EE241B : Advanced Digital Circuits

Lecture 11 – Timing

Borivoje Nikolić

So Long, Fry’s. I Learned Everything About Gadgets From You

Now that the big-box electronics store has shuttered, future generations need a place where they can touch and discover the next great technology.


Announcements

• Work on the labs 3 & 4!
• New assignment this week
• Response to project proposals this week
  • Project web pages
Outline

• Static timing
  • With variations
• Latch-based timing

3. Design for Performance

3.A Flip-Flop Timing
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

1. Clock Generation
2. Skew
3. Power Supply
4. Interconnect
5. Temperature
6. Capacitive Load
7. Coupling to Adjacent Lines
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
Clock Constraints in Edge-Triggered Systems

3.B Timing with Uncertainty/Variations
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 = 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*

ACV - across chip variation

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

*P. S. Zuchowski, ICCAD'04
Traditional Timing Corners

- Fast chip early
- Fast chip late
- Intra-chip variation
- Slow chip early
- Slow chip late
- Intra-chip variation
- Chip-to-chip variation

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**

\[ A = a_0 + \sum_{i=1}^{n} a_i \Delta X_i + a_{n+1} \Delta R_a \]

\[ B = b_0 + \sum_{i=1}^{n} b_i \Delta X_i + b_{n+1} \Delta R_b \]

\[ \sigma_A = \sqrt{\sum_{i=1}^{n+1} a_i^2} \]

\[ \sigma_B = \sqrt{\sum_{i=1}^{n+1} b_i^2} \]

\[ \rho = \frac{\sum_{i=1}^{n+1} ab_i}{\sigma_A \sigma_B} \]

\[ \theta \equiv \left( \sigma_A^2 + \sigma_B^2 - 2 \rho \sigma_A \sigma_B \right)^{1/2} \]

\[ t = \phi \left( \frac{a_0 - b_0}{\theta} \right) \]

\[ E[\max(A, B)] = a_0 + b_0 (1 - t) + \theta \phi \left( \frac{a_0 - b_0}{\theta} \right) \]

\[ E[\max(A, B)]^2 = (a_0^2 + \theta^2) t + (\sigma_A^2 + \sigma_B^2) (1 - t) + (a_0 + b_0) \theta \phi \left( \frac{a_0 - b_0}{\theta} \right) \]

---

**Unified View of Correlations**

- **Independently random part**
  \[ D = a_0 + \sum a_i \Delta X_i + a_i \Delta X_R \]

- **Spatially correlated part**
  within-chip distance-related correlation

- **Globally correlated part**
  chip-to-chip, wafer-to-wafer, batch-to-batch variation
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**

![Latch Timing Diagram]

- 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

Single latch

D Q
Clk

Logic

D Q
Clk

Two-latch segment

D Q
Clk

Logic

D Q
Clk

D Q
Clk

Preventing Late Arrivals

Early RAT

Data can launch during transparency

Late RAT

Clk

Clk

Clk

Clk

Clk

Clk

Logic

Logic

Logic

Clk

Clk

Clk

Clk

Clk

Clk

Clk

Clk
Preventing Premature Arrivals

1) Make $t_{lmin} > PW$
   $t_{lmin} + t_{ca} + t_{tr} > PW$

2) Make PW small

- Data should not be able to race through during transparency

Two-Phase Latch-Based Design

- L1 latch is transparent when Clk = 1
- L2 latch is transparent when Clk = 0

- 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