**Timing**

Dr. Paul D. Franzon

Outline

1. Clock distribution
2. Standard single-phase, single-clock design
3. Two phase design styles

References

- Doane & Franzon, “Multichip Module Technologies & Alternatives”, CH. 11
- Hodges & Jackson, Analysis & Design of Digital Integrated Circuits
- + many other texts

---

**Clock Distribution**

Goal: Distribute master clock with identical clock edge arrival times at flip-flops or latches

Problem:
- Clock (capacitive) load is large
- Clock load is not evenly distributed on the chip

“Standard” Solutions
- Clock Trees (most ICs)
- Clock Grid (some microprocessors)
- Sepertines (some microprocessors and some small high speed circuits)

Measures of clock distribution quality
- Worst-case skew
  - systematic variation in space
- Jitter
  - random variation in time (cycle to cycle edge movement)
- Clock edge rate
**H-tree Clock Layout**

Standard Technique for Low-Skew clock distribution
- Objective: Identical delay to all flip-flops from source
- Methods:
  - Balance RC delays along all paths
  - Can be difficult as Cload not uniformly distributed
  - Rebuffer to increase edge rate at local level

![H-tree Clock Layout Diagram]

**Other Layouts**

**Clock Grid:**
- Low (but none zero) skew
- Easy to design
- Higher power
- Can be used to replace first layers of tree

![Clock Grid Diagram]

**Serpentine**
- Again trades ease of design for higher power
- Point to point wires
  - Identical delay

![Serpentine Diagram]
**Clock Skew**

Skew due to differential delays:

- Skew = systematic variation
  - Varied buffer delays
  - Wiring delay differences
- Jitter = cycle to cycle variation at any one point
  - Phase lock loop noise
    - caused by noise in Vdd/Gnd
  - Varying trigger points at buffer
  - Environmental effects
  - Seen as “fuzz” in eye diagram

---

**Single Phase Clock with Master-Slave Flip-Flops**

Best used with Static Logic, as no precharge period available.

Minimum Clock Period:

- $t_{\text{clock}}$
- To prevent ‘setup violations’

$$ t_{\text{clock}} = t_{\text{flip-flop-max}} + t_{\text{logic-max}} + t_{\text{interconnect-max}} + t_{\text{setup-max}} + t_{\text{skew-max}} + t_{\text{jitter-max}} $$
…Timing Equations

Requirement to Prevent Race:
- Prevents ‘hold violations’

\[ t_{\text{hold\text{-}max}} < t_{\text{flip\text{-}flop\text{-}min}} + t_{\text{logic\text{-}min}} + t_{\text{skew\text{-}max}} + t_{\text{interconnect\text{-}min}} \]

Logic + Interconnect

Clock-1 \rightarrow t_{\text{flip\text{-}flop\text{-}min}} \rightarrow t_{\text{logic\text{-}min}} + t_{\text{interconnect\text{-}min}}

Clock-2

\[ T_{\text{skew\text{+}jitter\text{-}max}} \rightarrow t_{\text{hold\text{-}max}} \]

Single Phase Latch Design

Structure:
- Requires two types of latches (n and p)
- Not seen much in practice
Two Phase Clocking

Locally generate clock’ from clock.

Clock

Clock’

Nominal timing relationships

“Slack Borrowing”: Up to half a clock cycle of extra logic delay can be accumulated in the pipeline.

Two Phase with Dynamic Logic

Nominal timing relationships

“Slack Borrowing”: Can adjust timing by trading logic delay before and after latch, e.g. ts1b can be reduced, so as to increase ts2a and put more logic in S2a and S2b combined.
Time Stealing

Locally move clock edges.

Clock

Clock'

Delayed clock

Variations on these Schemes

Variations:

- Underlapped two-phase clocks to account for local clock skew
- Time stealing with static logic and Master-slave flip-flops
- Logic with feedback
  - Requires that feedback structure looks like the pipeline
  - e.g., a sequence of a pair of latches
  - Take care in timing rules