Dynamic CMOS

- In static circuits at every point in time (except when switching) the output is connected to either GND or VDD via a low resistance path.
  - fan-in of N requires 2N devices
- Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.
  - requires on N + 2 transistors

Conditions on Output

- Once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.
- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on CL

Properties of Dynamic Gates

- Logic function is implemented by the PDN only
  - number of transistors is N + 2 (versus 2N for static complementary CMOS)
- Full swing outputs (V_{OL} = GND and V_{OH} = VDD)
- Nonratioed - sizing of the devices is not important for proper functioning
- Faster switching speeds
  - reduced load capacitance due to lower input capacitance (C_{in})
  - reduced load capacitance due to smaller output loading (C_{out})
  - no I_{ss}, so all the current provided by PDN goes into discharging C_{L}

Dynamic Gate

Two phase operation
Precharge (CLK = 0)
Evaluate (CLK = 1)
Properties of Dynamic Gates, con’t

- Overall power dissipation usually significantly higher than static CMOS
  - no static current path ever exists between $V_{DD}$ and GND (including $P_{on}$)
  - no glitching
  - higher transition probabilities
  - extra load on CLK
- PDN starts to work as soon as the input signals exceed $V_{TN}$, so set $V_{TH}$ and $V_{IL}$ equal to $V_{TN}$
- low noise margin (NM)
- Needs a precharge clock

Issues in Dynamic Design 1: Charge Leakage

PDN starts to work as soon as the input signals exceed $V_{TN}$, so set $V_{TH}$ and $V_{IL}$ equal to $V_{TN}$

Power only dissipated when previous Out = 0

Solution to Charge Leakage

Same approach as level restorer for pass transistor logic

Dynamic Power Consumption is Data Dependent

Dynamic 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>

Assume signal probabilities
- $P_{A=1} = 1/2$
- $P_{B=1} = 1/2$

Then transition probability
- $P_{0 \rightarrow 1} = P_{out=0} \times P_{in=1}$
  - $= 3/4 \times 1 = 3/4$

Switching activity always higher in dynamic gates!
- $P_{0 \rightarrow 1} = P_{out=0}$

Issues in Dynamic Design 2: Charge Sharing

Charge stored originally on $C_L$ is redistributed (shared) over $C_L$ and $C_g$, leading to static power consumption by downstream gates and possible circuit malfunction.
**Charge Sharing Example**

**Charge Sharing**

\[ V_{DD} \]

- **case 1)** if \( \Delta V_{out} < \Delta V_{sense} \)
  \[ C_L V_{DD} = C_L V_{out}(t) + C_A V_{DD} - V_{Th} \]
  or
  \[ \Delta V_{out} = V_{out}(t) - V_{DD} = C_A V_{DD} - V_{Th} \]

- **case 2)** if \( \Delta V_{out} > \Delta V_{sense} \)
  \[ \Delta V_{out} = -V_{DD} \]

**Backgate Coupling Effect**

**Solution to Charge Redistribution**

- Precharge internal nodes using a clock-driven transistor (at the cost of increased area and power)

**Issues in Dynamic Design 3:**

**Backgate Coupling**

**Issues in Dynamic Design 4:**

**Clock Feedthrough**

Coupling between Out and CLK input of the precharge device due to the gate to drain capacitance. So voltage of Out can rise above \( V_{DD} \). The fast rising (and falling edges) of the clock couple to Out.
Clock Feedthrough

Other Effects
- Capacitive coupling
- Substrate coupling
- Minority charge injection
- Supply noise (ground bounce)

Why Domino?
- Like falling dominos!

Cascading Dynamic Gates
- Only 0 → 1 transitions allowed at inputs!

Domino Manchester Carry Chain
Properties of Domino Logic

- Only non-inverting logic can be implemented
- Very high speed
  - Static inverter can be skewed, only L-H transition
  - Input capacitance reduced – smaller logical effort

Designing with Domino Logic

Logical effort

Logical Effort

Logical Effort
### Differential (Dual Rail) Domino

\[ \overline{\text{Out}} = \overline{\overline{\text{AB}}} \]

Solves the problem of non-inverting logic

### np-CMOS Adder Circuit

### np-CMOS

Only $0 \to 1$ transitions allowed at inputs of PDN
Only $1 \to 0$ transitions allowed at inputs of PUN

### Dynamic CVS Logic

PDN1 and PDN2 are mutually exclusive

### NORA Logic

WARNING: Very sensitive to noise!

### How to Choose a Logic Style

- Must consider area, performance, power, robustness (noise immunity), ease of design, system clocking requirements, fan-out, functionality, ease of testing

<table>
<thead>
<tr>
<th>Style</th>
<th># Trans</th>
<th>Ratioed</th>
<th>Delay</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp Static</td>
<td>8</td>
<td>no</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CPL*</td>
<td>8</td>
<td>no</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>domino</td>
<td>6 + 2</td>
<td>no</td>
<td>2</td>
<td>2 + clk</td>
</tr>
<tr>
<td>DCVSL*</td>
<td>10</td>
<td>yes</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

* Dual Rail