
Lecture 12:

Introduction to Link Design

Computer Systems Laboratory
Stanford University
horowitz@stanford.edu

Copyright © 2000 by Mark Horowitz

EE371

Lecture 12-1

Horowitz

Overview

Reading

Chapter 19 - High Speed Link Design, by Ken Yang, Stefanos Sidiropoulos

Introduction

There has been an explosion of interest in high-speed IO over the past 10 years. It is now being used in products ranging from DRAMs to interconnects in high-end servers and routers. This lecture will give an overview of the basic elements needed in a high-speed link, and will set up what we will discuss in the next few lectures. We start by looking at what makes driving external wires different from the work we have done driving internal wires.

EE371

Lecture 12-2

Horowitz

Basic IO Design

All external signal paths can be represented by three elements:



Transmitter

- Converts bits to an analog electrical signal to transmit on pin

Channel

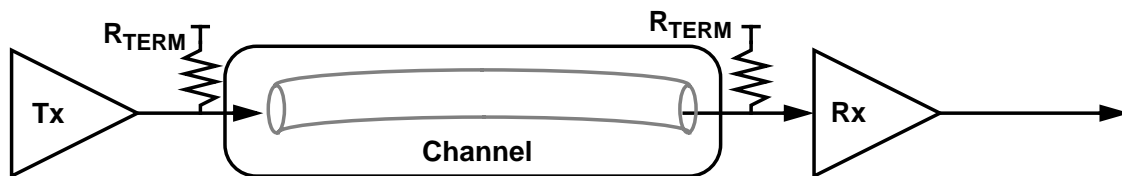
- Transmission media for the analog signal, which is sometimes pretty nasty to signal fidelity

Receiver

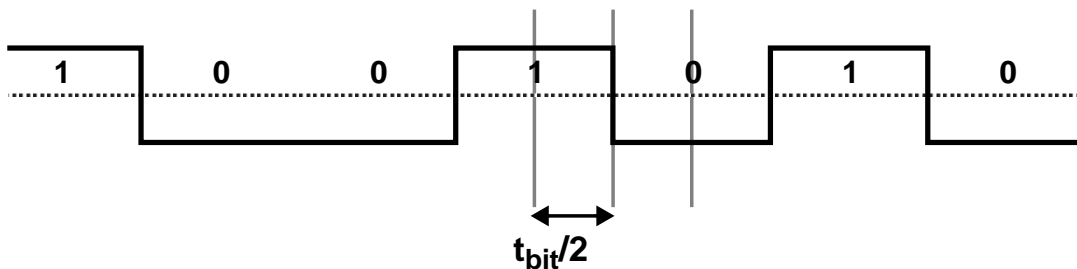
- Convert the analog signals back to bits (quantized in voltage and time)
- It is the need to convert back to digital signals that can be a problem

Basic Issues

Voltage Margins: making sure voltage quantization yields the correct result:



Timing Margins: knowing which bit is which:



- For a high-speed signal, bits are pretty short ($< 1\text{ns}$)

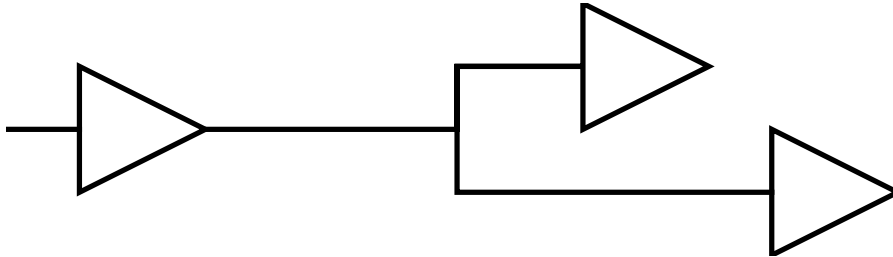
What is the Problem

Deal with analog signals all the time on chip. What is the issue with IO?

- Why are timing and voltage margins an issues for external signals?

The speed of light is not infinite

- Wire connecting the gates is not an equipotential
 - It is not even an RC line
- So life gets complicated

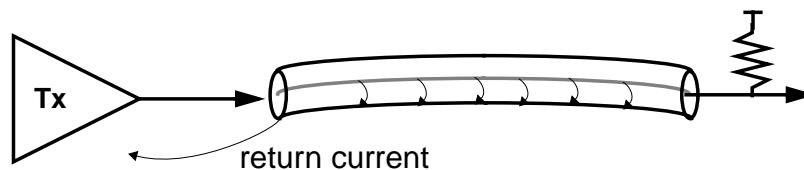


Finite Speed of Light

Signals on wires must experience delay in reaching destination -- $T_d = L/v$

- Bit arrive at a different time then when they were sent
- Must sample the data at the 'correct' time
- And the clocks to the two chips might not arrive at the same time

Wires store energy



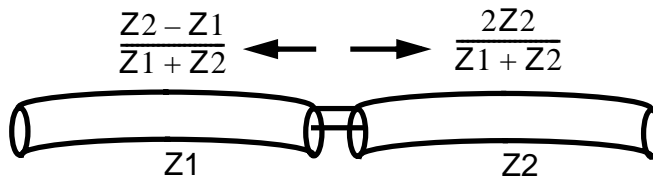
- While the signal is travelling on the wire,
 - Current from Tx is initially set by the wire (can't see resistor at $t=0$). V/i for the line is its impedance, Z , and is set by the geometry of the wire
 - Signal is a pair of currents that propagates out from source (current and return)

Transmission Lines

(What wires are called when you notice v is not infinite)

Two constraints govern behavior at any junction:

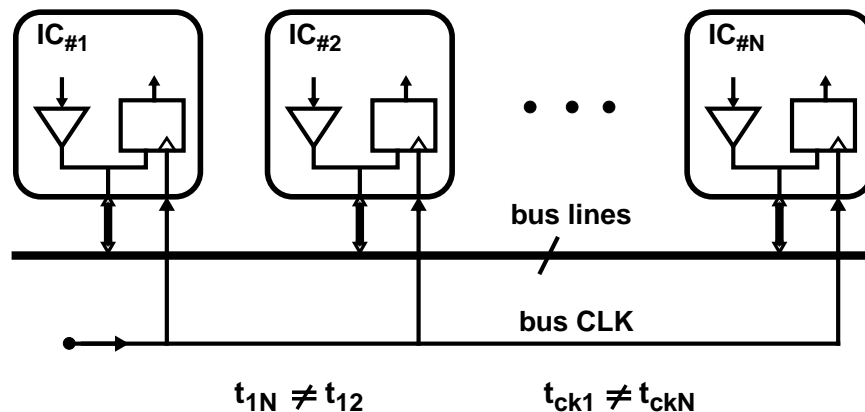
- Voltage at all components must be equal
Electrically connected
- Power flow into junction must equal power flow out of junction
Conservation of energy
- Leads to reflections



Signals can return to the source, as well as propagate forward

Note: the signal return path is usually drawn; it is as important as the signal

Conventional Buses

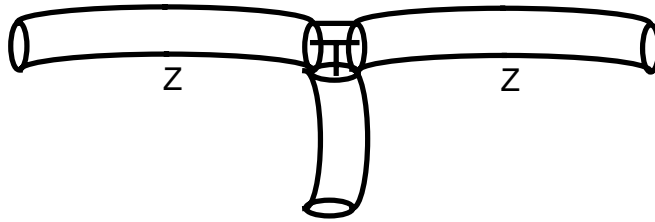


Have many problems

- Electrical distances from chip to chip vary, have stubs in the transmission lines
 - Make signal environment difficult

Stubs

Can't connect to the middle of a transmission line without causing trouble:



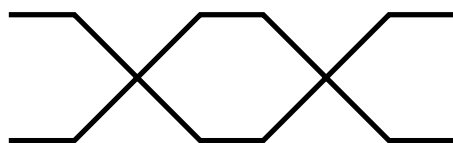
Unless stub is short, it will cause reflections, since energy will split and only part will go into each transmission line segment

- Add lots of 'noise' to the signal
- Slow the signal propagation
 - Energy must reflect off all the stubs before settling down

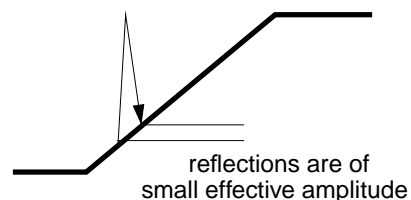
What Length is Short Enough?

Length is compared to distance light travels over what time?

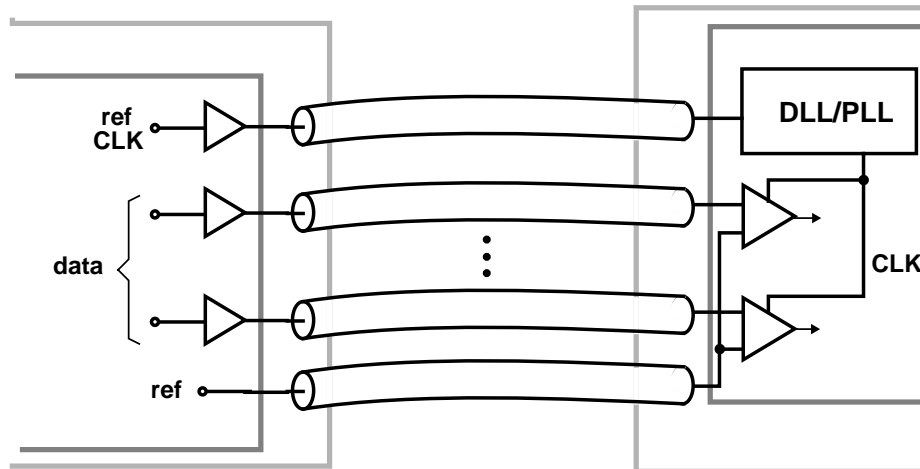
- Related to the rise time of the signal, not frequency
 - If the signal does not change much by the time the reflection returns, you won't see the reflection -- the reflection settles during the transition
 - If stub is short, energy storage in line is not significant
 - Model by lumped parameters
- Since fewer transmission lines are better, we want to slow signal edge rates
 - Bit should be 1/3 rise, 1/3 high, 1/3 fall
 - Risetime should be 1/2 to 2/3 of bit time



Nice edge rate for bits



High Speed Links

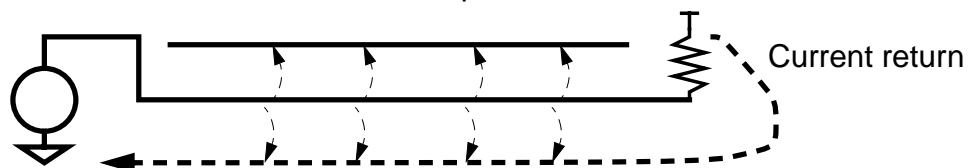


Almost all high-speed links are point to point.

- Sending a clock as a additional data bit helps determine timing if the data cable lengths are all matched
- Called source synchronous links

Coupling

All transmission lines need a current return path



And are really differential systems

- Voltage difference between input and return is equal to the voltage difference between output and the voltage of the return at that end. The two return need not be at exactly the same voltage.

If return path is far away, another signal can 'see' the signal too

- Coupled transmission lines
 - $V_{out} = a V_{in1} + b V_{in2}$
- Coupling on PC board, and coax cables are small
- Coupling on Twisted pairs and IC packages is significant

Metrics

Need to measure links

- Look at a couple of metrics

Performance

- Bit rate
 - Normalize out the fabrication technology
 - As technology scales, how fast will link become?
 - Use FO4
- Bit Error Rate (BER)
 - Link reliability
 - Signal to noise ratio
 - Should scale with technology, performance easy to predict

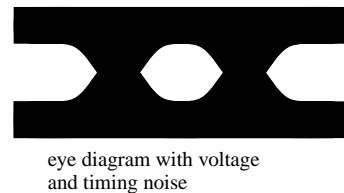
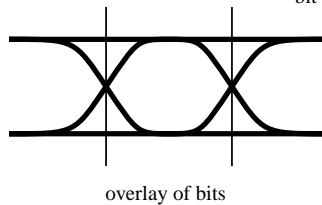
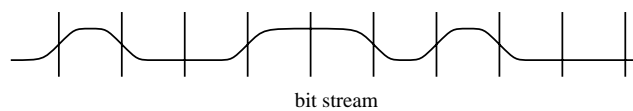
Bit Error Rate

Receiver needs to convert analog signal to digital value

- Possible to make an error -- noise is greater than signal
 - Voltage noise
 - Timing noise

Reduces the amount of signal available

- BER
 - Depends on signal to noise ratio (SNR)
 - Also depends on noise statistics

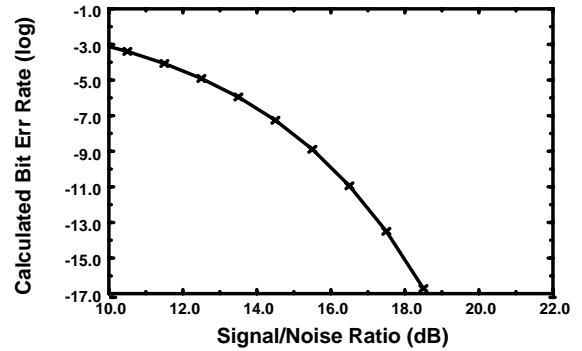


BER and SNR

Many textbooks give plots like:

- Show BER exponentially related to SNR
- But assume gaussian noise
- Real noise is not gaussian
 - Small white (or colored noise) gaussian
 - Large self-induced noise

Not true noise, but hard to calculate



Take true signal (signal - self-induced noise) and compare to white noise

- Give effective SNR
- White noise is small (mVs, but hard to estimate)
- Can make BER for most electronic links (non-optical) very, very small

Summary

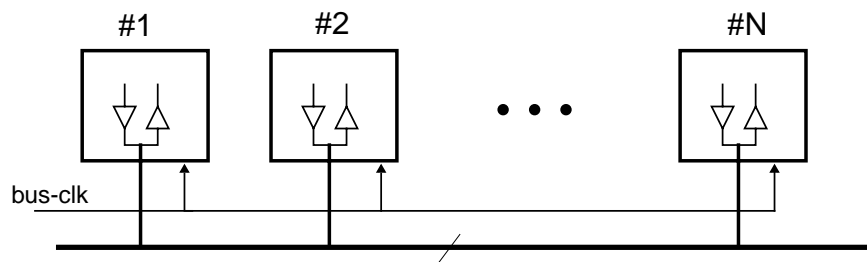
Electronics need to deal with:

- Transmission line impedance
 - Need to have some method of dissipating the energy
 - Need to drive relatively low impedances (< 100 ohms)
- Noisy Signals
 - Some noise is proportional to signal amplitude
 - Can't perfectly set impedance or resistance, will have reflections
 - Coupling of other signals
 - Fundamental noise in analog world
- Line Delay
 - Need to extract timing from signal or some other reference.

Transmitter and Receiver Design Outline

- System Architectures
 - What does the system look like ?
- Noise
 - What does the “signal integrity engineer” have to do ?
- Drivers
 - How do I generate these 500-mV swing signals out of a 3.3-V chip ?
- Receivers
 - How do I restore these 500-mV signals to 3.3-V ?
- Bidirectional Signalling
 - What can I do to save pins and wires ?

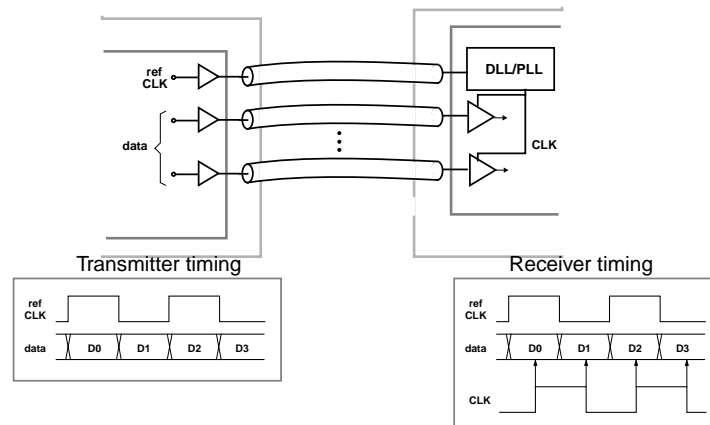
The Conventional Bus Bottleneck



- Timing is uncertain:
 - Distances of data from chip to chip and from clock to any chip vary
 - -> So we need to slow down to have margins for the worst case
- Signals don't look that great either:
 - Multiple discontinuities on bus transmission line create reflections
 - Using a conventional buffer to drive a low impedance generates noise and burns a lot of power (3.3V to 50 Ohms ~ 210 mWatts !!)

Point-to-Point Parallel Links

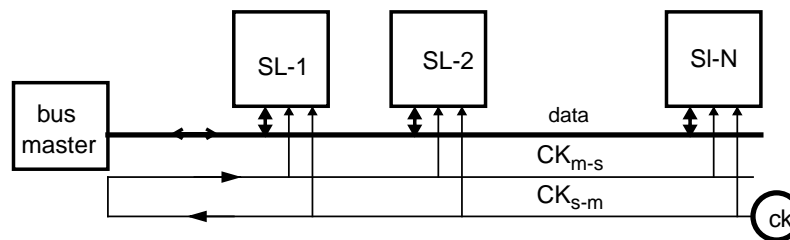
- “Source Synchronous”/low-swing design:



- Bandwidth is set by delay uncertainty and *not total delay* through wires
Uncertainty is created by: skew, jitter, rcv/xmit offsets, setup+hold time .
PLL/DLL used to create the 90° clock on the receiver side.
- Use small swing signals to minimize power and noise

High Speed Buses

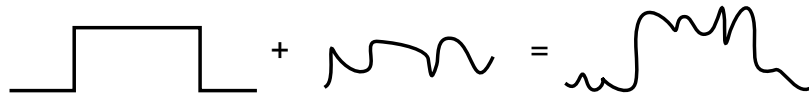
Rambus channel: talk only from master->slave, or slave->master



- Same timing idea: *make sure data & clock travel the same distance*
 - Now both transmitter and receiver need to align with the system clock
- More difficult environment than point-point:
 - Multiple discontinuities on transmission line are dealt with careful package and board design
- Again PLL/DLL used for timing. More on these later...

Noise

Need to send signals that can be distinguished from environment noise



- **Independent noise**

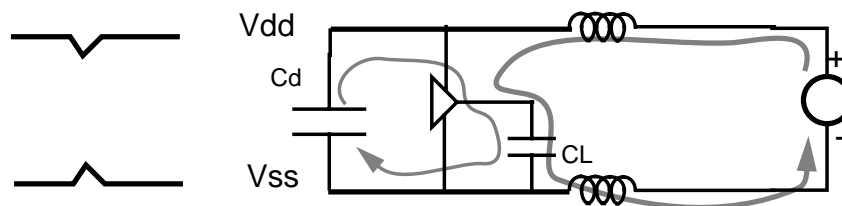
- Gaussian (unbounded) but very small probability ($< 10^{-20}$) for appreciable (1-mV) noise.
- Unrelated power supply noise: background activity of the chip and other drivers switching unpredictably.

- **Proportional noise** (scales with signal swing):

- Self Induced di/dt noise (also called signal return noise)
- Crosstalk/Coupling from other signals.
- Mismatch -> reflections

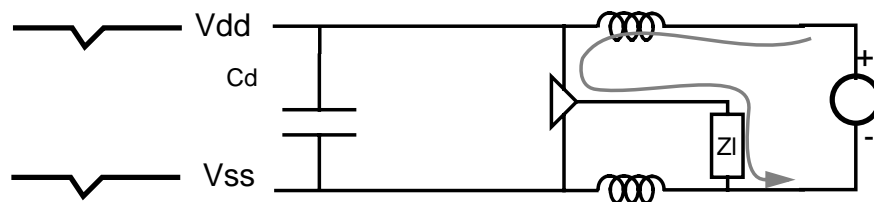
Aside on Supply Noise

- On-chip switching



Causes Vdd and Vss to droop out of phase. On chip Vdd-Vss capacitance can be used to minimize this effect by supplying the required charge.

- Off chip driving



Causes Vdd and Vss to move in phase. The on chip Vdd-Vss capacitance does not help minimize the noise. It prevents the supply from collapsing.

Noise: What can you do.

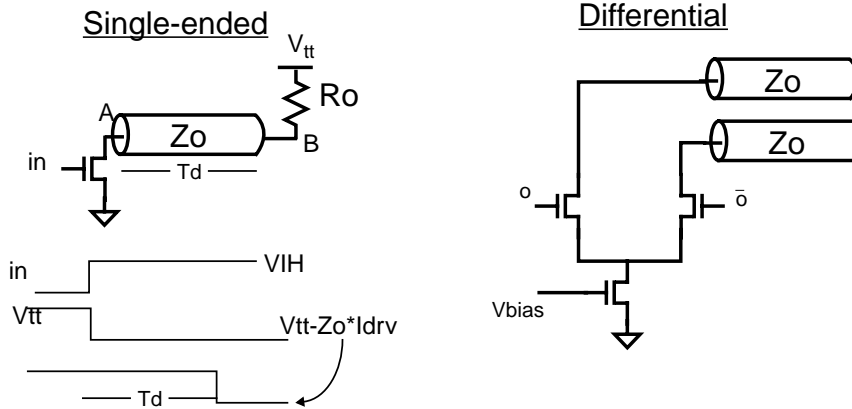
- Overpower it with large signal swings
 - Works great for Gaussian noise and unrelated bounded noise
- Cancel by using differential signalling
 - Works for self-induced dl/dt noise crosstalk and unrelated PS noise
 - Pseudo-differential signalling works to a certain extent
- Minimize by careful/conservative design
 - Don't route large swing signals close to low swing signals
 - Route differential signals close together

Always do worst case estimation: E.g. $N \cdot L \cdot dl/dt$ use max N, max L, FF corner to get the max dl/dt

Output Drivers

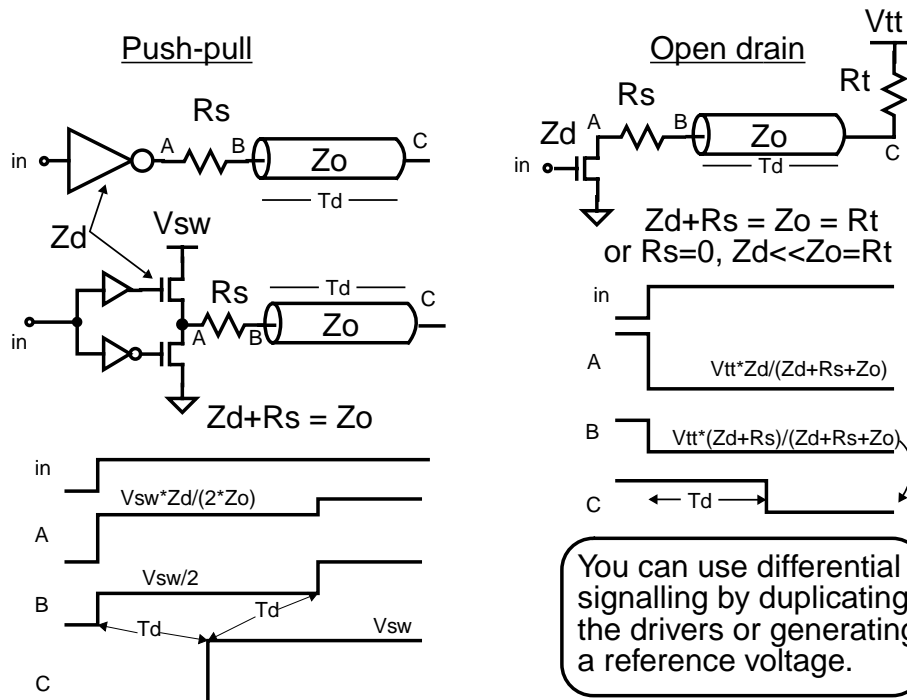
- Output Impedance:
 - High -> parallel terminated current source
more power, better supply rejection
 - Low -> series terminated voltage source
lower power, poor supply rejection
- Output swing: 300 mV - 1 V (scalable with V_{dd})
- Differential or Single-Ended
 - Differential: more wires and pins but better noise immunity
 - Single-Ended: Pure single ended has lots of problems due to unrelated PS noise. Usually generate a reference and share it among many pins. Still more problems with noise than fully-differential.

High Impedance Drivers

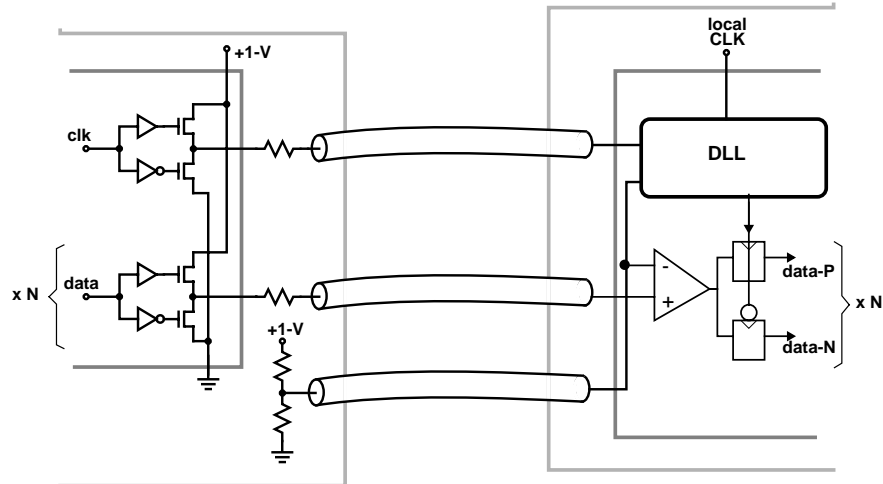


- Keep current source in saturation region
 $V_{tt} - V_{swing} > V_{dsat}$ of transistor
- Keep driver current constant:
 - > IR drops will shift the bias point: use thick Vss lines or current references
 - > can use feedback to set Vbias (or adjust tail-CS width)

Source Terminated Drivers



Example: Push-pull signalling



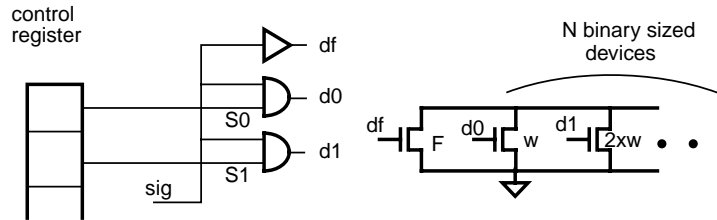
Reference voltage can be generated on-chip but noise tracking is limited
Loading of reference on the receiver side is much larger than that of the signal

Driver Issues

- Driver Impedance/Current control
use active circuits to compensate for process/supply/temp variations
- Drivers turn-on time is an issue (slew rate)
If turn on is too fast it will increase the self-induced di/dt noise so we need to control the slew rate of the pre-driver.
This is hard to do: if you compensate for the FF corner the SS corner will become too slow.

Driver Impedance/Current Control

- Need to match the driver impedance to the line impedance ($Z_d=Z_o$) or regulate the current to keep the swing constant.
- Adjust the width of the driver digitally



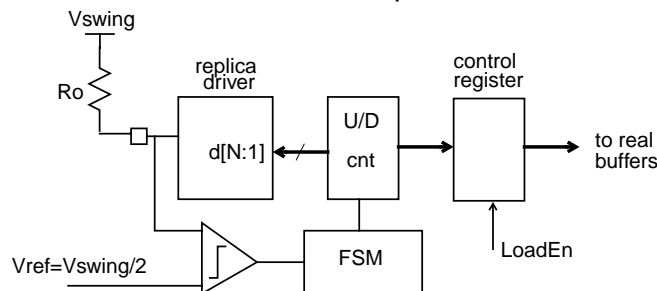
F should give $Z_{max} > Z_o$ at FF corner

$(2^N - 1) \times W$ should give $Z_{min} < Z_o$ at SS corner ($S_0 = \dots = S_N = 1$)

Driver Impedance Control (cont'd)

How do you set the value of the control register ?

- Set it with scan at system power-up (what about variations?)
- Integrate a feedback mechanism with a replica driver



Move the value of the counter to the control register periodically

Glitches when changing from 011... to 100...

-> Assert *LoadEn* only when not transmitting

-> Change from binary weights to thermometer-like code

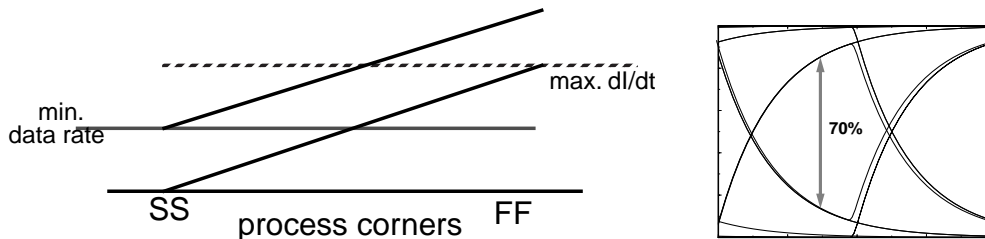
Output Slew Rate Control Problem

Sharp slew-rates introduces high-frequency components

EMI issue at the output and reflections from parasitics on the channel

So we need to control the slew rate of the pre-driver... but it is a hard problem.

Slow down the pre-driver?

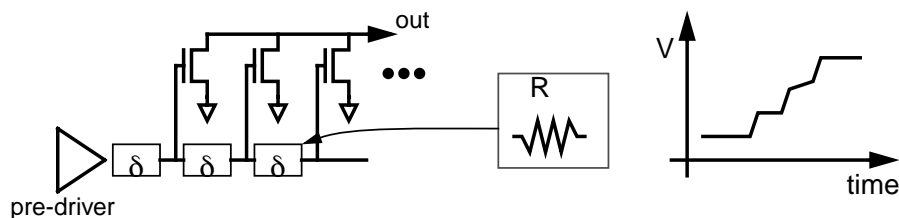


If you compensate for the FF corner the SS corner will become too slow and cause inter-symbol interference of the data.

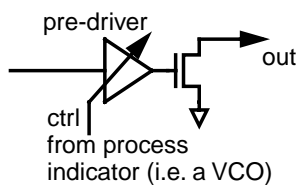
Slew Rate Control

- Delay the turn on.

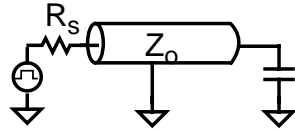
Use RC delay (or buffer delays) [TI]



- Set the pre-driver slew-rate using a control voltage from a process indicator [6].

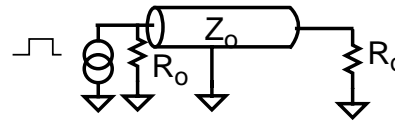


Output Driver Summary



Voltage-mode driver

- series-terminated voltage source
- lower power
- Worse supply rejection



Current-mode driver

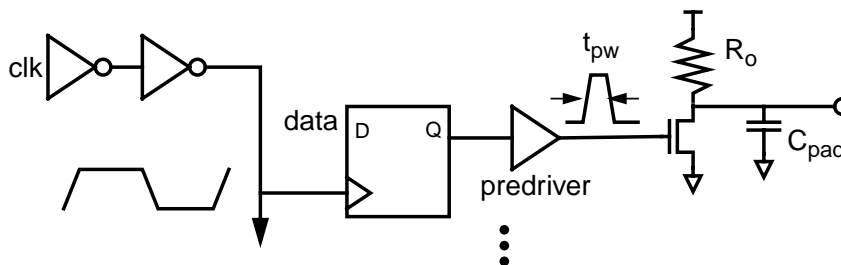
- parallel-terminated current source
- more power, less reflection noise
- better supply rejection

- Deal with process variations:
 - control the current and output impedance using a feedback.
 - control the slew rate using feedback
- Differential signalling reduces noise but uses 2x the number of pins.

Are we done?

- Not yet. What's the bandwidth limitation?

Where is the Bandwidth Limit?



$R_o C_{pad}$ at the output?

No, usually very small since $R_o \leq 50\Omega$.

Minimum pulse width (t_{pw})?

Maybe, $3x t_{prop-dly}$ of predriver.

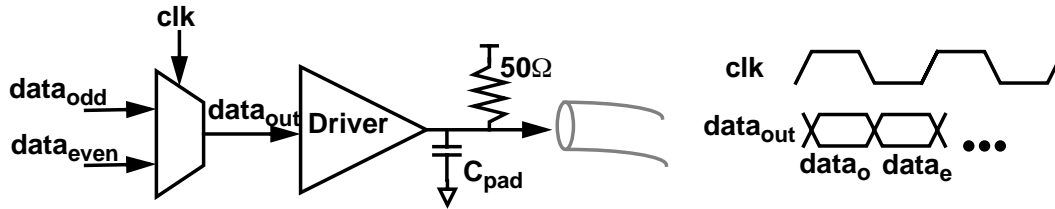
Clock cycle-time?

Yes, FO-4 buffer chain need clock period of 6-8 FO-4 delay.

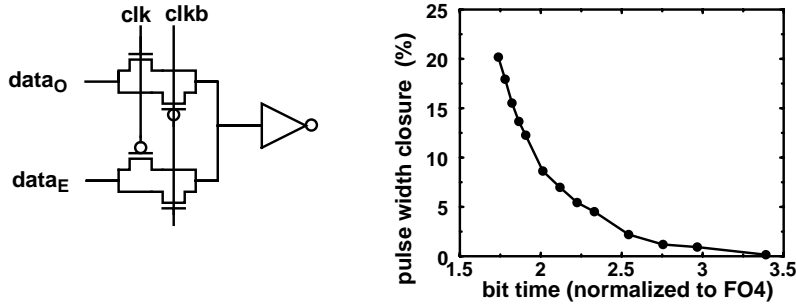
Solution: use more bits/cycle

Parallelism

Use multiplexer to improve the bandwidth.



2:1 multiplexer has a bit-time limit of 2 FO-4.

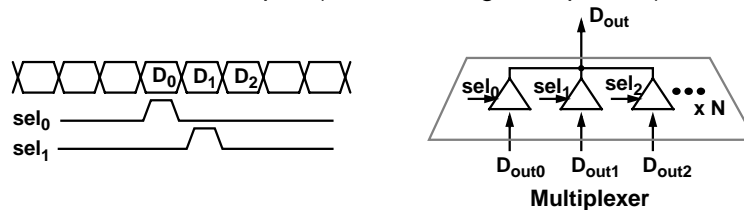


Clock is still limits bit-time (3-4 FO-4), but higher multiplexing is limited by mux

More Bits/Cycle

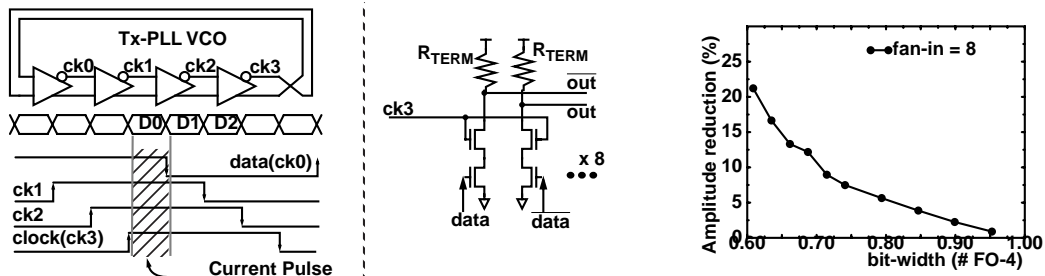
Use low swings and higher fan-in mux.

Convenient to mux at the output. (trades off larger output RC)

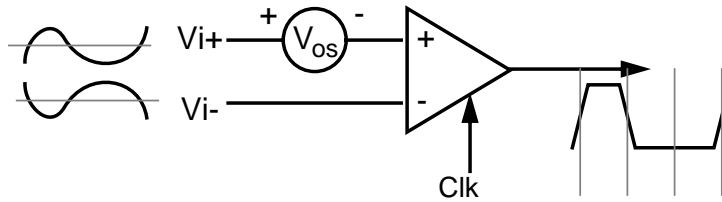


Limited by the minimum pulse width on-chip (2 FO-4),

Use multiple phases and overlapping currents. Reach bit-time of 1 FO-4.[11]



Receiver

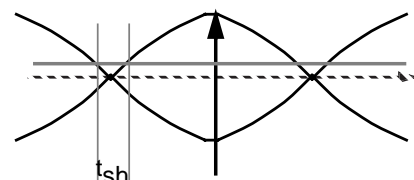
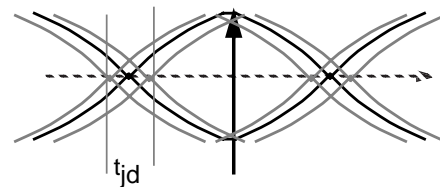
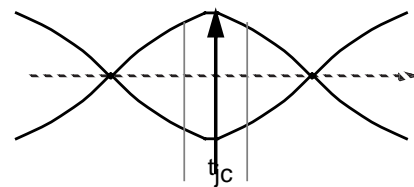


- Amplify and latch the signal stream into a digital bit sequence.
- Issues
 - bandwidth
 - resolution
 - limited by noise and offset
 - ensure good timing margin

Timing Margin

Factors that degrade the margin:

- Sampling clock jitter:
 - Transmitter clock
- Data jitter:
 - Transmitter clock
- Receiver uncertainty window:
 - offset, noise, metastability ($t_{\text{setup-hold}}$)



$$\text{Remaining: } t_{\text{margin}} = 0.5 * (t_{\text{bit}} - t_{\text{j_c}} - t_{\text{j_d}} - t_{\text{sh}})$$

Receiver Design

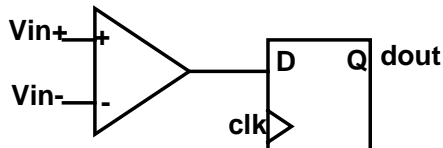
Differential vs single-ended:

Every receiver has a reference voltage (implicit for single-ended)

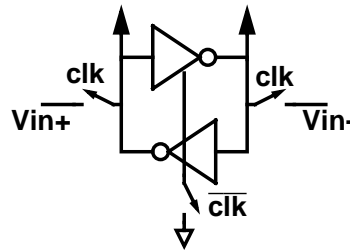
Differential receiver rejects common-mode noise — can be used for single-ended inputs (pseudo-differential).

Try to use the reference information sent along with the signal.

Circuit topology

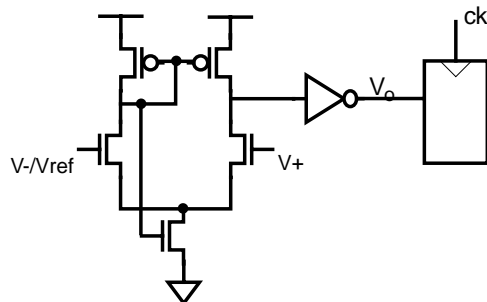


Amplifier followed by a latch.



Latching sense-amplifier structures

Amplifying receiver [1]



- Self biased amplifier with medium/high input common mode

self biasing improves P/N tracking.
can use the dual structure if inputs have low common mode.

- Resolution

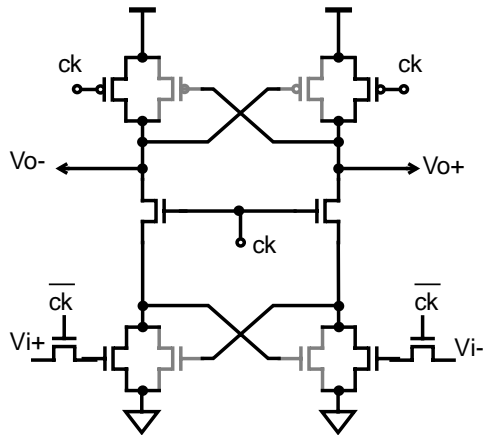
input-referred offset: transistor random mismatch (V_T , K_P) and systematic errors (V_{o_min} from latch)

- Timing Errors

The delay is sensitive to PS — increase the uncertainty on the switching time of V_o .
Setup-hold time depends on latch (which can be poor.)

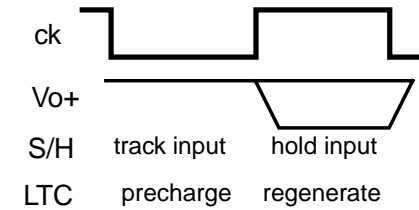
- Gain-bandwidth limitation introduces inter-symbol interference for high data rates. (4-6 FO-4)

Sampling receiver [7]



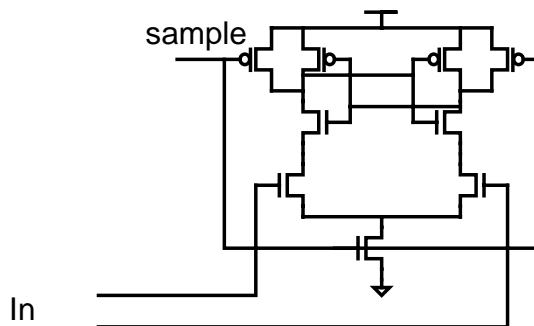
Grey device show cross-coupled inverters that regenerate.

Need a latch at the output to hold th data for the full clock cycle.



- No ISI because the outputs are equalized for each incoming bit.
- Slightly worse input offset than before: 50-100mV
Setup/hold window of < 100ps
- Be careful about sampling noise and charge-kick back.
- Bit-time is limited by the cycle-time (to have enough gain) of 6-8 FO-4.

Sampling Receiver

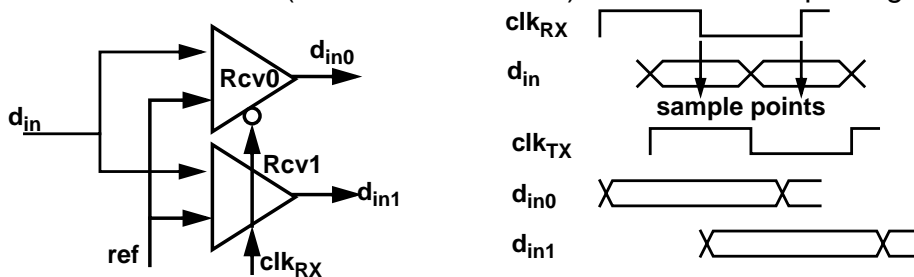


'Strong-Arm' Latch

- Small Kick-back onto inputs
- Good gain

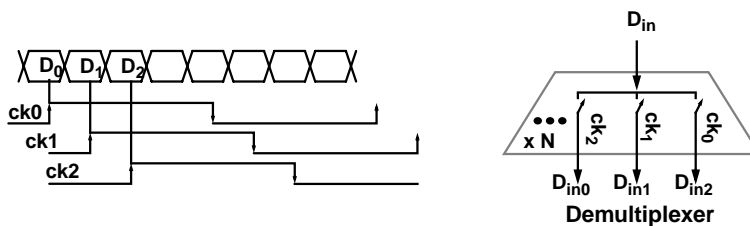
Demultiplexing

Double the data bandwidth (bit-time of 3-4 FO-4) with 2:1 demultiplexing



Can extend to higher bandwidth (~ 0.5 FO-4) [11]

Limit in data rate is really the sampling aperture of the samplers and not the cycle time of the latch.

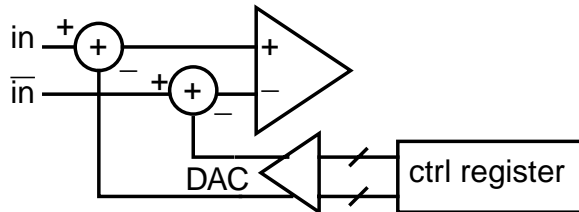


Input Offset Correction

Resolution is limited by offset (V_T and K_P) between differential inputs, but it's a static offset.

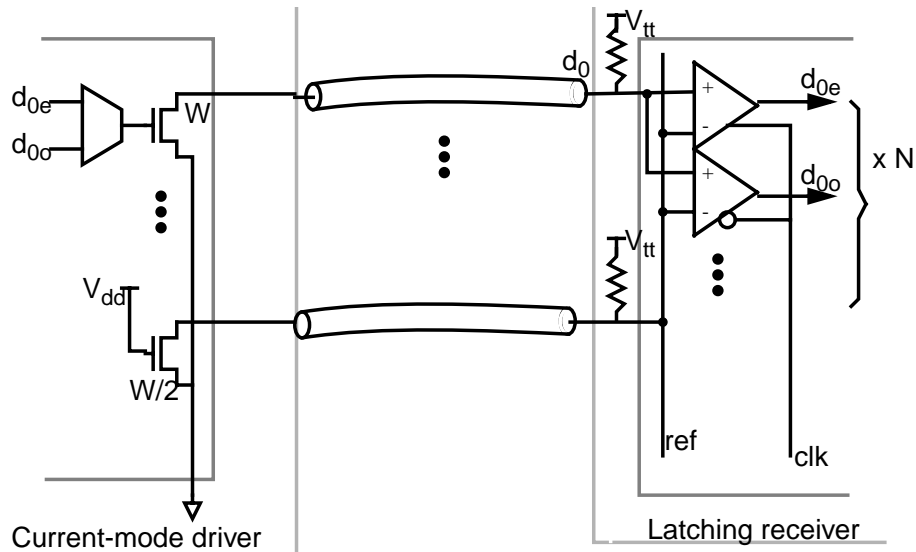
- Statically trim the offset per latch

can use digital correction (DAC)



- Active offset cancellation: connect in a feedback [8].

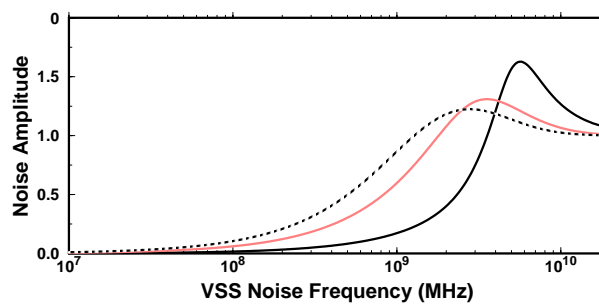
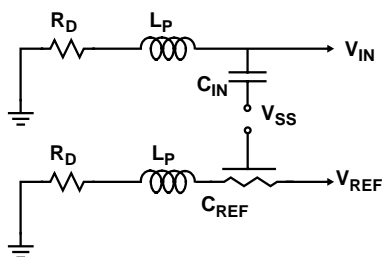
Parallel Link Example



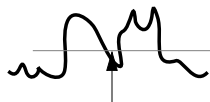
- Share the reference to save pins and wires.
- Sending reference along allows some tracking of driver side noise.
But the noise tracking is limited, especially at the receiver...

Reference Noise is Different

Reference is filtered differently from data (for multiple parallel inputs)
so noise couples differently between signal and reference.



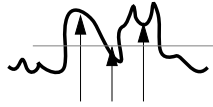
So far we only take a single sample of the data — noise can occur any time.



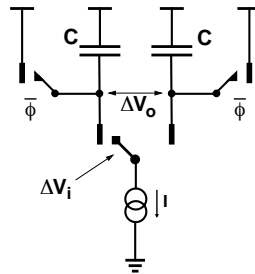
Integrating Receiver

To increase robustness:

Take multiple samples and do averaging [12]



Integrate the input data and decide at the end [5].



Noise does not affect polarity of ΔV_o .

You can amplify and latch ΔV_o with a conventional receiver afterwards.

Receiver Summary

Two types of receivers:

- amplify + latch: better offsets but bandwidth limited by amplifier

- sample + latching: no ISI but sampling noise.

Bandwidth:

- Can reach 3-4 FO-4 easily using 1:2 demultiplexing.

- More demultiplex for better bandwidth: sampling bandwidth limits to 0.5 FO-4.

Resolution:

- Static offsets: cancel with offset cancellation

- Differential to reduce noise.

- Reference noise: need to filter the input.

What about timing noise?

Transmitter and Receiver References

- [1] B. Chappel, et. al. "Fast CMOS ECL Receivers With 100 mV Sensitivity", *IEEE Journal of Solid State Circuits*, vol. 23, no. 1, Feb. 1988.
- [2] N. Kushiyama et. al., "A 500Mbyte/sec Data-Rate 4.5M DRAM," *IEEE Journal of Solid State Circuits*, vol. 28, no. 4, April 1993
- [3] A. DeHon et. al. "Automatic Impedance Control", *International Solid State Circuits Conference Digest of Technical Papers*, pp. 164-165, Feb. 1993.
- [4] S. Kim et. al. "A pseudo-synchronous skew-insensitive I/O scheme for high bandwidth memories", *IEEE Symposium on VLSI Circuits*, June 1994.
- [5] S. Sidiropoulos, M. Horowitz, "A 700 Mbps/pin CMOS Signalling Interface Using Current Integrating Receivers," *IEEE Symposium on VLSI Circuits*, Jun. 1996.
- [6] K. Donnelly et. al., "A 660Mb/s Interface Megacell Portable Circuit in 0.35um-0.7um CMOS ASIC", *International Solid State Circuits Conference Digest of Technical Papers*, pp. 290-291, Feb. 1996.
- [7] A. Yukawa, et. al. "A CMOS 8-bit high speed A/D converter IC". *1988 Proceedings of the Tenth European Solid-State Circuits Conference* p. 193-6
- [8] J.T. Wu, et. al. "A 100-MHz pipelined CMOS comparator" *IEEE Journal of Solid-State Circuits*, Jun. 1988, vol. 23, no.6, p. 1379-85

EE371

Lecture 12-49

Horowitz

- [9] B. Gunning, et. al. "A CMOS low-voltage-swing transmission-line transceiver", *1992 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb. 1992, p. 58-9
- [10] S. Sidiropoulos, et. al. "A CMOS 500 Mbps/pin synchronous point to point link interface" *Proceedings of 1994 IEEE Symposium on VLSI Circuits. Digest of Technical Papers* p. 43-4
- [11] C.K. Yang, et. al. "A 0.5- μ m CMOS 4.0-Gbps Serial Link Transceiver with Data Recovery using Oversampling", *IEEE Journal of Solid State Circuits*, May 1998, vol.33, no.5, p. 713-22
- [12] S. Kim, et. al. "An 800Mbps Multi-Channel CMOS Serial Link with 3x Oversampling," *IEEE 1995 Custom Integrated Circuits Conference Proceedings*, pp. 451, Feb. 1995.
- [13] JEDEC, "Stub Series Terminated Logic for 3.3V (SSTL_3)", EIA/JESD8-8, www.jedec.org
- [14] JEDEC, "High-speed Transceiver Logic (HSTL)", EIA/JESD8-6, www.jedec.org

EE371

Lecture 12-50

Horowitz