Lecture 10:

Skew Tolerant Domino Clocking

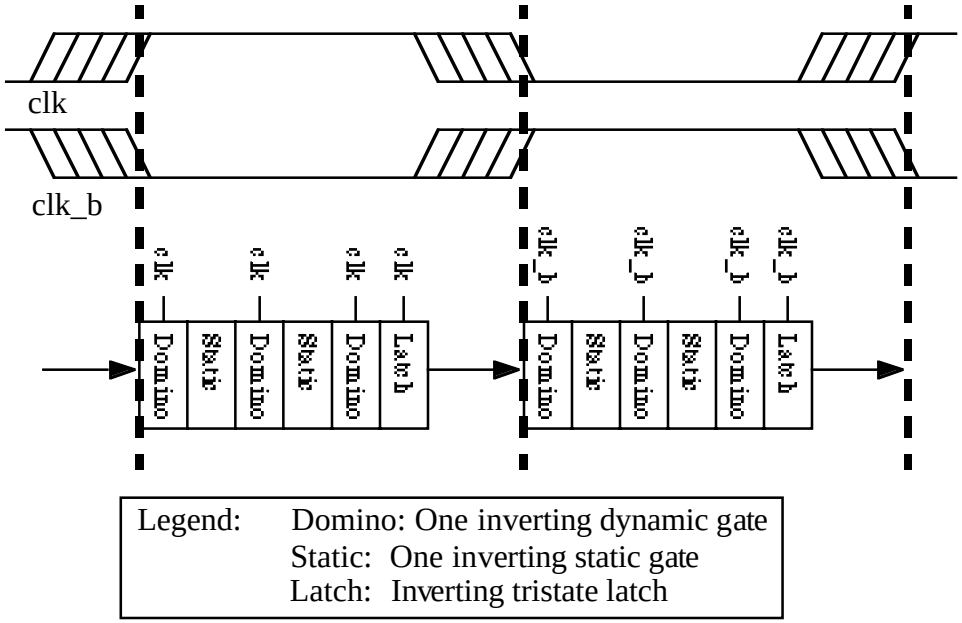
Computer Systems Laboratory
Stanford University
horowitz@stanford.edu

Introduction

- Domino Circuits are becoming ubiquitous in high speed digital ICs
  - Offer 30% (or more) speedup over static CMOS raw gate delay
  - Dual-rail domino becoming more common because many functions are nonmonotonic, area is less of an issue
- Nevertheless, traditional domino pipelines have significant overhead
  - Latch required to hold result while next stage evals, prev. precharges
  - Skew budget, no time borrowing, latch delay
- Look at several ways to reduce this overhead
  - Better latches, Self-timing
  - Skew-tolerant domino is a powerful new technique
- Evaluate performance benefits of skew-tolerant domino

Domino from a System Perspective

- Domino doesn't look so attractive in the context of a traditional pipeline



1. Pay clock skew twice each phase
2. Balancing short phases is hard since there is no time borrowing
3. Latches become a significant fraction of the cycle time

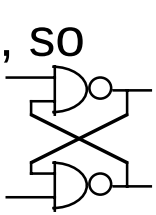
Traditional Domino Performance Evaluation

- Let T = cycle time = 20 FO4 delays; t_{skew} = 2; t_{setup} = 1[1]
  - Difficult filling cycle exactly (no time borrowing) -> $t_{imbalance}$ = 1
- $T_{phase\text{-}logic} = T/2 - t_{skew} - t_{setup} - t_{imbalance}$
- Baseline Design:
  - $T_{phase\text{-}logic} = 20/2 - 2 - 1 - 1 = 6$
  - 40% of the phase is wasted in overhead! Slower than static!
- Optimized Design:
  - Define clock domains and use $t_{skew\text{-}local}$ = 1
  - Work hard to balance logic between phases: $t_{imbalance}$ = 0 (optimistic)
  - $T_{phase\text{-}logic} = 20/2 - 1 - 1 - 0 = 8$
  - Still, 20% of the phase is overhead!

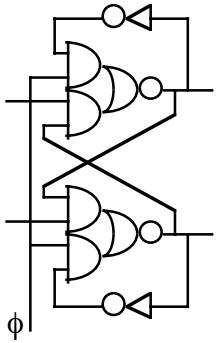
1. Remember for this situation, the setup time must be large enough that the output has settled before clock arrives since the output might go into a dynamics gate on the next cycle and might not be monotonic

Early Enhancements

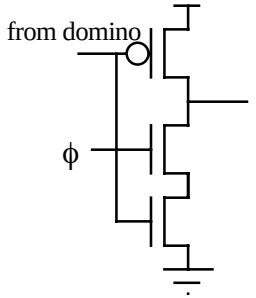
- Good designers have recognized this problem for years.
- The largest problem is the hard edges set by the latches.
- A variety of latches soften this edge:
  - Gate outputs are already _q1, so why use another clock. An SR latch will work instead
  - Use the monotonic nature of the signal to feed it into a precharged latch stage



SR Latch



Dual-Monotonic Latch



TSPC Latch

- Still have a problem if you want to use non-monotonic logic somewhere, since logic must settle before earliest clock, while gate might not evaluate unitl a late clock

- But if you only have monotonic gates ...

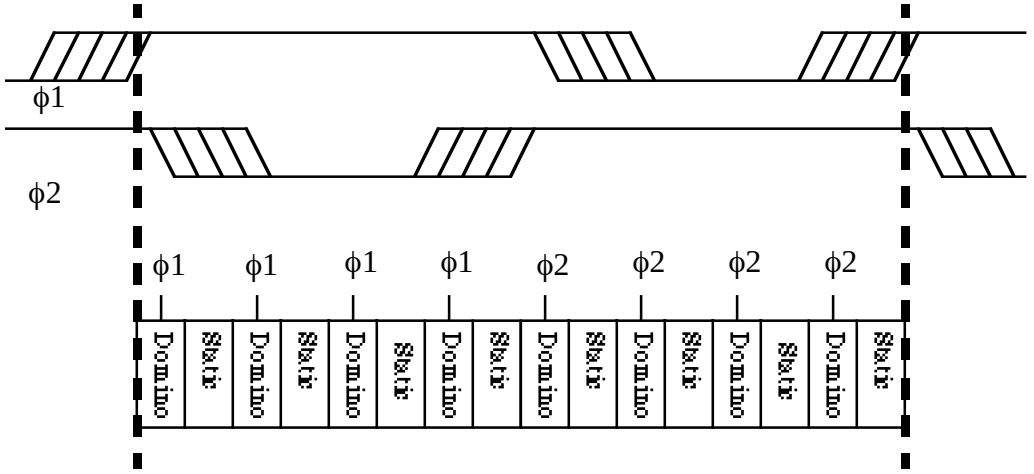
Skew Tolerant Domino Clocking

- If inputs are all dual rail, then as long as the clock arrives before the data,
  - The gate will wait and fire when the data arrives
- If the next gate fires before the current gate precharges,
  - There is no need for a latch
- Like the self-timed pipeline

- Can generate these properties using overlapping clocks

Skew-Tolerant Domino Circuits

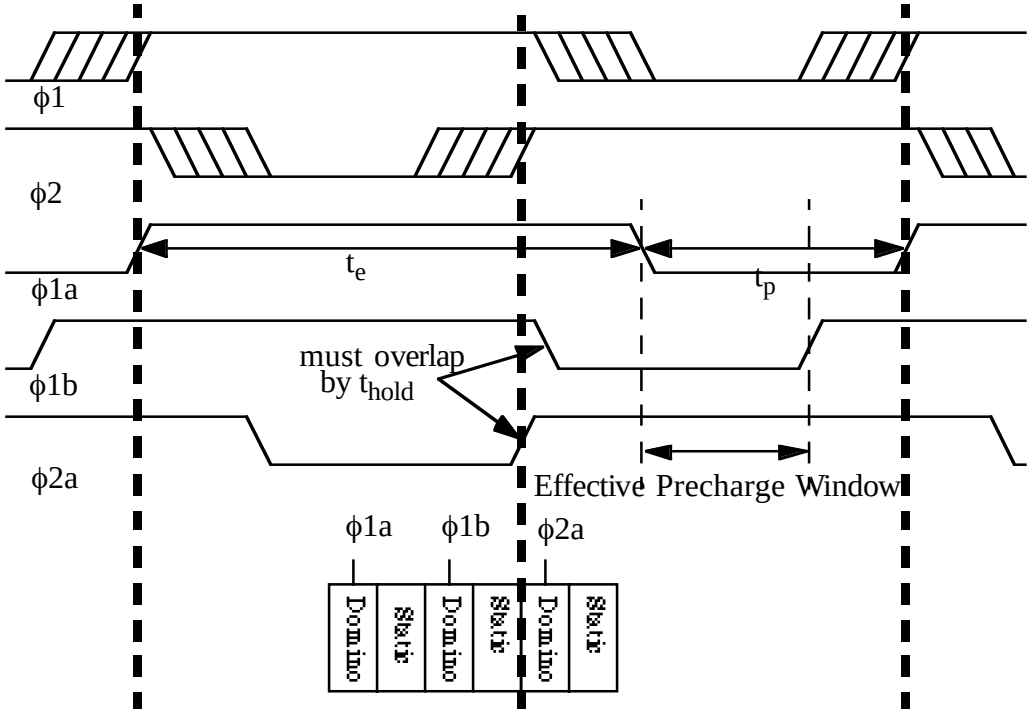
- How much clock skew could we tolerate given N clock phases?
  - Divide logic into N phases of T/N duration each.
  - Overlapping clocks eliminates need for latches
  - Extra overlap accommodates clock skew and time borrowing



- As with other domino techniques, budget skew on the transition from static to domino

Skew Tolerance

- $T = t_e + t_p$
- $t_p = t_{prech} + t_{skew}$; $t_e = T/N + t_{skew} + t_{hold}$
- Hence $t_{skew\text{-}max} = [T(N-1)/N - t_{prech} - t_{hold}] / 2$



Numerical Example

- Let t_{prech} = 4, long enough to:
- precharge domino gate
- make subsequent skewed static fall below V_t
- t_{hold} is slightly negative for reasonable cell libraries
- next phase can evaluate before precharge ripples through static gate
- conservatively bound t_{hold} at 0

N	t_{skew}	t_p
2	2	6
3	3.33	7.33
4	4	8
6	4.66	8.66
8	5	9

- Sweet spots: N=2 (fewest clocks), N=4 (good tolerance, 50% duty cycle)

Global & Local Skew

- This is good, but we can do better!
- Local skew can be more tightly controlled than global skew (~ 1 FO4)
  - Require that each phase of logic fit in a local clock domain:
- $t_p = t_{prech} + t_{skew\text{-}local}$; $t_e = T/N + t_{skew\text{-}global} + t_{hold}$
- Hence $t_{skew\text{-}global\text{-}max} = T(N\text{-}1)/N - t_{skew\text{-}local} - t_{prech} - t_{hold}$
- When $t_{skew\text{-}global}$ gets huge, precharge interferes with subsequent phase

N	$t_{skew\text{-}global}$	t_p
2	2	5
3	5.66	5
4	6	6
6	6	7.33
8	6	8

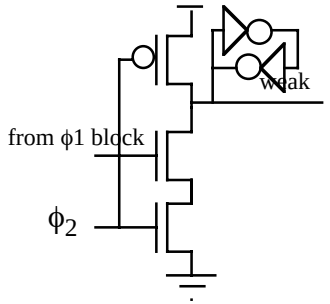
Time Borrowing

- We don't need such a large global skew tolerance!
  - Use some of this time instead to allow time borrowing
- $t_{borrow} = T(N-1)/N - t_{skew\text{-}global} - t_{skew\text{-}local} - t_{prech} - t_{hold}$
- Intentional borrowing helps balance logic between phases
- Opportunistic time borrowing compensates for uncertainties in models, analysis tools, and processing
- If actual $t_{skew\text{-}global} = 2$, $t_{skew\text{-}local} = 1$:

N	t_{borrow}	t_p
2	1	5
3	3.66	5
4	5	5
6	6..33	5
8	7	5

Other Design Issues

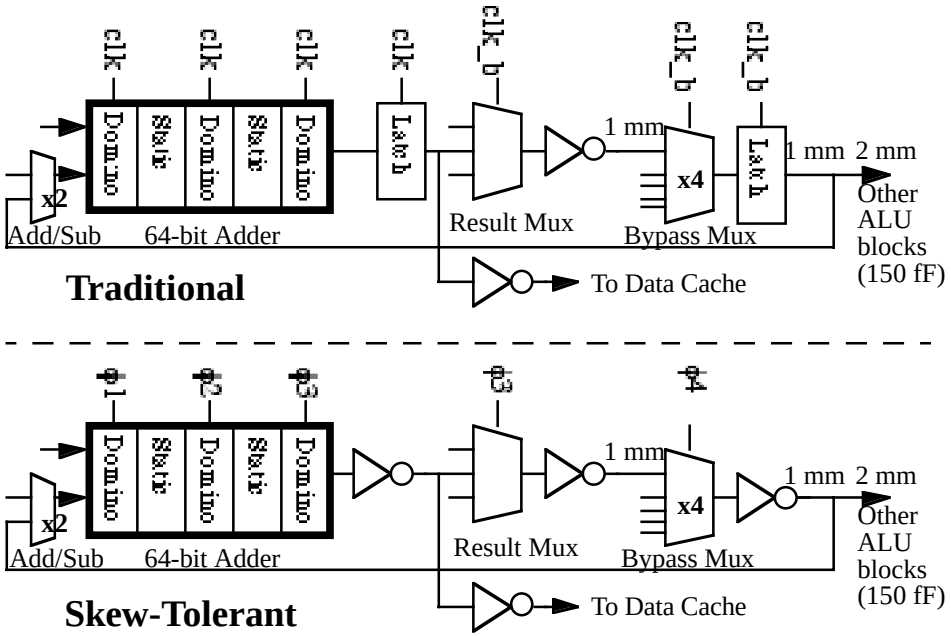
- State is no longer stored in the latch at the end of a phase
  - Instead, it is held by the first domino gate in the phase
  - Use a “full keeper” to allow stop-clock operation



- All systems with overlapping clocks require min-delay checks
  - Domino paths are presumably critical anyway, so few min-delay errors
  - 4-phase has effectively no min-delay risk
    - Overlap of all four phases is at most very small
    - A minimum of 8 gates are in the cycle anyway

Skew-Tolerant Performance Evaluation

- Evaluate ALU self-bypass of superscalar μproc (like DEC Alpha)
- 3-metal 0.6 μm process
- FO4 delay in TT corner = 138 ps
- Compare traditional domino to 4-phase skew-tolerant domino



Simulation Results

- No Skew:
  - Traditional Domino: Latency = 13.0 FO4, cycle time = 16.6
    - Cycles are unbalanced; no time borrowing available
  - Skew-Tolerant Domino: Latency = 11.9 FO4, cycle time = 11.9
    - Remove latches from critical path, balance pipe stages
- 1 FO4 local skew:
  - Traditional Domino: Latency = 15.0 FO4, cycle time = 17.6
    - Skew adds to both phases for latency
    - Unbalanced second stage already has margin in the cycle time
  - Skew-Tolerant Domino: Latency = 11.9 FO4, cycle time = 11.9
    - Skew is tolerated

Summary

- Offers most of the benefits of self-timed designs while preserving the simplicity of a synchronous methodology.
- Clock generation & distribution becomes key issue. However, control generation and distribution can be just as tough in self-timed designs.
- Skew-Tolerant Domino eliminates most of the overhead found in traditional domino systems:
  - Tolerates clock skew
  - Removes latches from the critical path
  - Allows time borrowing
  - Robust
- High-performance microprocessor designs have used these ideas but they don't talk about them.