (B + C)
Complex CMOS Gate

\[ \text{OUT} = D + A \cdot (B + C) \]

Diagram showing a CMOS gate with inputs A, B, C, and D, and output OUT. The diagram includes labeled regions SN1, SN2, SN3, and SN4 with the equation for the output expression.
The threshold voltage of $M_2$ is higher than $M_1$ due to the body effect ($\gamma$).

$$V_{Tn1} = V_{Tn0}$$

$$V_{Tn2} = V_{Tn0} + \gamma (\sqrt{2|\phi_F|} + V_{int}) - \sqrt{2|\phi_F|}$$

since $V_{SB}$ of $M_2$ is not zero (when $V_B = 0$) due to the presence of $Cint$.
Switch Delay Model

NAND2

INV

NOR2
Input Pattern Effects on Delay

- Delay is dependent on the pattern of inputs
- Low to high transition
  - both inputs go low
    - delay is $0.69 \times \left( \frac{R_p}{2} \right) C_L$
  - one input goes low
    - delay is $0.69 \times R_p \times C_L$
- High to low transition
  - both inputs go high
    - delay is $0.69 \times 2R_n \times C_L$
Adding devices in series slows down the circuit, and devices must be made wider to avoid performance penalty.

When sizing the transistors in a gate with multiple inputs, we should pick the combination of inputs that triggers the worst case conditions.

For the NAND gate to have the same pull-down delay ($t_{phl}$) with an inverter, the NMOS devices in the PDN stack must be made twice as wide (2.5 times, if velocity saturation is effective). PMOS devices can remain unchanged.

(Extra capacitance introduced by widening is ignored here, which is not a good assumption)
**Delay Dependence on Input Patterns**

_NAND Gate_

<table>
<thead>
<tr>
<th>Input Data Pattern</th>
<th>Delay (psec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=B=0 → 1</td>
<td>69</td>
</tr>
<tr>
<td>A=1, B=0 → 1</td>
<td>62</td>
</tr>
<tr>
<td>A=0 → 1, B=1</td>
<td>50</td>
</tr>
<tr>
<td>A=B=1 → 0</td>
<td>35</td>
</tr>
<tr>
<td>A=1, B=1 → 0</td>
<td>76</td>
</tr>
<tr>
<td>A= 1 → 0, B=1</td>
<td>57</td>
</tr>
</tbody>
</table>

NMOS = 0.5μm/0.25 μm
PMOS = 0.75μm/0.25 μm

CL = 100 fF
Transistor Sizing

\[
\begin{align*}
\text{Diagram 1:} & \quad R_p \quad R_p \\
\text{Diagram 2:} & \quad R_n \quad C_{\text{int}} \\
\end{align*}
\]
Transistor Sizing a Complex CMOS Gate

\[ \text{OUT} = D + A \cdot (B + C) \]
Fan-In Considerations

Distributed RC model (Elmore delay), assuming all NMOS devices of equal size

\[ t_{pHL} = 0.69 \frac{R_{eqn}(C_1+2C_2+3C_3+4C_L)}{R_1 R_2 R_3 R_4 C_L} \]

Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

\[ t_{pHL} = 0.69 \left[ R_1 C_1 + (R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3 + (R_1 + R_2 + R_3 + R_4)C_L \right] \]
\[ t_p \text{ of CMOS NAND as a Function of Fan-In} \]

(Assuming a fixed fan out of one inverter)

Gates with a fan-in greater than 4 should be avoided.
All gates have the same drive current.

Slope is a function of “driving strength”
$t_p$ as a Function of Fan-In and Fan-Out

- Fan-in: quadratic due to increasing resistance and capacitance
- Fan-out: each additional fan-out gate adds two gate capacitances to $C_L$
Fast Complex Gates: Design Technique 1

- Transistor sizing
  - as long as fan-out capacitance dominates
- Progressive sizing

![Diagram](image)

- Distributed RC line
- M1 > M2 > M3 > … > MN (the FET closest to the output is the smallest)
- Can reduce delay by more than 20%; decreasing gains as technology shrinks
Fast Complex Gates: Design Technique 2

- Transistor ordering

An input signal is called “critical” if it is the last signal to assume a stable value, the path which determines the ultimate speed is called “critical path”

![Diagram showing transistor ordering and critical path](image)

- delay determined by time to discharge $C_L$, $C_1$ and $C_2$
- delay determined by time to discharge $C_L$
Fast Complex Gates: Design Technique 3

- Alternative logic structures

\[ F = ABCDEFGH \]
Fast Complex Gates: Design Technique 4

- Isolating fan-in from fan-out using buffer insertion

![Diagram showing buffer insertion technique]
Logical Effort

\[ t_p = t_{p0} \left( 1 + \frac{C_{ext}}{\gamma C_g} \right) = t_{p0} \left( 1 + \frac{f}{\gamma} \right) \]

\[ t_p = t_{p0} (p + g f / \gamma) \]

\( p \) – ratio of the intrinsic (unloaded delays of the complex gate and the simple inverter
\( g \) – logical effort – how much more input capacitance a gate presents to deliver the same output current as an inverter (how much worse it is at producing output current than an inverter)
\( f \) – effective fanout \((C_{ext}/C_g)\) (electrical effort)

Normalize everything to an inverter: \( g_{inv} = 1, \ p_{inv} = 1 \)
\( P = n \) for \( n \) input NAND and NOR gates
Assume \( \gamma = 1 \).
Logical Effort

- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates.
- Logical effort of a gate presents the ratio of its input capacitance to the inverter capacitance when sized to deliver the same current.
- Logical effort increases with the gate complexity.
Delay in a Logic Gate

Gate delay:

\[ d = h + p \]

Effort delay: intrinsic delay

Effort delay (gate effort):

\[ h = g f \]

Logical effort: effective fanout = \( C_{out}/C_{in} \)

Logical effort is a function of topology, independent of sizing

Effective fanout (electrical effort) is a function of load/gate size
Logical Effort

Assuming PMOS/NMOS ratio of 2, the input capacitance of a minimum sized symmetrical inverter equals 3 times the gate capacitance of a minimum sized NMOS (called $C_{\text{unit}}$)

$$g = \frac{4}{3}(4 \ C_{\text{unit}})$$

$$g = \frac{5}{3}(5 \ C_{\text{unit}})$$
Intrinsic Delay Term, $p$

- The more involved the structure of the complex gate, the higher the intrinsic delay compared to an inverter.

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>$n$-input NAND</td>
<td>$n$</td>
</tr>
<tr>
<td>$n$-input NOR</td>
<td>$n$</td>
</tr>
<tr>
<td>$n$-way mux</td>
<td>$2n$</td>
</tr>
<tr>
<td>XOR, XNOR</td>
<td>$n \ 2^{n-1}$</td>
</tr>
</tbody>
</table>

Ignoring second order effects such as internal node capacitances.
# Logical Effort

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>Number of Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Inverter</td>
<td>1</td>
</tr>
<tr>
<td>NAND</td>
<td>4/3</td>
</tr>
<tr>
<td>NOR</td>
<td>5/3</td>
</tr>
<tr>
<td>Multiplexer</td>
<td>2</td>
</tr>
<tr>
<td>XOR</td>
<td>4</td>
</tr>
</tbody>
</table>

For the same input capacitance, 2 input NAND and NOR gates have 4/3 and 5/3 less driving strength than the inverter.
Logical Effort of Gates

- Fan-out ($f$)
- Normalized delay ($d$)

For a pINV gate:
- $g = 1$
- $p = 1$
- $d = f + 1$

For a pNAND gate:
- $g = 4/3$
- $p = 2$
- $d = (4/3)f + 2$

Graph showing the normalized delay ($d$) vs. fan-out ($f$) for pINV and pNAND gates.
Delay as a Function of Fan-Out

- The slope of the line is the logical effort of the gate \( d = p + fg \)
- The y-axis intercept is the intrinsic delay
- Can adjust the delay by adjusting the effective fan-out (by sizing) or by choosing a gate with a different logical effort
- Gate effort: \( h = fg \)
**Multistage Networks**

Total delay of a path:

\[ t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0} \sum_{j=1}^{N} \left( p_j + \frac{f_j g_j}{\gamma} \right) \]

Using a similar procedure with the sizing of the inverter chain (finding N-1 partial derivatives and equating to zero), we find that each stage should bear the same gate effort:

\[ f_1 g_1 = f_2 g_2 = \ldots = f_N g_N \]

Here we have some definitions:

Path effective fan-out (Path electrical effort): \( F = C_L / C_{g1} \)

Path logical effort: \( G = g_1 g_2 \ldots g_N \quad G = \prod_{i=1}^{N} g_i \)
Branching effort of a logical gate on a path:

\[ b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}} \]

Path branching effort: \[ B = \prod_{i=1}^{N} b_i \]

The path electrical effort can now be related to the electrical and branching efforts:

\[ F = \prod_{i=1}^{N} \frac{f_i}{b_i} = \frac{\prod f_i}{B} \]
Finally the total path effort $H$ can be defined:

$$H = \prod_{i=1}^{N} h_i = \prod_{i=1}^{N} g_i f_i = GFB$$

From here on, the analysis can be carried out as in the case of inverter chain. The gate effort that minimizes the path delay is:

$$h = \sqrt[N]{H}$$

And the minimum delay through the path is:

$$D = t_{p0} \left( \sum_{j=1}^{N} p_j + \frac{N\left(\sqrt[N]{H}\right)}{\gamma} \right)$$
We assume that a unit-size gate has a driving capability equal to a minimum-size inverter. This means that its input capacitance is $g$ times larger than that of the reference inverter ($C_{\text{ref}}$). With $s_1$ the sizing factor of the first gate in the chain, the input capacitance of the chain can be written as:

$$C_{g_1} = g_1s_1C_{\text{ref}}$$

Including the branching effort, the input capacitance of gate 2 will be $f_1/b_1$ times larger:

$$g_2s_2C_{\text{ref}} = \left(\frac{f_1}{b_1}\right)g_1s_1C_{\text{ref}}$$

For gate $i$ in the chain:

$$s_i = \left(\frac{g_1s_1}{g_i}\right)\prod_{j=1}^{i-1}\left(\frac{f_j}{b_j}\right)$$
**Example: Optimize Path**

![Diagram of a circuit with nodes labeled](image)

**Effective fanout, F = 5**

- $G = 1 \times (5/3) \times (5/3) \times 1 = 25/9$
- $H = \text{GFB (no branching)} = 125/9 = 13.9$
- $h = \sqrt[4]{H} = 1.93$

Since $f_1g_1 = f_2g_2 = f_3g_3 = f_4g_4 = h$:
- $f_1 = 1.93$, $f_2 = 1.93 \times (3/5) = 1.16$, $f_3 = 1.16$, $f_4 = 1.93$
- $a = s_2 = f_1g_1/g_2 = 1.16$
- $b = s_3 = f_1f_2g_1/g_3 = 1.34$
- $c = s_4 = f_1f_2f_3g_1/g_4 = 2.60$
Path Logical Effort Variation With Restructuring
(8 – input AND)

\[ g = \frac{10}{3}, \quad g = 1 \]
\[ G = 3.33 \]

\[ g = 2, \quad g = \frac{5}{3} \]
\[ G = 3.33 \]

\[ g = \frac{4}{3}, \quad g = \frac{5}{3}, \quad g = \frac{4}{3}, \quad g = 1 \]
\[ G = 2.96 \]
Dynamic Power Consumption is Data Dependent

- Switching activity, $P_{0\rightarrow 1}$, has two components
  - A static component – function of the logic topology
  - A dynamic component – function of the timing behavior (glitching)

$$P_{\text{dyn}} = C_L V_{DD}^2 f_{0\rightarrow 1} = C_L V_{DD}^2 P_{0\rightarrow 1} f = C_{\text{EFF}} V_{DD}^2 f$$

2-input NOR Gate

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

Static transition probability

$$P_{0\rightarrow 1} = P_{\text{out}=0} \times P_{\text{out}=1}$$

$$= P_0 \times (1-P_0)$$

With input signal probabilities

$$P_{A=1} = 1/2$$
$$P_{B=1} = 1/2$$

NOR static transition probability

$$= 3/4 \times 1/4 = 3/16$$
\( \text{N}_0 \) and \( \text{N}_1 \) are the number of zero and number of one entries in the output column of the truth table of the function.

Assuming that the \( N \) inputs are independent and uniformly distributed (the \( 2^N \) possible states are equally likely):

\[
P_{0\rightarrow1} = \frac{\text{N}_0}{2^N} \frac{\text{N}_1}{2^N} = \frac{\text{N}_0 \left(2^N - \text{N}_0\right)}{2^{2N}}
\]
NOR Gate Transition Probabilities

- Switching activity is a strong function of the input signal statistics
  - $P_A$ and $P_B$ are the probabilities that inputs A and B are one

\[ P_0 \rightarrow 1 = P_0 \times P_1 = (1-(1-P_A)(1-P_B)) \times (1-P_A)(1-P_B) \]

\[ P_1 = (1-P_A)(1-P_B) \]
### Transition Probabilities for Some Basic Gates

<table>
<thead>
<tr>
<th>Gate</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR</td>
<td>((1 - (1 - P_A)(1 - P_B)) \times (1 - P_A)(1 - P_B))</td>
</tr>
<tr>
<td>OR</td>
<td>((1 - P_A)(1 - P_B) \times (1 - (1 - P_A)(1 - P_B)))</td>
</tr>
<tr>
<td>NAND</td>
<td>(P_A P_B \times (1 - P_A P_B))</td>
</tr>
<tr>
<td>AND</td>
<td>((1 - P_A P_B) \times P_A P_B)</td>
</tr>
<tr>
<td>XOR</td>
<td>((1 - (P_A + P_B - 2P_A P_B)) \times (P_A + P_B - 2P_A P_B))</td>
</tr>
</tbody>
</table>

For X: \(P_{0\rightarrow1} = P_0 \times P_1 = (1 - P_A) P_A\)

\[= 0.5 \times 0.5 = 0.25\]

For Z: \(P_{0\rightarrow1} = P_0 \times P_1 = (1 - P_X P_B) P_X P_B\)

\[= (1 - (0.5 \times 0.5)) \times (0.5 \times 0.5) = 3/16\]
Inter-signal Correlations

- Determining switching activity is complicated by the fact that signals exhibit correlation in space and time.
  
  If B is not connected to A: \( P(Z=1) = P(B=1, C=1) = 1/4 \)

Reconvergent fan-out

\[
P(Z=1) = P(C=1 | B=1) \cdot P(B=1) = 0
\]

- Have to use **conditional probabilities**
Glitching in a chain of NAND gates
Balanced Delay Paths to Reduce Glitching

- Glitching is due to a mismatch in the path lengths in the logic network; if all input signals of a gate change simultaneously, no glitching occurs.

So equalize the lengths of timing paths through logic.
Logic Restructuring

- Logic restructuring: changing the topology of a logic network to reduce transitions

AND: \( P_{0\rightarrow1} = P_0 \times P_1 = (1 - P_A P_B) \times P_A P_B \)

\[
\begin{align*}
0.5 & \quad (1-0.25)\times0.25 = 3/16 \\
 A & \quad \quad W \\
 B & \quad \quad 7/64 \\
 C & \quad \quad X \\
 D & \quad \quad 15/256 \\
0.5 & \quad F \\
\end{align*}
\]

Chain implementation has a lower overall switching activity than the tree implementation for random inputs

Ignores glitching effects
Input Ordering

Beneficial to postpone the introduction of signals with a high transition rate (signals with signal probability close to 0.5)

Time – multiplexing resources may significantly effect switching activity!
Ratioed Logic

Resistive Load

(a) resistive load

Nominal $V_{OH} = V_{DD}$ but nominal $V_{OL} \neq 0$

Goal: to reduce the number of devices over complementary CMOS
(At the cost of reduced robustness and extra power)
Ratioed Logic

- N transistors + Load
- \( V_{OH} = V_{DD} \)
- \( V_{OL} = \frac{R_{PN}}{R_{PN} + R_L} V_{DD} \)
- Assymetrical response
- Static power consumption
- \( t_{pL} = 0.69 \ R_L \ C_L \)
Pseudo-NMOS

Pseudo-NMOS NOR Gate

Smaller area & Load but Static power dissipation!!

To obtain $V_{OL}$, equate the currents at load (PMOS) and driver (NMOS), when $V_{in} = V_{DD}$ It is reasonable to assume NMOS at linear (output should be close to 0V), PMOS at saturation

$$k_n \left( (V_{DD} - V_{Tn}) V_{OL} - \frac{V_{OL}^2}{2} \right) + k_p \left( -(V_{DD} - V_{Tp}) V_{DSATp} - \frac{V_{DSATp}^2}{2} \right) = 0$$
Assuming that $V_{OL} \ll (V_{DD}-V_T)$, $V_{DSATp} \ll (V_{DD}-V_T)$
and $|V_{Tn}| = |V_{Tp}|$

$$V_{OL} \approx \frac{k_p (V_{DD} + V_{Tp})}{k_n (V_{DD} - V_{Tn})} \approx \frac{\mu_p W_p}{\mu_n W_n} V_{DSATp}$$

In order to make $V_{OL}$ small, PMOS must be sized much smaller than NMOS. But this has negative effect on the propagation delay. A major disadvantage of the pseudo-NMOS is the static power dissipation when the output is low

$$P_{static} = V_{DD} I_{low} \approx V_{DD} \left| k_p \left( (-V_{DD} - V_{Tp}) V_{DSATp} - \frac{V_{DSATp}^2}{2} \right) \right|$$
Pseudo-NMOS VTC

INVERTER

<table>
<thead>
<tr>
<th>W/L_p</th>
<th>V_oL</th>
<th>P (μW)</th>
<th>t_{ph}(ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.69</td>
<td>564</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>0.27</td>
<td>298</td>
<td>56</td>
</tr>
<tr>
<td>1</td>
<td>0.13</td>
<td>160</td>
<td>123</td>
</tr>
<tr>
<td>0.5</td>
<td>0.06</td>
<td>80</td>
<td>268</td>
</tr>
<tr>
<td>0.25</td>
<td>0.03</td>
<td>41</td>
<td>569</td>
</tr>
</tbody>
</table>

\[
V_{\text{out}} [\text{V}] \\
W/L_p = 4
\]

\[
W/L_p = 2
\]

\[
W/L_p = 0.5
\]

\[
W/L_p = 0.25
\]
Improved Loads (Differential Logic & Positive Feedback)

Differential Cascode Voltage Switch Logic (DCVSL)
**DCVSL Example**

Out \( = AB' + A'B \)

Out' \( = AB + A'B' \)

**XOR – XNOR Gate**
$Out = \overline{A \overline{B}}$

$\overline{Out} = A \overline{B}$

$A \rightarrow M_1 \quad \overline{A} \rightarrow M_3 \overline{B} \quad M_4$

$B \rightarrow M_2$

Graph showing Voltage (V) vs. Time (ns):

- $\overline{A \overline{B}}$
- $\overline{A, \overline{B}}$
- $A \overline{B}$
(a) Single ended

(b) Differential
NMOS Transistors in Series/Parallel

- Primary inputs drive both gate and source/drain terminals
- NMOS switch closes when the gate input is high

![Diagram of NMOS transistors in series/parallel]

- X = Y if A and B
- X = Y if A or B

- Remember - NMOS transistors pass a strong 0 but a weak 1
PMOS Transistors in Series/Parallel

- Primary inputs drive both gate and source/drain terminals
- PMOS switch closes when the gate input is low

\[
X = Y \text{ if } \overline{A} \text{ and } \overline{B} = A + B
\]

\[
X = Y \text{ if } \overline{A} \text{ or } \overline{B} = A \bullet B
\]

- Remember - PMOS transistors pass a strong 1 but a weak 0
Pass-Transistor Logic

- \( N \) transistors
- No static consumption
Example: AND Gate
Pass Transistor (PT) Logic

- Gate is static – a low-impedance path exists to both supply rails under all circumstances
- N transistors instead of 2N
- No static power consumption
- Ratioless
- Bidirectional (versus undirectional)
NMOS Only PT Driving an Inverter

- $V_x$ does not pull up to $V_{DD}$, but $V_{DD} - V_{Tn}$

- Threshold voltage drop causes static power consumption ($M_2$ may be weakly conducting forming a path from $V_{DD}$ to GND)

- Notice $V_{Tn}$ increases of pass transistor due to **body effect** ($V_{SB}$)
Voltage Swing of PT Driving an Inverter

- **Body effect** – large $V_{SB}$ at $x$ - when pulling high (B is tied to GND and S charged up close to $V_{DD}$)

- So the voltage drop is even worse

$$V_x = V_{DD} - (V_{Tn0} + \gamma(\sqrt{|2\phi_f| + V_x} - \sqrt{|2\phi_f|}))$$
Pass transistor gates should *never* be cascaded as on the left.

Logic on the right suffers from static power dissipation and reduced noise margins.
VTC of the pass transistor AND gate

Assuming $V_M$ (of the inverter) = $V_{DD}/2$

Pure pass-transistor gate is not regenerative
Differential PT Logic (CPL)

- **PT Network**
  - Input: A, B
  - Output: F

- **Inverse PT Network**
  - Input: A, B
  - Output: F

- **AND/NAND**
  - Output: $F = AB$

- **OR/NOR**
  - Output: $F = A + B$

- **XOR/XNOR**
  - Output: $F = A \oplus B$
4-input AND/NAND gate in CPL
LEVEL RESTORATION

When A goes high, all nodes are either 0 or \( V_{DD} \). To pull node X down, \( M_n \) need to be stronger than \( M_r \). That requires careful transistor sizing (if \( R_r \) is too small with respect to \( R_n \), it is impossible to bring \( V_x \) below the switching threshold of the inverter).
Transmission Gates (TGs)

- Most widely used solution for level restoration

- Full swing bidirectional switch controlled by the gate signal $C$, $A = B$ if $C = 1$
Transmission gate 2-input inverting multiplexer

\[ (A\overline{S} + B\overline{S}')' \]
Transmission gate XOR

\[ F = A'B + AB' \]
CPL Properties

- Differential so complementary data inputs and outputs are always available (so don’t need extra inverters)
- Still static, since the output defining nodes are always tied to $V_{DD}$ or GND through a low resistance path
- Design is modular; all gates use the same topology, only the inputs are permuted.
- Simple XOR makes it attractive for structures like adders
- Fast (assuming number of transistors in series is small)
- Additional routing overhead for complementary signals
- Still have static power dissipation problems