inst.eecs.berkeley.edu/~ee241b

EE241B : Advanced Digital Circuits

Lecture 10 – Latch Timing

Borivoje Nikolić

February 24, 2020, imore: Report: iPhone 12 to have new short-range WiFi standard, AirTags to charge like Apple Watch

A new report claims that Apple's unannounced iPhone 12 will feature a new short-range WiFi technology, called 802.11ay.

This is a rumor – but it would be cool!

Announcements

- Response to project abstracts today, by e-mail
  - Team web pages
  - Be careful not to leak proprietary info (interface tools via Hammer)
- Quiz 1 today
- Reading
- Chapter 11 (Partovi) in Chandrakasan, Fox, Bowhill

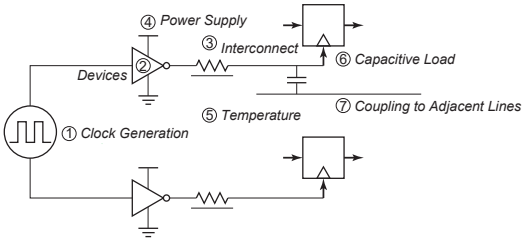
Outline

- Module 3
  - Flip-flop timing
  - Latch-based timing

3. Design for Performance

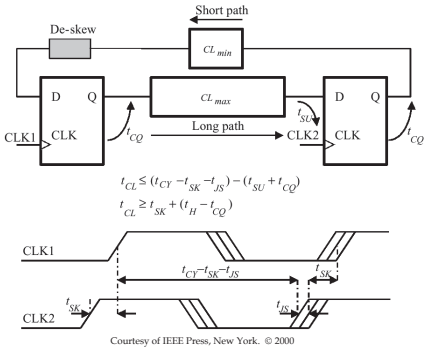
3.A Flip-Flop Timing

Clock Uncertainties



Sources of clock uncertainty

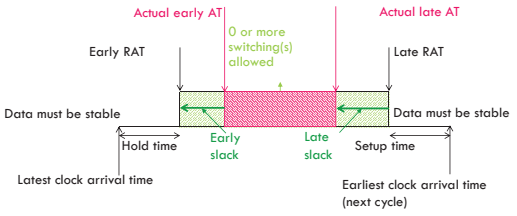
Clock Constraints in Edge-Triggered Systems



Courtesy of IEEE Press, New York. © 2000

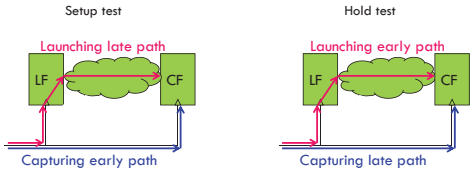
3.B Timing with Uncertainty/Variations

Pictorial View of Setup and Hold Tests



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

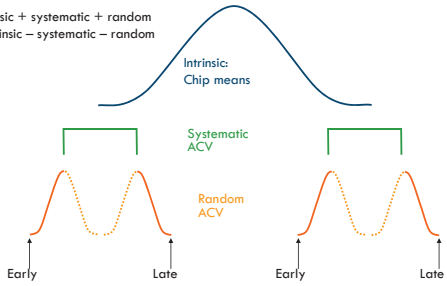


How is the Early/Late Split Computed?

- 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 = α_L * FC_delay + β_L * NOM_delay + γ_L * SC_delay
    Early delay = α_E * FC_delay + β_E * NOM_delay + γ_E * SC_delay
  - Across-chip variation is therefore assumed to be a fixed proportion of chip-to-chip variation for each cell type

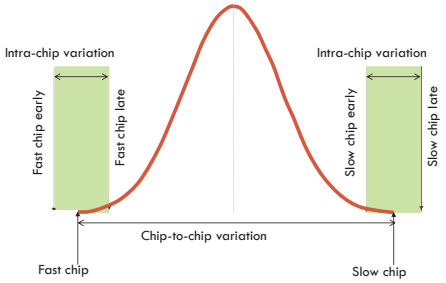
IBM Delay Modeling*

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



*P. S. Zuchowski, ICCAD'04

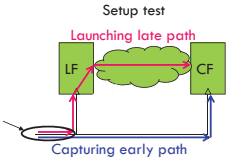
Traditional Timing Corners



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)

This physically common portion can't be both fast and slow at the same time

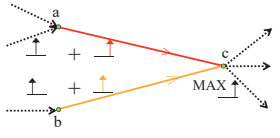


How to Have Less Pessimism?

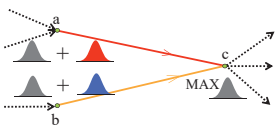
- Common path pessimism removal
- Account for correlations
- Credit for statistical averaging of random

Statistical Timing

- Deterministic



- Statistical

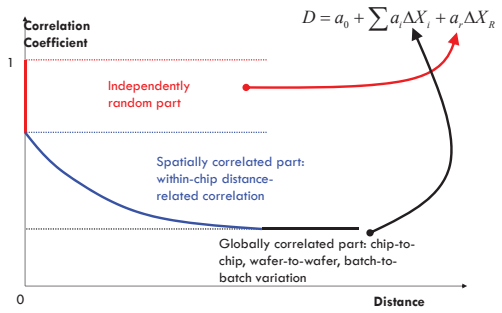


Statistical Max Operation

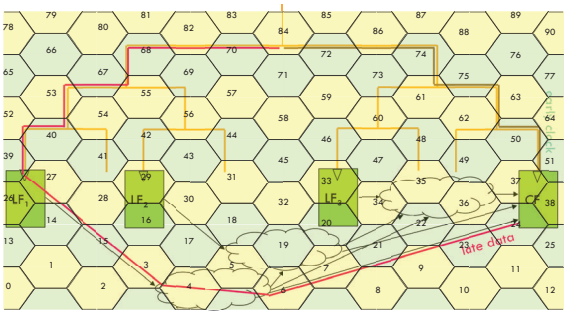
$$
\begin{aligned}
A &= a_0 + \sum_{i=1}^{n} a_i \Delta X_i + a_{n+1} \Delta R_a \\
B &= b_0 + \sum_{i=1}^{n} b_i \Delta X_i + b_{n+1} \Delta R_b \\
\sigma_A &= \sqrt{\sum_{i=1}^{n+1} a_i^2} \\
\sigma_B &= \sqrt{\sum_{i=1}^{n+1} b_i^2} \\
\rho &= \frac{\sum_{i=1}^{n} a_i b_i}{\sigma_A \sigma_B} \\
\theta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

*C. E. Clark, "The greatest of a finite set of random variables," OR Journal, March-April 1961, pp. 145—162
**M. Cain, "The moment-generating function of the minimum of bivariate normal random variables," American Statistician, May '94, 48(2)

Unified View of Correlations



Spatial Correlation vs. Early/Late Split



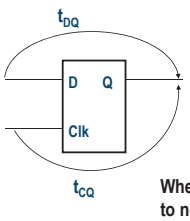
Dependence on common virtual variables cancels out at the timing test

3.C Latch Timing

Key Point

- Latch-based sequencing can improve performance, but is more complicated
  - Timing analysis not limited to a consecutive pair of latches

Latch Timing



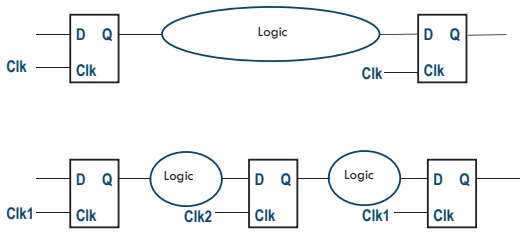
When data arrives to transparent latch

Latch is a 'soft' barrier

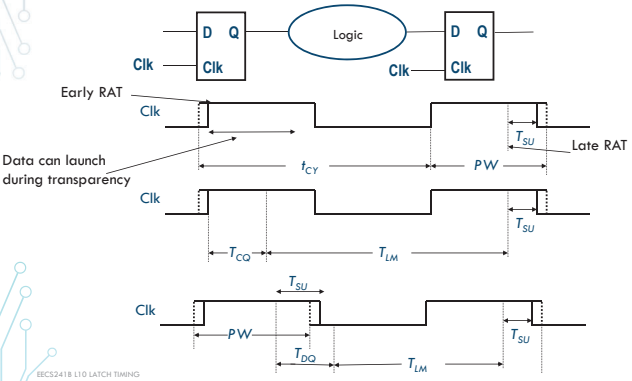
When data arrives to non-transparent latch

Data has to be 're-launched'

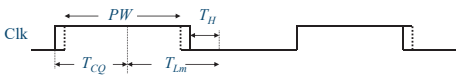
Latch Sequencing



Preventing Late Arrivals

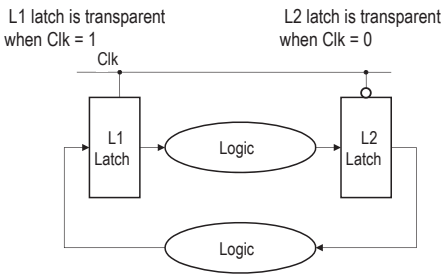


Preventing Premature Arrivals



- Data should not be able to race through during transparency

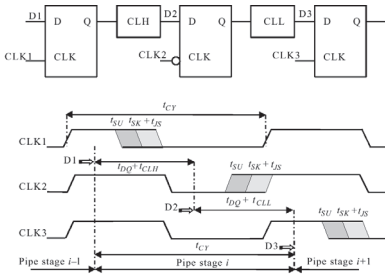
Two-Phase Latch-Based Design



- Two-phase non-overlapping is safer, but adds margin

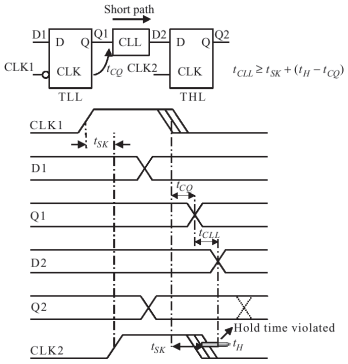
Latch-Based Timing

- Single-phase, two-latch



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

Latch Design and Hold Times

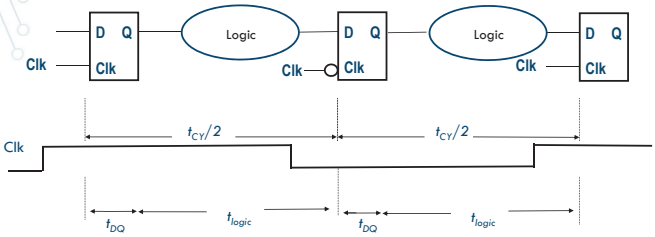


Soft-Edge Properties of Latches

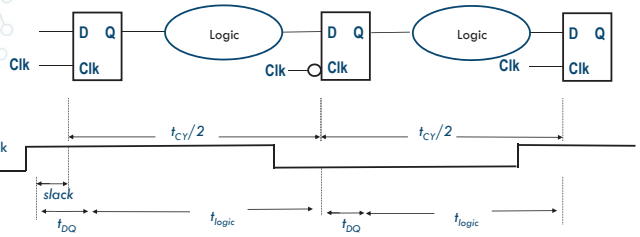
- Slack passing – logical partition uses left over time (slack) from the *previous* partition
- Time borrowing – logical partition utilizes a portion of time allotted to the *next* partition
- Makes most impact in unbalanced pipelines

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

Slack Passing and Time Borrowing

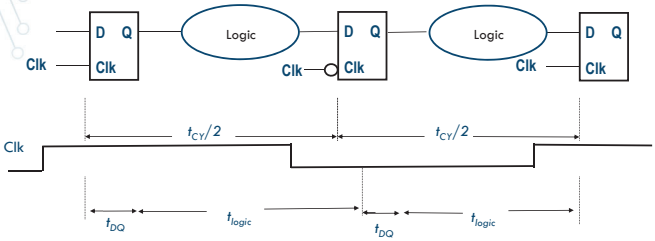


Slack Passing and Time Borrowing



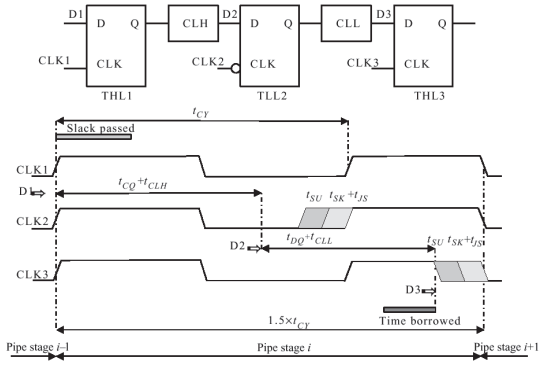
- Slack passed

Slack Passing and Time Borrowing



- Time borrowed

Slack-Passing and Cycle Borrowing



For N stage pipeline, overall logic delay should be < N Tcl

Next Lecture

- Flip-flops