Lecture 6-B
Dynamic Logic

C.K. Ken Yang
University of California at Los Angeles
yang@ee.ucla.edu

Overview

Reading
– W&H: 6.2, 7.5.1

Introduction
Logic gates can be faster if we leverage the dynamic storage nature of the nodes in CMOS technology. Instead of pulling both up and down with the gate, we can reset the gate into the HIGH state and optionally pull down.

The logic is no longer static and purely input driven. A clock signal alternately resets and enables the logic. The logic is evaluated and the data is propagated only during a portion of the clock cycle. This style is known as dynamic logic. The specific type that this lecture addresses is Domino Logic.
Dynamic Logic

- Since clocks can be used to sequence events, why not use clocks to simplify logic.
  - One limitation of Static CMOS is that 2 switch networks are needed: pull-up and pull-down.
  - PMOS are not as efficient, so pull-up network is costly.
- We only need pull-down network, if we initialize the output HIGH.
  - Use clock to initialize.
- Two phases of operation:
  - Precharge – initialize the output
  - Evaluate – pull-down (depending on the logical function).
- Logic function may not discharge output, so output may be floating (DYNAMIC)

---

Domino Logic

- A robust type of dynamic logic.
  - Pull-down network turns on when inputs are above $V_{TN}$.
  - If inputs are from a floating node (another dynamic gate), the logic can be VERY sensitive to noise.
- Use an static CMOS gate (an inverter) after the dynamic gate.
  - Separates dynamic node from input
    - Dynamic node needs to be corrupted by nearly $V_{THL}$ to flip.
  - CMOS gate has strong drive (inputs less sensitive to noise)
Domino 2-Input AND Gate Example

- Only the pull-down network needs to be built. G=AB.
  - Pull-down function, F = G, because inverter inverts the output.
- Assuming that inputs are from other Domino circuits.
  - B arrives last and discharges DS.
  - DS causes OUT to transition HIGH.
  - The only available output transition is from L-H.
- Precharge device can be small (minimize capacitance).
  - Delay before OUT precharges LOW.

![Diagram of a Domino 2-Input AND Gate Example]

Domino Effect

- Connect many gates to be evaluated in a single cycle.
  - Talk about clocking methodology later.
- Reason for the name is because earlier gates trigger the later gates like dominos.
- Result from last gate must be latched (held) before the precharge occurs.
  - Otherwise data is lost.
  - Latched data can feed into another domino chain that is clocked by φb.
Speed, Power and Robustness

- Many design decisions that involve significant tradeoffs.
- Evaluate Transistor
  - If inputs are always from other domino gates, then start L.
    - Why do we need the extra clocked transistor at the bottom?
      - Improve evaluation speed by removing it.
      - But increase power dissipation during precharge. Why?
- Keepers
  - Add a small device to pull up DS when output is LOW.
    - To keep DS from being floating.
  - Fights with the pull-down network
    - Slightly more power and slower.

---

Monotonicity

- Domino outputs can only be L-H.
  - Inputs from other domino gates are only L-H.
  - Can a domino input transition from H-L?
- If input is High, it discharges the pre-charge node.
  - Transitioning Low would not impact the output.
    - If intend to reflect a LOW input, then there would be an error.
- Non-inverting logic family!
  - Makes sense because a single Domino gate is effectively 2 gates: a dynamic and a static.
    - Each gate inverts once.
- A non-inverting family of logic is incomplete.
  - How do you invert?
Dual Rail Domino

- To enable ALL possible functions, Domino must guarantee both true and complement to be available as inputs.
  - Each stage must then generate both true and complement DOMINO outputs.
  - Not actual complementary signals because inputs must be monotonically increasing during evaluate.
    - Can’t just use an inverter.

Domino Logic Chain

- Logic is still being done during pre-charge
- Use a second phase, $\phi_2$, that pre-charges during the eval of $\phi_1$.
- Evaluated data from $\phi_1$ passes to $\phi_2$.
  - Should hold the data constant for a while (more later).
- Overlapping phases is actually o.k. (guarantees handoff).
### Interfacing with Domino Gates

Several rules for interfacing with Domino Gates

- The inputs must be either monotonically increasing.
- Or a fixed input is considered o.k.
  - Interface with a non-precharged Static CMOS gate is o.k. as long as the output does not change during evaluate.
    - Use a latch
- First domino stage at the interface should have an evaluate device.
  - Input may go away during pre-charge.

### Examples of Building a Dual-Rail Domino Gate

- Need to build double the pull-down network.
  - Dual networks with inverted inputs.
  - Lose some of the gain over Static CMOS.
  - But can potentially merge the networks (as in XOR)
Example of Domino Delay Calculation

- $R_{N_{DN}} = 3 \Omega \cdot \mu \text{m}$, $R_{P_{UP}} = 7.5 \Omega \cdot \mu \text{m}$, $C_{DN} = C_{DP} = 2 fF/\mu \text{m}$, $C_{GN} = C_{GP} = 2 fF/\mu \text{m}$
- Treat it as 2 gate delays
  - Static Inverter, $R_{INV} = 1.25 k\Omega$, $C_{INV} = 16 fF$, $C_L = 12 fF$
    - $T_{INV} = 35 ps$
  - Dynamic, $R_{A,B} = 0.5 k\Omega$, $C_{N1} = 12 fF$, $C_{O} = 8*2 fF + 12*2 fF = 40 fF$
    - $T_{DYN} = 0.5*12 + 1*40 = 46 ps$

Total Delay = 81 ps

What Makes Domino Fast?

Even though there are 2 gates, Domino Gates are faster than static CMOS.
- Gate load driven is only NMOS tree
  - $1/(\beta + 1)$ of the Static Gate Capacitance
- Self-loading driven is only NMOS tree.
  - Precharge is significantly smaller than a PMOS tree.
- Inverter output only requires $V_{TN}$ to turn on the pull-down path.
- Inverter is often skewed in size to have HIGH $V_{THL}$ to speed up the transition.
- Denser logic
  - Replace inverter with simple 2-input NAND.
  - Slightly deeper stacking.
**Charge Sharing**

- Domino gates are particularly sensitive to charge sharing.
- During evaluate, the precharged node is floating and can charge share depending on the input transitions.

\[
\begin{align*}
V_{\text{final}} &= \frac{V_{DD}C_{out}}{C_p + C_{out}} \\
V_{CP} &\text{ cannot be } > V_{DD} - V_{TN}
\end{align*}
\]

If \( C_p \) is large enough, this can cause logic to flip. Otherwise, it can still cause weakened logic levels.

Add extra precharge to fix the problem.

**Domino Charge Sharing Example**

- Charge sharing worst case depends on the input sequence
- Example: AB+C
  - \( C_C = 40fF, \ C_N = 12fF \)
  - If \( C_N \) is discharged while \( C_O \) is precharged and A transitions L-H.

\[
\Delta V_{DS} = V_{DD}(C_O/(C_N+C_O)) = 0.76V_{DD}
\]

Input sequence
1st cycle: B=L-H, A=X, C=X
2nd cycle: B=L, C=L, A=L-H
Domino Power Dissipation

- Because clock is driven to each gate
  - Significant Clock Loading
    - Use smallest precharge device as possible.
    - Remove evaluate every 2 or 3 gates is a power saving.
  - Typically higher power dissipation for the higher performance.
- One signal (true or complement of dual rail domino) transitions every clock transition.
  - $C V^2 f$
    - $C$ of clock loading and $C$ of one of the two outputs (typically approximately the same)
  - Little dependence on activity factor (only power from internal capacitance of a pull-down network)

Summary

- Instead of statically driving both high and low, we can build a network that “conditionally” pulls-down depending on the input.
  - The output can be pre-charged high and held there dynamically.
  - Domino logic
    - Follow a dynamic gate with an inverter to improve signal integrity
    - Faster because the logic eliminates the PMOS network and emphasizes one transition.
    - Need dual rail for logical inversions.