ISSUES IN TIMING
The Clock Skew Problem

Clock Rates as High as 500 Mhz in CMOS!

Clock Edge Timing Depends upon Position
Delay of Clock Wire

\[ r = 0.07 \ \Omega/\mu \text{m}, \ c = 0.04 \ \text{fF}/\mu \text{m}^2 \]

(Tungsten wire)
Constraints on Skew

(a) Race between clock and data.

(b) Data should be stable before clock pulse is applied.
Clock Constraints in Edge-Triggered Logic

\[ \delta \leq t_{r, \text{min}} + t_i + t_{l, \text{min}} \]

\[ T \geq t_{r, \text{max}} + t_i + t_{l, \text{max}} - \delta \]

Maximum Clock Skew Determined by Minimum Delay between Latches
Minimum Clock Period Determined by Maximum Delay between Latches
Positive and Negative Skew

(a) Positive skew

(b) Negative skew
Clock Skew in Master-Slave Two Phase Design

\[ \phi_2 \]

\[ \phi_1 \]

\[ In \rightarrow CL1 \rightarrow M1 \rightarrow S1 \rightarrow CL2 \rightarrow M2 \rightarrow S2 \rightarrow CL3 \rightarrow M3 \rightarrow S3 \]

\[ \phi_1' \]

\[ \phi_2' \]
Clock Skew in 2-phase design

\[ t_{\text{min}} > \delta - T_{\phi_{12}} \]
\[ t_{\text{max}} > T + \delta - T_{\phi_{12}} \]
How to counter Clock Skew?

Data and Clock Routing
Clock Distribution

H-Tree Network

Observe: Only Relative Skew is Important
Clock Network with Distributed Buffering

Reduces absolute delay, and makes Power-Down easier
Sensitive to variations in Buffer Delay
Example: DEC Alpha 21164

Clock Frequency: 300 MHz - 9.3 Million Transistors

Total Clock Load: 3.75 nF

Power in Clock Distribution network : 20 W (out of 50)

Uses Two Level Clock Distribution:

- Single 6-stage driver at center of chip
- Secondary buffers drive left and right side clock grid in Metal3 and Metal4

Total driver size: 58 cm!
Clock Drivers
Clock Skew in Alpha Processor
Self-timed and asynchronous design

Functions of clock in synchronous design
1) Acts as completion signal
2) Ensures the correct ordering of events

Truly asynchronous design
1) Completion is ensured by careful timing analysis
2) Ordering of events is implicit in logic

Self-timed design
1) Completion ensured completion signal
2) Ordering imposed by handshaking protocol
Self-timed pipelined datapath

![Diagram of a self-timed pipelined datapath with three stages: R1, F1, R2, F2, R3, F3. Each stage has inputs and outputs labeled with time delays: $t_{pF1}$, $t_{pF2}$, $t_{pF3}$. The diagram shows the flow of signals through the stages with Req, Ack, Start, and Done signals.](image)
Completion Signal Generation

Using Delay Element (e.g. in memories)
Completion Signal Generation

Using Redundant Signal Encoding

<table>
<thead>
<tr>
<th>B</th>
<th>B0</th>
<th>B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>in transition (or reset)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>illegal</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Completion Signal in DCVSL
Self-timed Adder

(a) Differential carry generation

(b) Completion signal
Hand-shaking Protocol

(a) Sender-receiver configuration

(b) Timing diagram

Two-Phase Handshake
Event Logic —
The Muller C-element

(a) Schematic

(b) Truth table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>$F_{n+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$F_n$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$F_n$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
2-phase Handshake Protocol
Example: Self-timed FIFO
4-phase Handshake Protocol (or RTZ)
4-phase Handshake Protocol - Implementation
Asynchronous-Synchronous Interface

Asynchronous System

Synchronous System

f_{in}

Synchronization

f_{\phi}
A Simple Synchronizer

- Data sampled on Falling Edge of Clock
- Latch will eventually Resolve Signal Value, but ... this might take infinite time!
Synchronizer: Output Trajectories

Single Pole Model for Flip-Flop

\[ v(t) = V_{MS}^+ (v(0) - V_{MS})e^{t/\tau} \]
Simulated Trajectory versus One Pole Model
Mean Time to Failure

\[ N_{sync}(0) = \frac{P_{init}}{T_\phi} = \left( \frac{V_{IH} - V_{IL}}{V_{swing}} \right) \frac{t_r}{T_{signal}} \frac{1}{T_\phi} \]

\[ N_{sync}(T) = \frac{P_{init} e^{-T/\tau}}{T_\phi} = \left( \frac{V_{IH} - V_{IL}}{V_{swing}} \right) e^{-T/\tau} \frac{t_r}{T_{signal} T_\phi} \]
Example

\[ T_f = 10 \text{ nsec} = T \]
\[ T_{\text{signal}} = 50 \text{ nsec} \]
\[ t_r = 1 \text{ nsec} \]
\[ t = 310 \text{ psec} \]
\[ V_{\text{IH}} - V_{\text{IL}} = 1 \text{ V} \quad (V_{\text{DD}} = 5 \text{ V}) \]

\[ N(T) = 3.9 \times 10^{-9} \text{ errors/sec} \]
\[ MTF (T) = 2.6 \times 10^8 \text{ sec} = 8.3 \text{ years} \]
\[ MTF (0) = 2.5 \mu\text{sec} \]
Cascaded Synchronizers Reduce MTF
Arbiters

(a) Schematic symbol

(b) Implementation

(c) Timing diagram

Req1
Req2
A
B
Ack1
Ack2

A

B

V_T gap

metastable

t
Synchronization at System Level

Crystal-based clock-generator

Reference clock $\phi$

PC board

Chip 1

Clock Generator

Logic

I/O Data

Chip 2

Logic

Clock Generator

$\phi_1$

$\phi_1^{''}$

$\phi_2$

$\phi_2^{''}$

Digital Integrated Circuits

Timing

© Prentice Hall 1995
Skew of Local Clocks vs Reference

(a) Skew of local clock signals with respect of reference clock.

(b) Local clock signals as produced by PLL based clock generator.
Phase-Locked Loop Based Clock Generator

Acts also as Clock Multiplier
Ring Oscillator

(a) VCO

(b) Current starved inverter
Example of PLL-generated clock

(a) Clock generator output at 18 MHz.

(b) Clock generator output at 15 kHz.