Modeling and Budgeting of Timing Jitter and Noise I

Prerequisite Reading: Chapter 8
High-Speed Digital System Design,

Acknowledgements: Intel Bus Boot Camp: Howard Heck
Agenda

- Voltage and Time
- Budgets
- Computer Signaling Elements and Circuits
- Flight time
- Synchronous Bus Operation
- Clock Skew and Jitter
- Setup and Hold
- Manufacturing Considerations
- Advanced Topics
Voltage and time

- SI boils down to meeting voltage and time specifications
- True for most I/O computer interfaces
- Violating a time or voltage specification i.e. exceeding a limit, may cause a circuit to fail
  - This is not a guarantee
  - Most limits are at least 3 sigma limits.
    - The actual sigma limits are usually a company secret.
  - Margin is the difference between a specification and the respective measured signal parameter.
    Margin is considered a quality factor for a design.
SI Budgets

- A SI budget is a technique used to report timing and voltage margins in terms of constituent voltage and timing components for all configurations and conditions of a particular bus design.
- The budget is often represented in a spreadsheet.

<table>
<thead>
<tr>
<th>Margin</th>
<th>Voltage Spec</th>
<th>Noise Bucket</th>
<th>Measured Voltage</th>
<th>Measurement Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>100</td>
<td>10</td>
<td>56</td>
<td>20 (mv)</td>
</tr>
</tbody>
</table>

=\text{B2}-(\text{C2}+\text{D2}+\text{E2}) \ldots \text{Cell formula}
What Failing SI Means - Negative margin

Probability a parameter is a certain value

- limit

Mean

+ limit

The integral is the failing population, Pf

Pf*volume*cost/unit = cost

Not the whole story – A bad name can cost billions
Simple I/O Architecture

- Pre-'00 the most common computer I/O interface was synchronous memory transfer
  - Intel Xeon 100 MHz bus about the last bus in this class
- Clock distribution is a challenge - more on this later
Synchronous Memory Elements - Operation

**Operation**
- A data signal (\textit{in}) that is present at the input to the flip-flop is "latched" into the flip-flop by the rising edge of the input clock signal (\textit{clk}).
- On the next rising edge of \textit{clk}, the data signal is released to the output of the flip-flop (\textit{out}).
- This means data is clocked out of device \textit{a} on one clock edge and received at device \textit{b} on the next clock edge.
- This is also called \textit{common clocking}. 
Synchronous Memory Elements - Timing

Timing

- Valid data must be present for a minimum amount of time prior to the input clock edge to guarantee successful capture of the data. This is known as setup time, $T_{\text{setup}}$.
- Data must remain valid for a minimum amount of time after the input clock edge to guarantee that the proper value is captured. This is called hold time, $T_{\text{hold}}$. 
Simple Flight Time Concept

- The time it takes a signal to travel from device a to device b or the delay between transmitted (a) and received (b) signals.
  - This is **not** the definition that SI engineers use in a timing budget.
  - There are issues with timing budgets and device timing parameters that make this a **poor definition**.
  - We will develop the exact definition of flight time for SI later.
- Do not confuse with the AC with terms for propagation delay.
- SI engineers use the term propagation delay but it is not the same as AC propagation delay. We will develop the exact definition of propagation delay later. For now let's consider all delays the same.
- AC is frequency domain analysis.
We wish to use the clock to control the transmission of data from the latch in the source (a) to the latch in the destination (b).

The initial clock pulse causes the source latch to release the data onto the interconnect.

The next clock pulse causes the destination latch to capture the data that was transmitted on the interconnect.

We have 1 full clock cycle to get the data from the source to destination.
Transmit Clock Sequence

1. Initial (driving) clock pulse transmission from clock generator to source.
   
   a) \( T_{\text{drv clk}}(a) \) = delay of the clock buffer circuit connected to the source (\( a \)) from node 1 to node 1a.
   
   b) \( T_{\text{prop clk}}(a) \) = delay of the interconnect that between \( \text{clk} \) & \( a \).
2. Data transmission from source to destination.
   a) $T_{drv} = \text{delay of the output buffer circuit for the data signal.}$
   b) $T_{prop} = \text{interconnect delay between source and destination.}$
   c) $T_{setup} = \text{delay of the input buffer plus the flip-flop setup requirement.}$
3. Second (receiving) clock pulse transmission from clock generator to destination.

a) \( T_{\text{drv}_{-} \text{clk}}(b) \) = delay of the clock buffer circuit connected to \( b \).

b) \( T_{\text{prop}_{-} \text{clk}}(b) \) = delay of the interconnect between \( \text{clk} \) & \( b \).

c) Ideal assumption: \( T_{\text{drv}_{-} \text{clk}}(a) = T_{\text{drv}_{-} \text{clk}}(b) \) & \( T_{\text{prop}_{-} \text{clk}}(a) = T_{\text{prop}_{-} \text{clk}}(b) \)
Clock Skew

- What happens if the clock signals at the source and destination are not in phase?
  - What if the clock arrives at the destination before it reaches the source? Vice-versa?
- What are the sources of uncertainty in the phase relationship between different clock signals?
- Clock Skew: pin-to-pin variation in the timing of input clock at each agent (source & destination, in our example) on a bus.
- The net effect of clock skew is that it can
  - reduce the total delay that signals are allowed to have for a given frequency target.
  - require larger minimum signal delays in order to avoid logic errors. (We’ll cover this in more detail shortly.)
Sources of Clock Skew

Clock skew is caused by:

- variation between the clock driver circuits in a given part ($T_{\text{drv}}$).
- variation in the loading between different agents on the bus ($C_L$).
- variation in interconnect characteristics ($Z_0$, $\tau_d$).
- variation in electrical lengths. What is electrical length?
Clock Jitter

- Cycle to cycle variation of clock
- Reduces the time it take for data to get from transmitter to receiver
- Jitter + Skew = Clock uncertainty for setup
- Skew = Clock uncertainty for hold
  - Hold uses same cycle of clock
  - In many cases we can ignore certain types of jitter
- There are other types of jitter - more advanced topic
Skew & Jitter Example

- 100 MHz bus
  - Minimum clock period = 10 ns

- Given:
  - Maximum skew = 250 ps
  - Maximum edge-edge jitter = 250 ps.

- Calculate the minimum effective clock period:
  - minimum effective period =
    - minimum period − maximum skew − maximum jitter
  - min effective period = 10.0 ns − 0.25 ns − 0.25 ns = 9.5 ns

- Therefore, maximum allowed for silicon plus interconnect delay is 9.5 ns.
Setup Timing Diagram & Loop Analysis

CLOCK @ clk input
CLOCK(a) @ clk output
CLOCK(a) @ a
DATA @ a
DATA @ b
CLOCK(b) @ clk output
CLOCK(b) @ b

\[ T_{cycle} + T_{drv\_clk(b)} + T_{prop\_clk(b)} - T_{jitter} - T_{setup} - T_{margin} - T_{prop} - T_{drv} - T_{prop\_clk(a)} - T_{drv\_clk(a)} = 0 \]
Hold Timing Equation

- Uses same clock edge

- Hold equation

\[
T_{drv\_clk}(a) + T_{prop\_clk}(a) + T_{drv} + T_{prop} - T_{margin\_hold} - T_{hold} - T_{prop\_clk}(b) - T_{drv\_clk}(b) = 0
\]

- Define
  - Clock Delay
    \[
    T_{clk} = T_{drv\_clk} + T_{prop\_clk}
    \]
  - Clock Skew
    \[
    T_{skew\_setup} = T_{clk}(b) - T_{clk}(a)
    \]

- Simplify

\[
T_{margin\_hold} = T_{drv} + T_{prop} - T_{hold} - T_{skew\_hold}
\]
Manufacturability Considerations

- Sources of variability in silicon:
  - manufacturing process (e.g. silicon gate length)
  - operating temperature (MOS speed $\downarrow$ as temp $\uparrow$)
  - operating voltage (MOS speed $\uparrow$ as voltage $\uparrow$)

- Impact: variability leads to a range of values for driver and receiver timings

- Example: Pentium® Pro GTL+ timings
  - Minimum driver valid delay = 0.55 ns
  - Maximum driver valid delay = 4.40 ns
  - Maximum receiver setup time = 2.20 ns
  - Maximum receiver hold time = 0.45 ns

- Sources of interconnect variability:
  - Manufacturing variation ($Z_0, \varepsilon_r$)
  - Trace length variation (e.g. 144 signals for FSB)
Revised Timing Equations

- Product specifications must comprehend the expected variation.
- We need to modify the setup & hold equations:

  **Setup**
  \[ T_{\text{margin\_setup}} = T_{\text{cycle,min}} - T_{\text{drv,max}} - T_{\text{setup}} - T_{\text{prop,max}} - T_{\text{skew\_setup}} - T_{\text{jitter}} \]

  **Hold**
  \[ T_{\text{margin\_hold}} = T_{\text{drv,min}} + T_{\text{prop,min}} - T_{\text{hold}} - T_{\text{skew\_hold}} \]

- The setup equation defines the minimum clock cycle time (max frequency) in terms of the maximum system delay terms. We want \( T_{\text{margin\_setup}} \geq 0 \).
  - Excessive system delays can be handled by increasing cycle time, at the cost of reduced performance.

- The hold equation defines minimum system delay requirements to avoid logic errors due to hold violations. We want \( T_{\text{margin\_hold}} \geq 0 \).
  - Minimum delay violations cannot be fixed by increasing cycle time. Why?
Skew & Jitter Example

- **100 MHz bus**
  - Minimum clock period = 10 ns

- **Given:**
  - Maximum skew = 250 ps
  - Maximum edge-edge jitter = 250 ps.

- **Calculate the minimum effective clock period:**
  - \( \text{min effective period} = \text{minimum period} - \text{maximum skew} - \text{maximum jitter} \)
  - \( \text{min effective period} = 10.0 \text{ ns} - 0.25 \text{ ns} - 0.25 \text{ ns} = 9.5 \text{ ns} \)

- Therefore, maximum allowed for silicon plus interconnect delay is 9.5 ns.
Device Specs and Test Loads

- Device specifications vs. system conditions
  - The manufacturer guarantees that the parts meet the values in the timing specifications when driving into the “spec load”.
    - This is really the only way devices can be tested.
  - The spec load is typically equal to the load presented to the device by the test environment.
  - This spec load is generally not the same as the load presented to the device by the system interconnect.
Impact of Spec Loads

- Since the spec load is NOT equal to the load on the device when placed in a system:
  - An output buffer will have a different delay in the system than in the test environment.

- To deal with this:
  - define new timing terms &
  - change the way we break the timings into separate components.
Flight Time

Clock Input to Transmitting Chip

Driver Pin into System Load

Driver Pin into Test Load

$T_{co}$

$T_{flight}$

$T_{drv}$

$T_{prop}$

Receiver Pin
Flight Time Explained

- Define $T_{co}$ (time from clock-in to data-out) as the delay from the input clock to the output data when driving into the test load.

- Define $T_{flight}$ (flight time) as the delay to the receiver minus the $T_{co}$.
  
  By defining the timings in this way, the flight time accounts for the propagation delay of the interconnect PLUS the difference between the driver delays when driving test load vs. the system load.

- Notice: $T_{drv} + T_{prop} = T_{co} + T_{flight}$
  
  We defined $T_{co}$ and $T_{flight}$ this way to guarantee the overall system timings remain the same.
Revised Timing Equations

Setup

\[ T_{\text{margin setup}} = T_{\text{cycle, min}} - T_{\text{co, max}} - T_{\text{setup}} - T_{\text{flight, max}} - T_{\text{skew setup}} - T_{\text{jitter}} \]

Hold

\[ T_{\text{margin hold}} = T_{\text{co, min}} + T_{\text{flight, min}} - T_{\text{hold}} - T_{\text{skew hold}} \]

- The system designer relies on the synchronous timing equations help her/him define the working flight time window (min-to-max), given the component timing specs.
- Ultimately, the equations provide a tool for a design team.
  - Use them to evaluate design trade-offs in order to achieve system performance (frequency) targets.
### Example: Bus Timing Spread Sheet - Setup times

<table>
<thead>
<tr>
<th>Tco Max (ns)</th>
<th>Tsu (ns)</th>
<th>Clk Skew (ns)</th>
<th>Clk Jitter (ns)</th>
<th>Length</th>
<th>Tflight</th>
<th>Tcyc</th>
<th>margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU 1</td>
<td>3.2</td>
<td>0.5</td>
<td>0.25</td>
<td>0</td>
<td>15.15152</td>
<td>11.40152</td>
<td>3.75152</td>
</tr>
<tr>
<td>CPU 2</td>
<td>3.2</td>
<td>0.5</td>
<td>0.25</td>
<td>5</td>
<td>0.86</td>
<td>10.04152</td>
<td>9.18152</td>
</tr>
<tr>
<td>Chip Set</td>
<td>7</td>
<td>1</td>
<td>0.25</td>
<td>7</td>
<td>1.204</td>
<td>9.197515</td>
<td>5.897515</td>
</tr>
<tr>
<td>CPU 3</td>
<td>3.2</td>
<td>0.5</td>
<td>0.25</td>
<td>7</td>
<td>1.204</td>
<td>9.697515</td>
<td>5.897515</td>
</tr>
<tr>
<td>CPU 4</td>
<td>3.2</td>
<td>0.5</td>
<td>0.25</td>
<td>10</td>
<td>1.72</td>
<td>9.181515</td>
<td>5.897515</td>
</tr>
</tbody>
</table>

**Tpd**: 172 ps/inch  
**Freq**: 66 MHz  
**Tcyc**: 15.15152 ns  
**5.897515 Min of margins**
Synchronous Timing Summary

- Synchronous memory elements require a stable data signal for a minimum amount of time prior to (SETUP) & after (HOLD) the input clock.

- Hold and setup conditions determine the minimum and maximum system delays.

- Setup and hold conditions can be analyzed by constructing timing loops in the timing diagrams.

- Component delays exhibit variation across process and environmental conditions. Interconnect delays contain variations due to design and process.

- Redefining driver and interconnect delays in terms of system and “spec” loads allows manufacturers to specify and test component delays.

- System timing equations provide a key tool for examining trade-offs during system design.
Extra credit Assignment #4

- Create Budget Spreadsheet for setup and hold
- Find and justify maximum frequency of operation
- Find all minimum lengths

<table>
<thead>
<tr>
<th>Tco Min</th>
<th>Tco Max</th>
<th>Tsu</th>
<th>Thld</th>
<th>Clk Skew</th>
<th>Clk Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ns)</td>
<td>(ns)</td>
<td>(ns)</td>
<td>(ns)</td>
<td>(ns)</td>
<td>(ns)</td>
</tr>
<tr>
<td>CPU 1</td>
<td>0.2</td>
<td>3.2</td>
<td>0.5</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>CPU 2</td>
<td>0.2</td>
<td>3.2</td>
<td>0.5</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>Chip Set</td>
<td>-0.5</td>
<td>7</td>
<td>1</td>
<td>-0.1</td>
<td>0.25</td>
</tr>
<tr>
<td>CPU 3</td>
<td>0.2</td>
<td>3.2</td>
<td>0.5</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>CPU 4</td>
<td>0.2</td>
<td>3.2</td>
<td>0.5</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

- Tpd: 172 ps/inch
- Freq: 66 MHz
- Tcyc: 15.15152 ns