Announcements

- Homework 2 due today
- Quiz #2 on Monday
- Midterm project report due next Wednesday
Outline

- Last lecture
  - ECC in SRAM
  - SRAM scaling options
- This lecture
  - Back to timing
  - Latch-based timing
  - Variability and timing

4. Design for performance

A. Timing
Flip-Flop Parameters

Delays can be different for rising and falling data transitions

Latch Parameters

Delays can be different for rising and falling data transitions

Unger and Tan
Trans. on Comp.
10/86
Clock Nonidealities

- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, $t_{sk}$
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) - $t_{js}$
  - Long-term - $t_{jl}$
- **Variation of the pulse width**
  - for level-sensitive clocking
Clock Skew and Jitter

- Both skew and jitter affect the effective cycle time
- Only skew affects the race margin, if jitter is from the source
- Distribution-induced jitter affects both

Clock Uncertainties

- Sources of clock uncertainty
  - Clock Generation
  - Clock Uncertainties
    - Devices
    - Power Supply
    - Interconnect
    - Capacitive Load
    - Temperature
    - Coupling to Adjacent Lines
Clock Constraints in Edge-Triggered Systems

Latch timing

When data arrives to transparent latch
Latch is a ‘soft’ barrier

When data arrives to closed latch
Data has to be ‘re-launched’
**Single-Phase Clock with Latches**

![Diagram of Single-Phase Clock with Latches]

Unger and Tan
Trans. on Comp.
10/86

In Rabaey Chapter 10:

\[ T_{sk} = T_{skl} + T_{skt} \]

**Preventing Late Arrivals**

![Diagram of Preventing Late Arrivals]

Data must arrive

\[ T_{cy} \]

\[ T_{cw} \]

\[ T_{su} \]

\[ T_{dm} \]

\[ T_{su} \]
Preventing Late Arrivals

\[ t_{CY} \geq \max \left\{ \frac{T_{skl} + T_{skt} + T_{SU} + T_{CQM} - PW}{T_{DQM}} \right\} + T_{LM} \]

Or:

\[ t_{CY} \geq T_{CQM} + T_{LM} + T_{SU} + T_{skl} + T_{skt} - PW \]
\[ t_{CY} \geq T_{DQM} + T_{LM} \]

Preventing Premature Arrivals

\[ T_{Lm} \geq T_{skl} + T_{skt} + T_{H} + PW - T_{CQm} \]
Single-Latch Timing Summary

Bounds on logic delay:

\[
t_{CY} \geq \max \left\{ \frac{T_{skl} + T_{skl} + T_{SU} + T_{CQM} - PW}{T_{DQM}} \right\} + T_{LM}
\]

\[
T_{LM} \geq T_{skl} + T_{skl} + T_{H} + PW - T_{CQm}
\]

Solutions:
1) Balance logic delays
2) Locally generated short PW

Latch-Based Design

L1 latch is transparent when Clk = 0

L2 latch is transparent when Clk = 1
Latch-Based Timing

As long as transitions are within the assertion period of the latch, no impact of position of clock edges.

Latch Design and Hold Times

$t_{CLL} \geq t_{SK} + (t_{H} - t_{Q})$
Latch-Based Timing

Longest path

\[ T_{CY} \geq 2T_{DQM} + T_{LHM} + T_{LLM} \]

\( \text{Independent of skew} \)

Short paths

\[ T_{CLLm} \geq T_{SK} + T_H - T_{CQm} \]

\[ T_{CLHm} \geq T_{SK} + T_H - T_{CQm} \]

\( \text{Same as register-based design but holds for both clock edges} \)

Can tolerate skew!
Soft-Edge Properties of Latches

- **Slack passing** – logical partition uses left over time (slack) from the *previous* partition.
- **Time borrowing** – logical partition utilizes a portion of time allotted to the *next* partition.
- Makes most impact in unbalanced pipelines.

Bernstein et al, Chapter 8, Chandrakasan, Chap 11 (by Partovi)

---

Slack-Passing and Cycle Borrowing

For *N* stage pipeline, overall logic delay should be < *N* Tcl
4. Design for performance

B. Statistical timing

Pictorial view of setup and hold tests

- Actual early AT
- Actual late AT
- Early RAT
- Late RAT
- Data must be stable
- Early slack
- Late slack
- Hold time
- Latest clock arrival time
- Setup time
- Earliest clock arrival time (next cycle)

ICCAD ’07 Tutorial
Chandu Visweswariah
Handling of across-chip variation

- Each gate has a range of delay: [lb, ub]
  - The lower bound is used for early timing
  - The upper bound is used for late timing
- This is called an early/late split
- Static timing obtains bounds on timing slacks
  - Timing is performed as one forward pass and one backward pass

Setup test
- Launching late path
- Capturing early path

Hold test
- Launching early path
- Capturing late path

How is the early/late split computed?

- The best way is to take known effects into account during characterization of library cells
  - History effect, simultaneous switching, pre-charging of internal nodes, etc.
  - This drives separate characterization for early and late; this is the most accurate method
- Failing that, the most common method is derating factors
  - Example:
    - Late delay = library delay * 1.05
    - Early delay = library delay * 0.95
- The IBM way of achieving derating is LCD factors (Linear Combination of Delay) (FC=fast chip, SC=slow chip, see next page)
  - Late delay = $\alpha_L \cdot FC_{\text{delay}} + \beta_L \cdot NOM_{\text{delay}} + \gamma_L \cdot SC_{\text{delay}}$
  - Early delay = $\alpha_E \cdot FC_{\text{delay}} + \beta_E \cdot NOM_{\text{delay}} + \gamma_E \cdot SC_{\text{delay}}$
- Across-chip variation is therefore assumed to be a fixed proportion of chip-to-chip variation for each cell type
IBM delay modeling*

At a given corner
late delay = intrinsic + systematic + random
early delay = intrinsic – systematic – random

*P. S. Zuchowski, ICCAD’04

Traditional timing corners

*ICCAD ’07 Tutorial Chandu Visweswarah
The problem with an early/late split

- The early/late split is very useful
  - Allows bounds during delay modeling
  - Any unknown or hard-to-model effect can be swept under the rug of an early/late split
- But, it has problems
  - Additional pessimism (which may be desirable)
  - Unnecessary pessimism (which is never desirable)

Additional pessimism: Clock tree view

This physically common portion can't be both fast and slow at the same time
How to have less pessimism?

- Common path pessimism removal
- Account for correlations
- Credit for statistical averaging of random

Statistical timing

- Deterministic

- Statistical

ICCAD '07 Tutorial
Chandu Visweswariah
The problem of correlations

- There are many reasons for correlations
  - Chip-to-chip variations are perfectly correlated within a single chip
  - Same circuit types
  - Same device families
  - Same metal levels
  - Same voltage islands
  - Same regions of the chip
  - Dependence on common sources of variation
  - Reconvergent fanout
  - Etc.
- In a reasonable-sized chip, there may be 100 million timing quantities, so we don't handle correlations in the classical way
  - Not by storing and manipulating a 100M x 100M covariance matrix

Canonical form

\[ a_0 + a_1 \Delta X_1 + a_2 \Delta X_2 + \cdots + a_n \Delta X_n + a_{n+1} \Delta R_a \]

- All timing quantities are parameterized by the sources of variation
- Correlation can be judged on-demand by inspection
Statistical timing basics

- Represent all timing quantities in canonical form
  - Delays, slews, guard times, ATs, RATs, slacks, PLL adjusts, constraints, CPPR adjusts
- Propagate ATs forward through the timing graph
  - Addition of two canonical forms is easy
  - Max/min operations are also easy with the help of some analytic formulas
- Propagate RATs backward through the timing graph
  - Subtraction of two canonical forms is easy
  - Use statistical max/min operations
- Slack is simply the difference between AT and RAT
  - Since this is available in canonical form, we get sensitivities of circuit performance to sources of variation for free
  - These can be used to ensure a robust design

Statistical max operation

\[
A = a_0 + \sum_{i=1}^{n} a_i \Delta X_i + a_{n+1} \Delta R_a \\
B = b_0 + \sum_{i=1}^{n} b_i \Delta X_i + b_{n+1} \Delta R_b \\
\sigma_A = \sqrt{\sum_{i=1}^{n+1} a_i^2} \\
\sigma_B = \sqrt{\sum_{i=1}^{n+1} b_i^2} \\
\rho = \sum_{i=1}^{n} a_i b_i \\
\theta = \left(\sigma_A^2 + \sigma_B^2 - 2\rho \sigma_A \sigma_B\right)^{1/2} \\
t = \Phi\left(\frac{a_0 - b_0}{\theta}\right) \\
E[\max(A, B)] = a_0 t + b_0 (1 - t) + \theta \Phi\left(\frac{a_0 - b_0}{\theta}\right) \\
E[\max(A, B)]^2 = (\sigma_A^2 + a_0^2) t + (\sigma_B^2 + b_0^2) (1 - t) + (a_0 + b_0) \theta \Phi\left(\frac{a_0 - b_0}{\theta}\right)
\]

*M. Cain, “The moment-generating function of the minimum of bivariate normal random variables,” American Statistician, May ’94, 48(2)
Unified view of correlations

\[ D = a_i + \sum a_i \Delta X_i + a_r \Delta X_r \]

**Spatial correlation vs. early/late split**

Dependence on common virtual variables cancels out at the timing test
Next Lecture

- Latches and flip-flops