Acknowledgment: Figures and data in this talk are excerpted from the papers below:

**High-Performance Electrical Signaling**
William J. Dally\(^1\), Ming-Ju Edward Lee\(^1\), Fu-Tai An\(^1\), John Poulton\(^2\), and Steve Tell\(^2\)

**CMOS High-Speed I/Os — Present and Future**
M.-J. Edward Lee\(^1\), William J. Dally\(^1,2\), Ramin Farjad-Rad\(^1\), Hiok-Tiaq Ng\(^1\),
Ramesh Senthinathan\(^1\), John Edmondson\(^2\), and John Poulton\(^1\)

**Designing Bang-Bang PLLs for Clock and Data Recovery in Serial Data Transmission Systems**
Richard C. Walker

**Near Speed-of-Light Signaling Over On-Chip Electrical Interconnects**
Richard T. Chang, Student Member, IEEE, Niranjan Talwalkar, Student Member, IEEE, C. Patrick Yue, Member, IEEE,
and S. Simon Wong, Fellow, IEEE

**LVDS I/O Interface for Gb/s-per-Pin Operation in 0.35-µm CMOS**
Andrea Boni, Member, IEEE, Andrea Pierazzi, and Davide Vecchi

**Figures of Merit to Characterize the Importance of On-Chip Inductance**
Yehea I. Ismail, Eby G. Friedman, and Jose L. Neves\(^1\)
High-Speed I/O

- Why standard approaches are slow
- Incident-wave signaling
- Line equalization and eye diagrams
- Clock recovery
- Coding and framing
Standard CMOS I/O

Sending Chip | PC Board Trace or Cable | Receiving Chip

CMOS Inverter | | CMOS Inverter

Slow (100 MHz rates, or less).
Power hungry (1nJ/bit, or more).
Bandwidth decreases with trace/cable length.

Simple models will help us understand why, and how to do better.
Trace/Cable can be modeled as a distributed RLC circuit.

Looking into a long cable, a circuit sees a characteristic impedance that is independent of the cable length.

\[ Z_0 = 50\Omega \]

A typical trace/cable has a characteristic impedance of about 50 ohms.
The highest frequencies of a pulse edge on a wire travel approach the "speed of light" of the wire medium ($c_w$).

And so, the fast rising edge of a pulse takes about $L/c_w$ seconds to traverse a wire of length $L$.

For our example, assume $L/c_w = 4$ns.
Inverter models.

Typical output impedance is 400 ohms.

Typical input impedance of pad is 1000 ohms.

We now have the tools to model a rising edge sent from chip to chip.
15 up-and-back traversals are needed to “ring-up” $V_r$ to 90% of $V_{dd}$!

The impedance mismatches at each cable end reduce the pulse heights and launch the cyan reflection waves.
Incident-Wave Signaling
Incident-wave I/O

Kill reflections by making input and output impedances match the line impedance.

Each bit is communicated by the first arriving wave (the incident wave).

Several bits can be in-flight on the wire at once.
Differential, low-voltage.

Direction of current (+/-) codes one/zero.
Magnitude of current sets voltage (V=IR).

Energy dominated by DC power ...
Line Equalization
Wire attenuation

Once we adopt the incident-wave approach, what limits our bandwidth?

The result is **intersymbol interference**. The slow-traveling low components of earlier bits *swamp* the fast edge of a new bit.
Equalization

Ideally, we would send pulses whose frequency spectrum looks like this.

Or more generally, we want to send an ideal pulse that has been \textbf{equalized} to invert the wire frequency response.
Equalization

How can we overcome intersymbol interference?

A simple 4-tap equalizer, sufficient for 4 Gb/s on a differential wire pair.

Original wire:

4-tap hi-pass EQ:

Combined, flatter response.

CMOS High-Speed I/Os — Present and Future

From: M.-J. Edward Lee¹, William J. Dally¹,², Ramin Farjad-Rad¹, Hiok-Tiaq Ng¹, Ramesh Senthinathan¹, John Edmondson¹, and John Poulton¹

CS 250 L13: High-Speed I/O
Equalization

The 4-tap equalizer, as seen in the time domain.

What receiver sees when we send non-EQ’d pulse.

Ideal pulse:

Ideal pulse after EQ:

What receiver sees when we send EQ’d pulse.

Figure 6: Jitter Response of a PLL and a MDLL
Eye Diagrams
Eye diagrams

A way to visualize bits on a wire.

Oscilloscope trace of receive-end of wire.

Fold the trace at the clock period. If the received signal is clean, an open eye is seen.
Eye diagrams

Receive waveform without EQ. Eye is closed, due to inter-symbol interference.

Receive waveform with EQ. Eye is open, due to boost of high-frequency pulse components.
What limits bandwidth?

**Uncertainty time.**
All sources of temporal uncertainty (jitter, etc).

**Rise time.** Depends on drive current of output transistor.

**Aperture time.** How long it takes sense amp to make +/- decision.

All should improve with process, but all have fundamental limits (thermal noise, non-perfect line equalization, etc).
Clock Recovery
Incident-wave I/O

How does the receiver place the clock edges?

Sense amp is clocked.

Sense amp should be clocked on positive edge of this clock. But receiver has to generate this clock from the data!

![Diagram showing the concept of incident-wave I/O and the clocking of the sense amp.](image-url)
Alexander detector

Make an initial guess for clock frequency, and clock in data on both edges.

If we guess clock frequency perfectly, A and B would be two adjacent bits, and T would be random …
Alexander detector

If our clock frequency guess is wrong, we can use this truth table on A, B, and T to see if the edges are arriving early or late.

<table>
<thead>
<tr>
<th>State</th>
<th>A</th>
<th>T</th>
<th>B</th>
<th>UP</th>
<th>DOWN</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>hold</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>early</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>error</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>late</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>late</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>error</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>early</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>hold</td>
</tr>
</tbody>
</table>
We can embed this truth table as logic gates, and use it to tune a VCO’s frequency to recover the transmitter’s clock.

Real-world versions track duty-cycle changes, etc ...
Coding and Framing
For clock recovery to work (and other reasons), we need to restrict the input data stream.

Input stream: a “river” of bits

010010101001001010101010101 ... 

We restrict the characteristics of this river ...
A river of bits ...

0100101010010010101010101101010010 ...

\[ M \]

The first \( M \) bits will be received with very high error.

Bits \( M+1 \) onward will be received correctly with high probability (but not 100% correct).

Low error condition holds as long as bit river keeps flowing at a constant clock rate.

At most, \( N \) consecutive 1’s or 0’s may appear in the river. \( N \) may be as low as 5.

Over “long” stretches of bits, the number of 1’s sent must equal the number of 0’s sent.
Given an arbitrary bitstream, we can code it to have these desired properties.

Example: 8b/10b coding

Bits we want to send:

... 0100101010010010101010101101010010 ...

Each 8 bit sequence recoded as 10 bits

... xxxxxxxxxxxx ...

Recoding algorithm guarantees 0/1 restrictions are met.

Code also offers “out-of-band” 10-bit codes that act as control characters, which higher-level protocols can use to frame the stream, etc.

Does not do error-correction on user data ...
Next Week: Project Details

<table>
<thead>
<tr>
<th>Tue Oct 13</th>
<th>All</th>
<th>Group meetings: Review initial proposal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu Oct 15</td>
<td>All</td>
<td>Group meetings: Review initial proposal.</td>
</tr>
</tbody>
</table>

Have a good weekend!