SP-25.7: Skew-Tolerant Domino Circuits

David Harris\(^1\) and Mark Horowitz

Stanford University  Stanford, CA

\(^1\)Also with Intel Corporation  Santa Clara, CA
Overview

General Domino Design
  • Domino performance
  • Domino overhead
  • Eliminating overhead

Skew-Tolerant Domino
  • Implementation issues
  • Performance results

Conclusions
**Ideal Domino Clocking**

Ping-pong between half-cycles

- One evaluates while other precharges
Clock Skew

- Evaluation begins at latest rising edge
- Latch input setup before earliest falling edge
- Clock skew twice each cycle
Balancing Logic

Logic may not exactly fit half-cycle

- No flexibility to borrow time
How bad is it?

Consider 500 MHz DEC Alpha:

- $T = 2\text{ns}$
- $t_{\text{skew}} = 200 \text{ps}$ (reported budget)
- $t_{\text{setup}} = 50 \text{ps}$ (optimistic estimate)

- 25% of cycle consumed by skew & setup
Prior Art

Assuming monotonic inputs to domino logic:

- Self-timed circuits (HaL Divider)
- Use better latch:

![TSPC Latch Diagram]

- Remove latch (Alpha 21164)
General Solution

How could Alpha remove latches?

Function of latches:

(1) Prevent glitches on domino input
(2) Hold result during precharge

Latches can be safely eliminated if:

(1) Input is from domino logic (no glitches)
(2) Result is consumed before precharge
Overlap clocks so $x$ evaluates before $y$ precharges

- We can remove latches within domino pipe
- Still budget skew from static $\rightarrow$ domino
Skew-Tolerant Waveforms

In general: $N$ overlapping clock phases

- Identical, each delayed $T/N$ from the previous
Precharge Time ($t_p$)

$Precharge: \ t_p = t_{prech} + t_{skew}$
Evaluation Time ($t_e$)

Overlap: $t_e = T/N + t_{skew} + t_{hold}$
Skew Tolerance

We’ve found the optimal duty cycle

- Precharge: \( t_p = t_{\text{prech}} + t_{\text{skew}} \)
- Overlap: \( t_e = T/N + t_{\text{skew}} + t_{\text{hold}} \)

Solve for the maximum tolerable skew

- \( t_{\text{skew-max}} = \frac{N-1}{N} T - t_{\text{prech}} - t_{\text{hold}} \)
- Skew tolerance increases with N
Local Skew & Time Borrowing

Local clock domains

- Reduce skew within a phase of logic
- Thereby to tolerate more skew globally

Time borrowing

- Clocks may overlap by more than global skew
- Overlap allows time borrowing into next phase
Example

Again consider 500 MHz DEC Alpha:

- \( T = 2\text{ns} \)
- \( t_{\text{skewG}} = 200 \text{ ps} / t_{\text{skewL}} = 50 \text{ ps} \)
- \( t_{\text{hold}} = 0 \) (conservative)

Available Time Borrowing vs. \# Clock Phases
(assuming \( t_{\text{prech}} = 500 \text{ ps} \))
2-Phase Clock Generation

Delay falling edge with clock chopper

Diagram showing a low-skew complement generator connected to clock choppers producing phases φ1 and φ2.
4-Phase Clock Generation

Inverters delay \( \phi_2 \) and \( \phi_4 \) by 1/4 cycle

- 4-phase has 1/4 cycle extra nominal overlap

\[
\text{GCLK#} \quad \text{\begin{array}{llllllllllllll}
\quad \\
\phi_1 & \phi_2 & \phi_3 & \phi_4
\end{array}}
\]

\text{Low-skew complement generator}

\text{1/4 Cycle Delay}

\text{Clock choppers}
Clock Generator Skew Analysis

How much skew does the clock generator cause?

- Variation in delay 20-30% relative to other clks
- 4-Phase is still a large net benefit
Other Design Issues

State stored in first gate of domino phase

- Use a “full keeper” to allow stop-clock

\[ \phi_1 \text{ block} \]

- Scan can be added to domino gate

Min-Delay

- Check when \( t_{skew} \) is large
Performance Evaluation

- 64 bit superscalar ALU self-bypass path

![Diagram of ALU with self-bypass path]
Simulation Results

- Skew-tolerant domino 25%+ faster
Conclusion

- Domino very attractive for high speed ICs
- Traditional domino limited by overhead
- Skew-tolerant domino eliminates overhead
- Clock generation
- Expect widespread use