12. Dynamic CMOS Logic

J. A. Abraham

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

VLSI Design
Fall 2015

October 12, 2015
Dynamic gates use a clocked pMOS pullup

Two modes of operation: precharge and evaluate
The “Foot” Transistor

- What if pulldown network is ON during precharge?
- Use series evaluation transistor to prevent fight between pMOS and nMOS transistors
Logical Effort

Inverter

unfooted

footed

NAND2

NOR2

\[ g_d = \frac{2}{3}, \quad p_d = \frac{3}{3} \]

\[ g_d = \frac{2}{3}, \quad p_d = \frac{3}{3} \]

\[ g_d = \frac{3}{3}, \quad p_d = \frac{4}{3} \]

\[ g_d = \frac{2}{3}, \quad p_d = \frac{5}{3} \]
Monotonicity

Dynamic gates require **monotonically rising** inputs during evaluation:

- 0 → 0
- 0 → 1
- 1 → 1
- But *not* 1 → 0

Diagrams illustrating the precharge, evaluate, and postcharge phases, with an example of an output that does not rise as expected, violating monotonicity during evaluation.

![Dynamic CMOS Logic Diagram](image-url)
Monotonicity Woes

- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!
Domino Gates

- Follow dynamic stage with inverting static gate
  - Dynamic/static pair is called domino gate
  - Produces monotonic outputs
Domino Optimizations

- Each domino gate triggers next one, like a string of dominos toppling over
- Gates evaluate sequentially, precharge in parallel
- Evaluation is more critical than precharge
- HI-skewed static stages can perform logic
Dual-Rail Domino

- Domino only performs noninverting functions:
  - AND, OR but **not** NAND, NOR, or XOR
- Dual-rail domino solves this problem
  - Takes true and complementary inputs
  - Produces true and complementary outputs

<table>
<thead>
<tr>
<th>sig_h</th>
<th>sig_l</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Precharged</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>‘0’</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>‘1’</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Invalid</td>
</tr>
</tbody>
</table>

Diagram:
- Input signals and logic gates with precharge and active clock signals.
Example: AND/NAND

- Given $A_h, A_l, B_h, B_l$
- Compute $Y_h = A \times B$, $Y_l = \sim(A \times B)$
- Pulldown networks are conduction complements
**Example: XOR/XNOR**

- Sometimes possible to share transistors
  - Sharing works well in implementations of symmetric functions
  - See papers on “relay logic” published over 50 years ago
Leakage

- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{off} \neq 0$)
  - Dynamic value will leak away over time
  - Formerly milliseconds, now nanoseconds!
- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation
- Leakage Power!

![Diagram of weak keeper circuit]
Charge Sharing

- Dynamic gates suffer from charge sharing

\[ V_x = V_y = \frac{C_y}{C_x + C_y} V_{DD} \]
Secondary Precharge

- Solution: add secondary precharge transistors
  - Typically need to precharge every other node
- Big load capacitance on Y helps as well

Diagram:
- φ
- A
- B
- Secondary precharge transistor on Y
Noise Sensitivity

- Dynamic gates are very sensitive to noise
  - Inputs: $V_{IH} \approx V_{tn}$
  - Outputs: floating output susceptible noise
- Noise sources
  - Capacitive crosstalk
  - Charge sharing
  - Power supply noise
  - Feedthrough noise
  - And more!

Chip power supply voltage map when executing a program
Dynamic CVSL XOR Gate

\[ Q = A \oplus B \oplus C \oplus D \]
Dual-Rail Domino Full Adder Design

- Very fast, but large and power hungry
- Used in very fast multipliers
“Manchester” Adders

Dynamic Stage

Static Stage

Multiplexer Stage

Propagate-Generate Logic

Sum Logic
Domino logic is attractive for high-speed circuits
  - 1.5x 2x faster than static CMOS

Many Challenges
  - Monotonicity
  - Leakage
  - Charge sharing
  - Noise

Used in previous generation high-performance microprocessors and in some recent embedded processors
Domino Logic in Current Designs

- Domino design from Intrinsity used in 1-GHz 0.75W ARM Cortex A8 from Samsung (Intrinsity later acquired by Apple)
- Fast Domino (called “Fast14 NDL”) gates are inserted selectively into critical speed paths, with custom SRAMs and optimized synthesized logic elsewhere
- Standard power saving techniques are also used
- Domino gates are clocked by multiphase clocks
- A type of “super-pipeline” where the domino footers form the barrier for the pipeline operation

(Source: *Electronic Design – Embedded*, August 29, 2009)
Intrinsity OR/NOR Implementation with “N-nary Logic”

2-bit function using 1-out-of-4 signals

Ref: U. S. Patent 6066965, Method and apparatus for a N-nary Logic Circuit Using 1 of 4 Signals
Intrinsity XOR/Equivalence Implementation

Using 1-out-of-2 signals

Ref: U. S. Patent 6066965, Method and apparatus for a N-nary Logic Circuit Using 1 of 4 Signals