ECE 124a/256c
Timing Protocols and Synchronization

Forrest Brewer
Timing Protocols

- Fundamental mechanism for coherent activity
  - Synchronous $\Delta \phi = 0$ $\Delta f = 0$
    - Gated (Aperiodic)
  - Mesochronous $\Delta \phi = \phi_c$ $\Delta f = 0$
    - Clock Domains
  - Plesiochronous $\Delta \phi =$changing $\Delta f =$slowly changing
    - Network Model (distributed synchronization)
- Asynchronous
  - Needs Synchronizer locally, potentially highest performance

- Clocks
  - Economy of scale, conceptually simple
  - Cost grows with frequency, area and terminals
Compare Timing Schemes I

- Signal between sub-systems
  - Global Synchronous Clock
  - Matched Clock Line Lengths

**Table 9-1: Timing Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Nominal</th>
<th>Skew (ps)</th>
<th>Jitter (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit cell</td>
<td>$t_{bit}$</td>
<td>2.5 ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmitter rise time</td>
<td>$t_r$</td>
<td>1.0 ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cable delay</td>
<td>$t_{wire}$</td>
<td>6.25 ns</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Receiver aperture</td>
<td>$t_a$</td>
<td>300 ps</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Transmitter delay</td>
<td>500 ps</td>
<td></td>
<td>150</td>
<td>50</td>
</tr>
<tr>
<td>Buffer stage delay</td>
<td>250 ps</td>
<td></td>
<td>100</td>
<td>50</td>
</tr>
</tbody>
</table>
- Send Both Clock and Signal – separately
  - Clock lines need not be matched
  - Buffer and line skew and jitter same as synch. Model
- Double Edge Triggered Clocks
Compare Timing Schemes III

- Gross Timing Margin: identical
  - Open Loop approach fails: time uncertainty 2.15nS (jitter+skew)
  - Closed Loop has net timing margin of 150pS (600pS - 450pS)
- Skew removed by reference clock matching
  - In general, can remove low bandwidth timing variations (skew), but not jitter

**TABLE 9-2  Skew and Jitter for Approaches A and B**

<table>
<thead>
<tr>
<th>Description</th>
<th>A Skew (ps)</th>
<th>A Jitter (ps)</th>
<th>B Skew (ps)</th>
<th>B Jitter (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter clock</td>
<td>600</td>
<td>250</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Receiver clock</td>
<td>600</td>
<td>250</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Transmitter</td>
<td>150</td>
<td>50</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>Receiver</td>
<td>100</td>
<td>50</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>Data cable</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference clock cable</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>1550</td>
<td>600</td>
<td>250</td>
<td>200</td>
</tr>
</tbody>
</table>
Open loop scheme requires particular clock frequencies
- Need for clock period to match sampling delay of wires
- Need Odd number of half-bits on wire e.g:

\[
\frac{t_{wire} + 0.5(t_r + t_a) + (t_{jitter} + t_{skew})}{N} \leq t_{bit} \leq \frac{t_{wire} - 0.5(t_r + t_a) - (t_{jitter} + t_{skew})}{N - 1}
\]

- For open loop scheme – this give 9nS/bit

- For redesign with jitter+skew = 550pS
  - Can operate with 2.5nS, 4.4nS, or 7.5nS+
  - But not 2.6nS!
- Moral-- avoid global timing in large distributed systems
Timing Nomenclature

- Rise and Fall measured at 10% and 90% (20% and 80% in CMOS)
- Pulse width and delays measured at 50%
- Duty Cycle \( d_B = \frac{t_{WBh}}{t_{CYB2}} \)
- Phase \( \phi_{AB} = 2\pi \frac{t_{AB}}{t_{CYA2}} \)
- RMS (Root Mean Square) \( V_{RMS} = \sqrt{\frac{1}{cycle} \int V(t)^2 dt} \)
Delay, Jitter and Skew

- Practical systems are subject to noise and process variations
  - Two signal paths will not have the same delay
  - Skew = average difference over many cycles
    - Issue is bandwidth of timing adjustment = PLL bandwidth
    - Can often accommodate temperature induced delay
  - Jitter = real-time deviation of signal from average
    - High frequency for which timing cannot be dynamically adjusted
    - Asynchronous timing can mitigate jitter up to circuit limit
Combinational Logic Timing

- Static Logic continuously re-evaluates its inputs
  - Outputs subject to "Glitches" or static hazards
  - A changing input will contaminate the output for some time \( t_{cAX} \)
  - But will eventually become correct \( t_{dhAX} \)
- \( t_{dhAX} \) is the sum of delays on the longest timing path from A to X
- \( t_{cAX} \) is the sum of delays on shortest timing path from A to X
Inertial Delay Model: Composition by Adding
- Both signal propagation and contamination times simply add
- Often separate timing margins are held for rising and falling edges
- Delays compose on bits – **not** busses!
  - Bit-wise composite delays are a gross approximation without careful design
Edge Triggered Flip-flop

- $t_a$ is the timing aperture width, $t_{ao}$ is the aperture offset
  
  $$t_{setup} = t_a/2 - t_{ao} + t_r/2$$
  
  $$t_{hold} = t_a/2 + t_{ao} + t_r/2$$

- $t_{cCQ}$ is the contamination delay

- $t_{dCQ}$ is the valid data output delay

- Note: in general, apertures and delays are different for rising and falling edges
Level Sensitive Latch

- Latch is transparent when clk is high
  - \( t_{dDQ}, t_{cDQ} \) are transparent propagation times, referenced to D
  - \( t_s, t_h \) referenced to falling edge of clock
  - \( t_{dCQ}, t_{cCQ} \) referenced to rising edge of clock
Double-(Dual)-Edge Triggered Flipflop

- D is sampled on both rising and falling edges of clock
  - Inherits aperture from internal level latches
  - Does not have data referenced output timing— is not transparent
- Doubles data rate per clock edge
  - Duty cycle of clock now important
■ Rectangle in eye is margin window
  ■ Indicates trade-off between voltage and timing margins
  ■ To have an opening: $t_{cy} \geq 2t_u + t_a + t_r$
  (\(t_u\) is a maximum value – the worst case early to late is 2\(t_u\))
Signal Encoding

- Aperiodic transmission must encode **that** a bit is transferred and what bit
  - Can encode events in time
  - Can encode using multiple bits
  - Can encode using multiple levels
More Signal Encoding

- Cheap to bundle several signals with a single clock
  - DDR and DDR/2 memory bus
  - RAMBUS
- If transitions must be minimized, (power?) but timing is accurate – phase encoding is very dense
Synchronous Timing (Open Loop)

- Huffman FSM
- Minimum Delay
- Maximum Delay

\[ t_{cAY} \geq t_k + t_h - t_{cCQ} \]

\[ t_{cy} \geq t_{dBY} + t_k + t_s + t_{dCQ} \]
Two-Phase Clocking (latch)

- Non-overlapping clocks $\phi_1$, $\phi_2$
  - Hides skew/jitter to width of non-overlap period
- 4 Partitions of signals
  - $A^2$ (valid in $\phi_2$)
  - $C^1$ (valid in $\phi_1$)
  - $B^{f2}$ (falling edge of $\phi_2$)
  - $D^{f1}$ (falling edge of $\phi_1$)
More 2-phase clocking (Borrowing)

- Each block can send data to next early (during transparent phase)
  - Succeeding blocks may start early (borrow time) from fast finishers
- Limiting constraints:
  \[ t_{dAB} \leq t_{cy} - t_{no21} - t_s - t_{sCQ} - t_k \]
  \[ t_{dCD} \leq t_{cy} - t_{no12} - t_s - t_{sCQ} - t_k \]
- Across cycles can borrow:
  \[ t_{dN} \leq N(t_{yc} - 2t_{dDQ}) \]
Still More 2-phase clocking

- **Skew/Jitter limits**
  - Skew+jitter hiding limited by non-overlap period, else:
    \[ t_{cCD} \geq t_k + t_h - t_{no12} - t_{cCQ} \quad t_{cAB} \geq t_k + t_h - t_{no21} - t_{cCQ} \]
  - Similarly, the max cycle time is effected if skew+jitter > clk-high:
    \[ t_{cy} \geq t_{dAB} + t_{dCD} + 2t_{dDQ} + 2\max(0, t_k + t_s + t_{dCQ} - t_w - t_{dDQ}) \]
Qualified Clocks (gating) in 2-phase

- Skew hiding can ease clock gating
  - Register above is conditionally loaded ($B^1$ true)
  - Alternative is multiplexer circuit which is slower, and more power
- Can use low skew “AND” gate:
**Pseudo-2Phase Clocking**

- **Zero-Overlap analog of 2 phase:**
  - Duty cycle constraint on clock

\[
\begin{align*}
t_{cAB} & \geq t_k + t_h - t_{cCQ} \\
t_{cCD} & \geq t_k + t_h - t_{cCQ} \\
t_{cy} & \leq t_{dAB} + t_{dCD} + t_k + 2t_{dDQ}
\end{align*}
\]
Pipeline Timing

- Delay successive clocks as required by pipeline stage
  - Performance limited only by uncertainty of clocking (and power!)
  - Difficult to integrate feedback (needs synchronizer)
  - Pipeline in figure is wave-pipelined: \( t_{\text{cyc}} < t_{\text{prop}} \) (must be hazard free)

\[
\begin{align*}
t_{nAB} &= \frac{(t_{cAB} + t_{dAB})}{2} \\
t_{uAB} &= \frac{(t_{dAB} - t_{cAB})}{2} \\
t_{cAB} &= t_{nAB} - t_{uAB} \\
t_{dAB} &= t_{nAB} + t_{uAB} \\
t_{\text{validB}} &= t_{\text{cyc}} - t_{uAB} - t_{rB}
\end{align*}
\]
More Pipeline Timing

- Valid period of each stage must be larger than ff aperture
  - By setting delay, one can reduce the cycle time to a minimum:
    \[ t_{cyc} > t_{uAB} + t_{u\phi} + t_{aperture} + t_{rB} \]

- Note that the cycle time and thus the performance is limited only by the uncertainty of timing – not the delay
  - Fast systems have less uncertain time delays
  - Less uncertainty usually requires more electrons to define the events => more power
Latch based Pipelines

\[ t_{uA} = \frac{(t_{\text{width}} - t_{cCQ} - t_s + t_dQQ)}{2} \approx \frac{t_{\text{width}}}{2} \]

\[ t_{uB} = t_{uAB} + t_{uA} \]

\[ t_{\text{cyc}} > t_{uB} + t_{uPH} + t_{\text{aperture}} + t_{rB} \]

- Latches can be implemented very cheaply
  - Consume less power
  - Less effective at reducing uncertain arrival time
Feedback in Pipeline Timing

- Clock phase relation between stages is uncertain
  - Need Synchronizer to center feedback data in clock timing aperture
  - Worst case: performance falls to level of conventional feedback timing (Loose advantage of pipelined timing)
- Delays around loop dependencies matter
  - Speculation?
Delay Locked Loop

\[ \Delta \phi = 2\pi \left( t_{\phi I} - t_{\phi O} \right) / t_{cyc} = \pi - 2\pi \left( t_d + t_b \right) / t_{cyc} \]

- Loop feedback adjusts \( t_d \) so that \( t_d + t_b \) sums to \( t_{cyc} / 2 \)
  - Effectively a zero delay clock buffer
  - Errors and Uncertainty?
Loop Error and Dynamics

- The behavior of a phase or delay locked loop is dominated by the phase detector and the loop filter
  - Phase detector has a limited linear response
  - Loop filter is low-pass, high DC (H(0) gain)
- Loop Response: \( \Delta \phi(s)/e(s) = 1/(1 + H(s)) \)
- When locked, the loop has a residual error:

\[
\Delta \phi_r = \frac{2\pi}{t_{cyc}} \left( \frac{t_{cyc}/2 - t_{d0} - t_b}{1 + k_l} \right)
\]

\[
k_l = \frac{2\pi}{t_{cyc}} H(0)
\]

Where \( k_l \) is the DC loop gain
More Loop Dynamics

- For simple low pass filter: \[ H(s) = \frac{ka}{s + a} \]

- Loop Response: \[ \frac{\Delta\phi(s)}{e(s)} = \left( \frac{s + a}{s + a(k + 1)} \right) \]

- Time response: \( \phi(t) = \phi(0) \exp(-a(k + 1)t) \)
  - So impulse response is to decay rapidly to locked state
  - As long as loop bandwidth is much lower than phase comparator or delay line response, loop is stable.
On-Chip Clock Distribution

- **Goal:** Provide timing source with desired jitter while minimizing power and area overhead
  - **Tricky problem:**
    - (power) Wires have inherent loss
    - (skew and jitter) Buffers modulate power noise and are non-uniform
    - (area cost) Clock wiring increases routing congestion
    - (jitter) Coupling of wires in clock network to other wires
    - (performance loss) Sum of jitter sources must be covered by timing clearance
    - (power) Toggle rate highest for any synchronous signal

- **Low-jitter clocking over large area at high rates uses enormous power!**
  - Often limit chip performance at given power
On-Chip Clock Distribution

- Buffers
  - Required to limit rise time over the clock tree
  - Issues
    - jitter from Power Supply Noise \( t_j \propto V_{\text{noise}}t_r \)
    - skew and jitter from device variation (technology)

- Wires
  - Wire Capacitance (Buffer loading)
  - Wire Resistance
    - Distributed RC delay (rise-time degradation)
    - Tradeoff between Resistance and Capacitance
      - wire width; Inductance if resistance low enough
  - For long wires, desire equal lengths to clock source.
Clock Distribution

- For sufficiently small systems, a single clock can be distributed to all synchronous elements
  - Phase synchronous region: Clock Domain
  - Typical topology is a tree with the master at the root
  - Wirelength matching
On-Chip Clock Example

- **Example:**
  - $10^6$ Gates
  - 50,000 Flip-flops
  - Clock load at each flop 20fF
  - Total Capacitance 1nF
  - Chip Size 16x16mm
  - Wire Resistivity 70mW/sq.
  - Wire Capacitance 130aF/μm$^2$ (area) +80aF/μm (fringe)
  - 2V 0.18μm, 7Metal design technology
On-Chip Example

Delay = 2.8nS
Skew < 560pS

<table>
<thead>
<tr>
<th>Level</th>
<th>Fan-Out</th>
<th>Wire Length (mm)</th>
<th>Wire Width (μm)</th>
<th>$C_w$ (ff)</th>
<th>$R_w$ (Ω)</th>
<th>$C_L$ (ff)</th>
<th>$τ_{wire}$ (ps)</th>
<th>$C_O/C_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1,880</td>
<td>93</td>
<td>250</td>
<td>93</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1,410</td>
<td>70</td>
<td>250</td>
<td>57</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1,880</td>
<td>93</td>
<td>250</td>
<td>93</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1,410</td>
<td>70</td>
<td>250</td>
<td>57</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>0.6</td>
<td>158</td>
<td>117</td>
<td>250</td>
<td>37</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>0.6</td>
<td>158</td>
<td>117</td>
<td>667</td>
<td>85</td>
<td>20</td>
</tr>
</tbody>
</table>
Systematic Clock Distribution

- Automate design and optimization of clock network
  - Systematic topology:
    - Minimal Spanning Tree (Steiner Route):
      - Shortest possible length
    - H-tree:
      - Equal Length from Root to any leaf (Square Layout)
  - Clock Grid/Matrix:
    - Electrically redundant layout
- Systematic Buffering of loss
  - Buffer Insertion
    - Jitter analysis
    - Power Optimization
- Limits of Synchronous Domains
  - Power vs. Area vs. Jitter
Minimal Spanning Tree

- Consider N uniformly distributed loads
- Assume L is perimeter length of chip
- What is minimal length of wire to connect all loads?

- Average distance between loads:
  \[ d = \frac{L}{\sqrt{N}} \]

- Pairwise Connect neighbors:
  \[ \frac{(N/2)L}{\sqrt{N}} = \frac{L}{2} \sqrt{N} \]

- Recursively connect groups
  \[ W = \frac{L}{2} \sqrt{N} + \frac{L}{4} \sqrt{N} + ... = L \sqrt{N} \]
H-tree

- Wire strategy to ensure equal path lengths = D
- Total Length = \( \frac{3D\sqrt{N}}{2} \)
- Buffer as necessary (not necessarily at each branch)
Local Routing to Loads

- Locally, route to flip-flops with minimal routing
  - Conserve Skew for long wire links (H-tree or grid) but use MST locally to save wire.
    - Most of tree routing length (c.f. capacitance) in local connect!
  - Penfield/Horowitz model distributed delay along wires
    - Determine both skew and risetime
  - Local nets of minimal length save global clock power
    - Locality implies minimal skew from doing this
Buffer Jitter from Power Noise

To first order, the jitter in a CMOS buffer from supply variation is proportional to the voltage variation and the slope at 50% of the swing.

\[
\frac{dV}{dt} = \frac{V_{dd}}{RC} (e^{-t/RC}) \Rightarrow \frac{dt}{dV} \bigg|_{50\% \text{swing}} = \frac{2RC}{V_{dd}} \Rightarrow \Delta t = 2RC \left( \frac{\Delta V}{V_{dd}} \right)
\]
Example 1 (Power lower bound)

- 100,000 10fF flip flops, 1cm$^2$ die
  - minimum clock length = 3.16 meters
  - For interconnect 0.18 wire (2.23pf/cm) => 705pF capacitance
  - Total Loading w/o buffers is 1.705nF
  - 1.8 Volt swing uses 3.05nC of charge per cycle
  - 300MHz Clock => $3 \times 10^8 \times 3.05 \text{nC} = 0.915 \text{A}$
  - Without any buffering, the clock draws $1.8V \times 0.91 \text{A} = 1.6 \text{W}$
Example 2 (Delay and Rise Time)

- Wire resistance 145Ω/mm
  - Assuming H-tree: R=5mm*145Ω, C=1.7nF
  - Elmore Delay From Root (perfect driver) to leaf--
  - Delay =\((1/2)R*(1/2)C+(1/2)R*(1/4)C = (3/8)RC\) 
    \+(1/4)R*(1/8)C+(1/4)R*(1/16)C = (3/64)RC 
    \+(1/8)R*(1/32)C+(1/8)R*(1/64)C = (3/512)RC 
    \+ ... 
    = (3/8)RC(1+1/8+1/64+1/512+...) = (3/7)RC = 528nS!
  - Clearly no hope for central buffer unless much lower wire resistance...
  - At W=100um, R=1.32Ω(5mm), C=2.17nF => (3/7)RC=1.2nS but this presumes a perfect clock driver of nearly 4A. (Here we assumed top level metal for top 5 levels then interconnect for rest).
Distributed Buffer Clock Network

- In general, tradeoff buffer jitter (tree depth) with wire width (power cost)
- Use Grid or H-Tree at top of tree
- MST at bottom of tree
- Lower Bound on number of Buffers: (vs. rise time requirement)
  - Total Capacitance of network: $C_t$
  - Delay and load of Buffer: $D = aC + b$; $C_b$
  - Given $N$ buffers, assume equal partition of total load $= C_t + NC_b$
  - Delay $D$ is 50%, rise time is 80% -- multiplier is 1.4:

\[
t_r = 1.4D = 1.4\left(\frac{a(C_t + NC_b)}{N + b}\right) \quad N = \frac{aC_t}{t_r/1.4 - b - aC_b}
\]
Example 3 (Distributed Buffer)

- Reprise: 1.8V 0.18um 100,000 10fF leaves, 1cm$^2$, 316cm
- Wire Cap + load = 1.7nF
- MMI_BUFC: 44fF load, delay(pS) = 1240*C(pF)+28pS
- Need 34,960 buffers, 1.54nF Buffer Cap to meet 200pS rise time at leaves.
- Total Cap = 3.24nF, so at 300MHz Power= 3.15W
- On a single path from root to leaf, need 111 buffers (1cm) – note that this is far from optimal delay product.
  - Clump to minimize serial buffers i.e. 11 in parallel each mm.
  - 1mm load = 224fF wire + 480fF Buffer = 700fF
  - Delay = 145*112+100*700fF + 28pS = 114pS/mm = 1.1nS
  - Issue: 10 buffers along path => jitter!
Clock Grid

- Structure used to passively lower delivered jitter (relative to tree)
- 150pF load, 350pF Wire Cap, 8.5mm², 14um wire width
  - Ground plane to minimize inductance
Example

- H-tree example
- 150pF load, 8.5mm², Variable wire width
  - plot of response, each layer (note TM effects on root notes)
Folded (serpentine)

- Used in Pentium Processors
  - Fold wire to get correct length for equal delay
- Results: Grid: 228pF, 21pS delay, 21pS skew
  - Tree: 15.5pF 130pS delay, skew low
  - Serp: 480pF 130pS delay, lowest skew
TM Model Improvement

- TM effects added to design of variable width tree
- TM issues important when wire widths are large –
  - IR small relative to LdI/dt