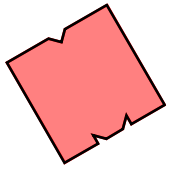
# CS-590.26, Spring 2014

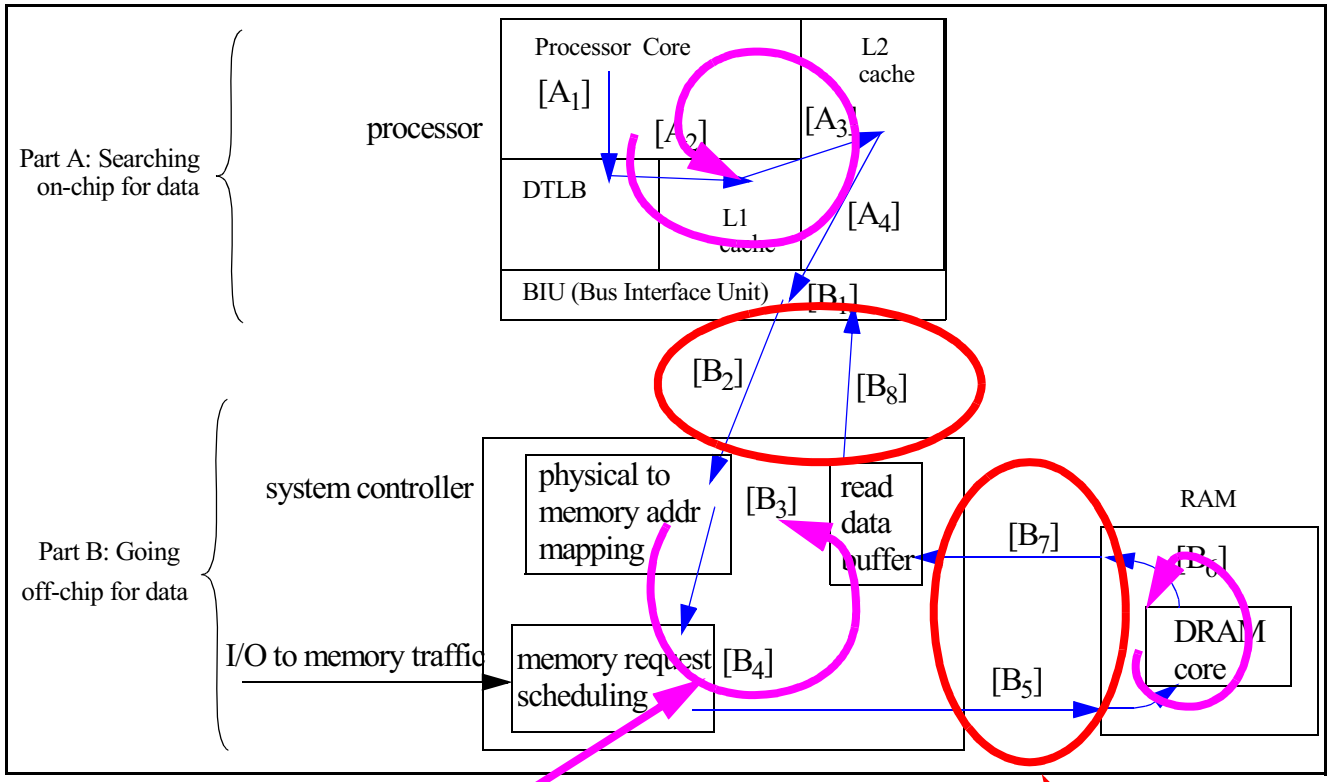
# *High Speed Memory Systems: Architecture and Performance Analysis*

## *Wires, Signalling, and Timing*



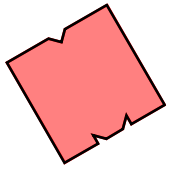


# "Digital" System Basics

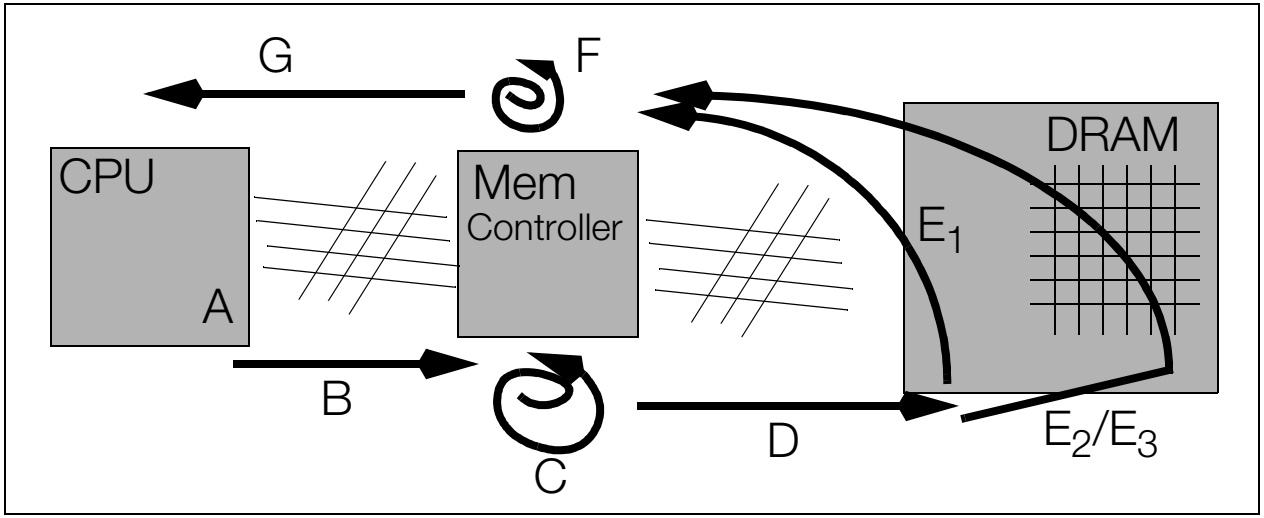



**Chip performance scaling with respect to Moore's Law**

**Data transport speed on system board does not scale w.r.t. Moore's Law.**



# How Do We Go "Faster"?




**Faster Logic**

**Faster Wires**

**Faster DRAM Circuits**

A: Transaction request may be delayed in Queue

B: Transaction request sent to Memory Controller

C: Transaction converted to Command Sequences (may be queued)

D: Command/s Sent to DRAM

$E_1$: Requires only a **CAS** or

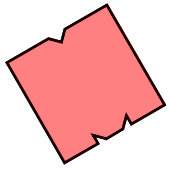
$E_2$: Requires **RAS + CAS** or

$E_3$: Requires **PRE + RAS + CAS**

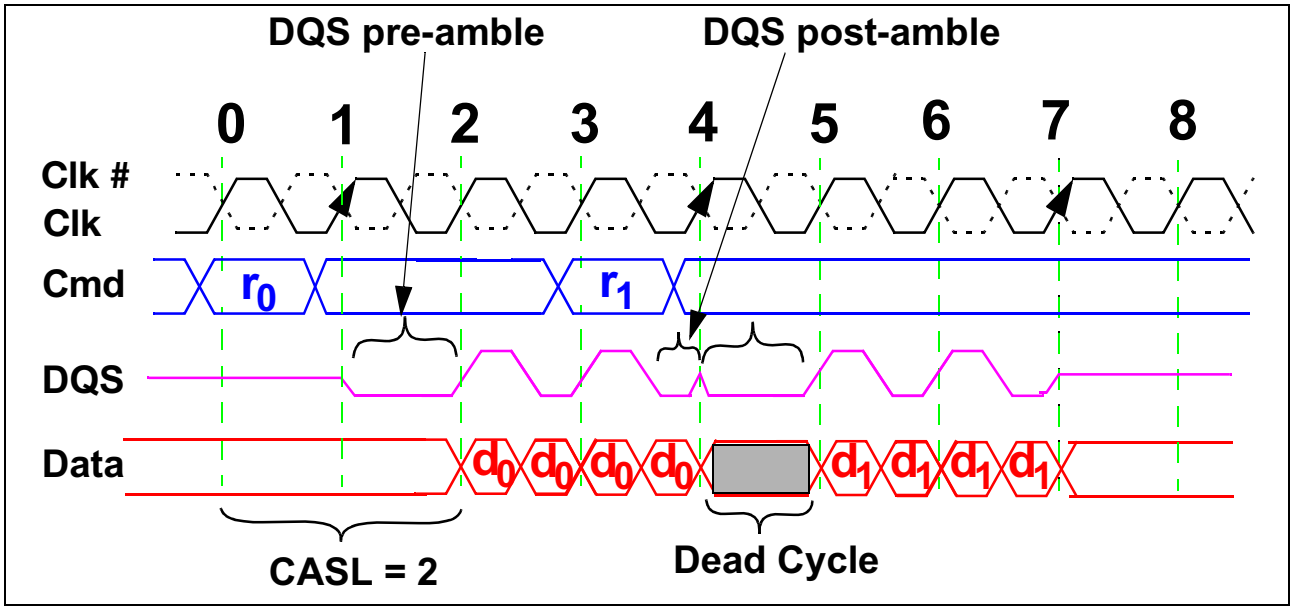
F: Data is staged at Controller

G: Transaction sent back to CPU

"DRAM Latency" = A + B + C + D + E + F + G

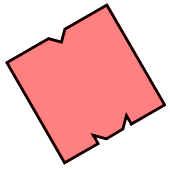


# Protocol "Efficiency"

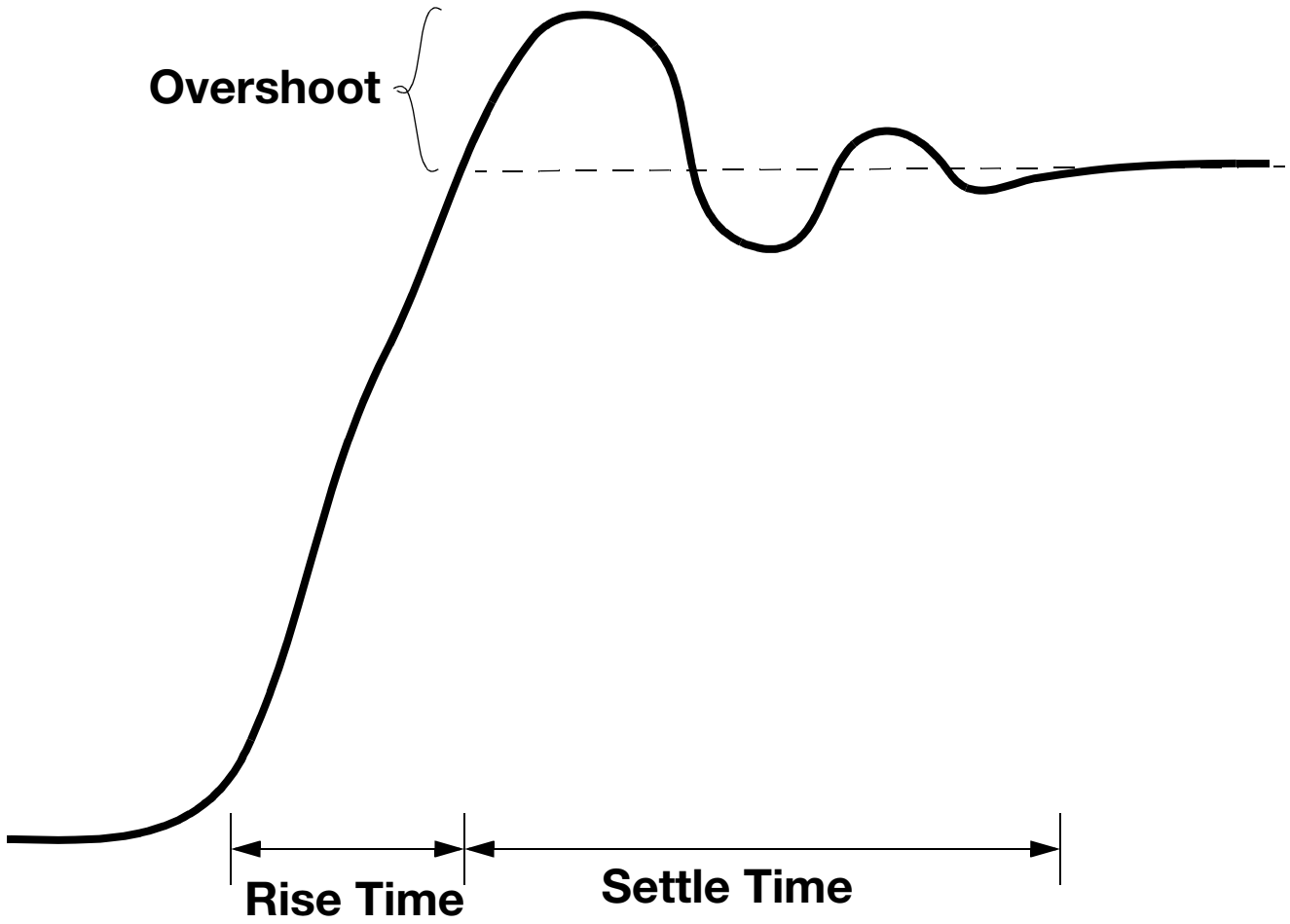


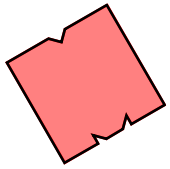

**Back-to-back reads to different ranks**

**Can we make the cycles smaller?**

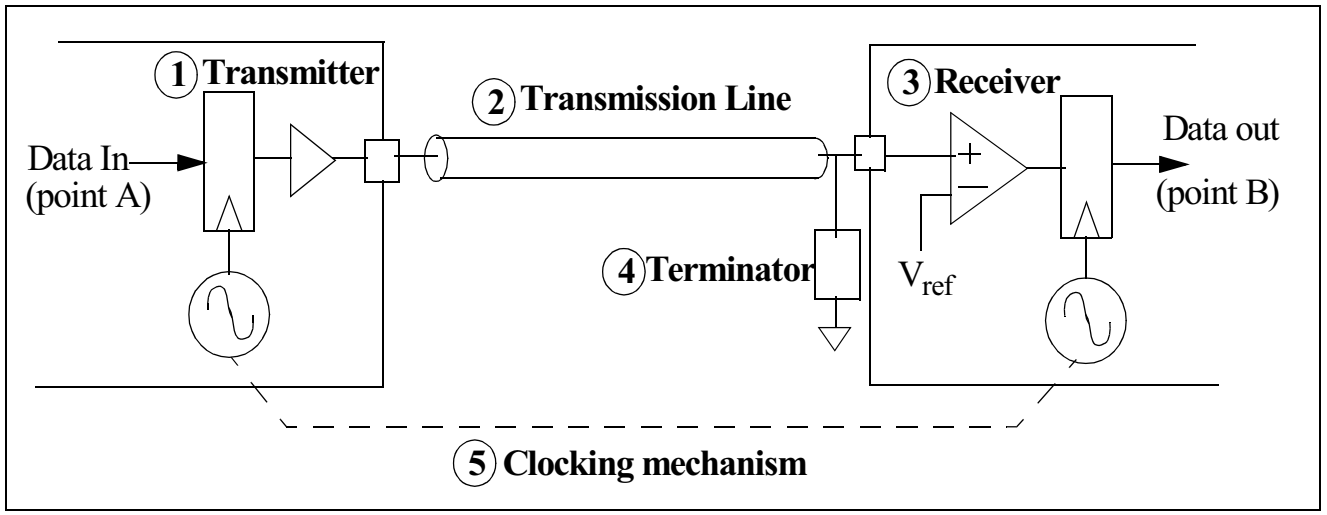


# Smaller Cycle Time Budget?



# A Signaling System




**1. Transmitter: Encodes data as current/voltage level onto the line**

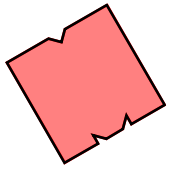
**2. Transmission Line: Deliver data from transmitter to receiver**

**3. Receiver: Compare against reference to extract data**

**4. Terminator: Remove signal from line, once they're received**

**5. Clock: Tells transmitter when to send, receiver when to sample signal**

* Poulton ISSCC 1999 Signaling Tutorial



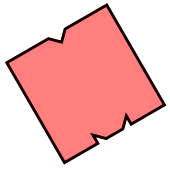
# ENEE 359a
# *Digital Electronics*

## *Interconnects*

**Prof. Bruce Jacob**

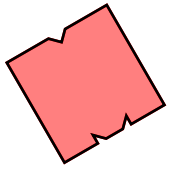
**blj@eng.umd.edu**

**Credit where credit is due**:
Slides contain original artwork (© Jacob 2004) as well as material taken liberally from Irwin & Vijay's CSE477 slides (PSU), Schmit & Strojwas's 18-322 slides (CMU), Dally's EE273 slides (Stanford), Wolf's slides for *Modern VLSI Design*, and/or Rabaey's slides (UCB).

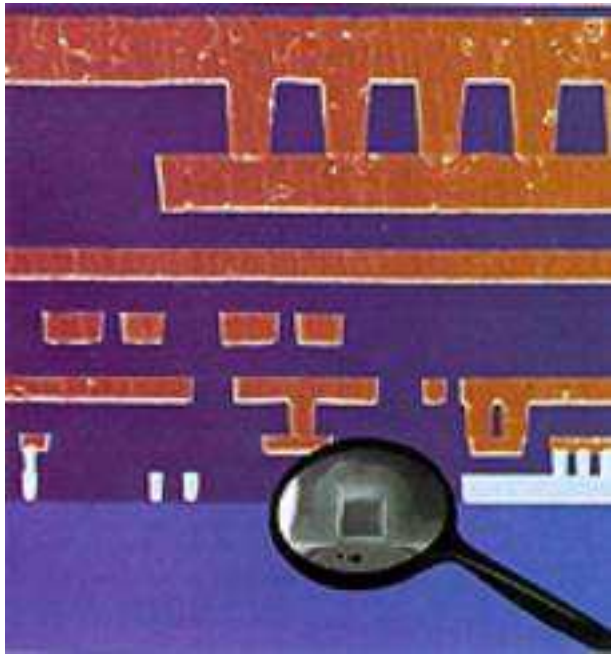


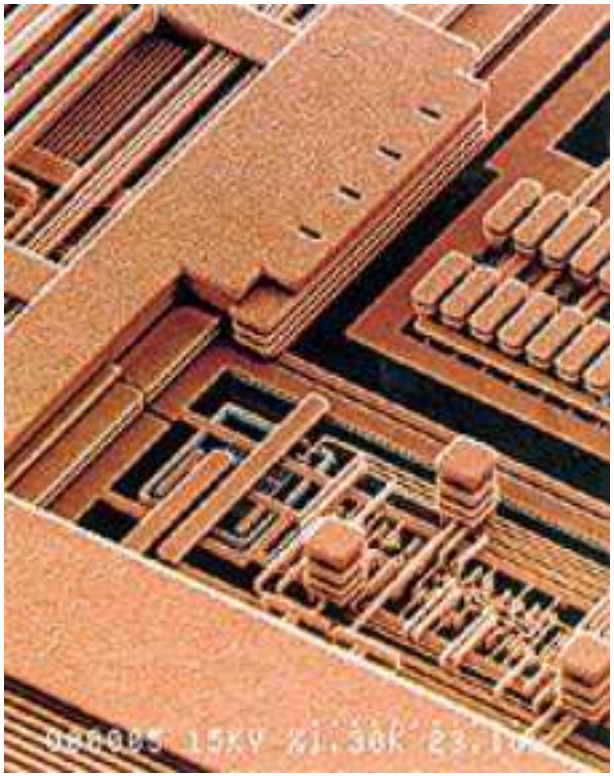
# Overview

- **Wires and their physical properties (MOSFETs, too …)**
- **LC/RC/RLC transmission lines, characteristic impedance, reflections**
- **Dynamic considerations (e.g. skin effect)**
- **The Bottom Line: propagation delay, transistor sizing, inductive (Ldi/dt) noise, capacitive coupling, signal degradation, various rules of thumb for design**



# Metal Layers in ICs





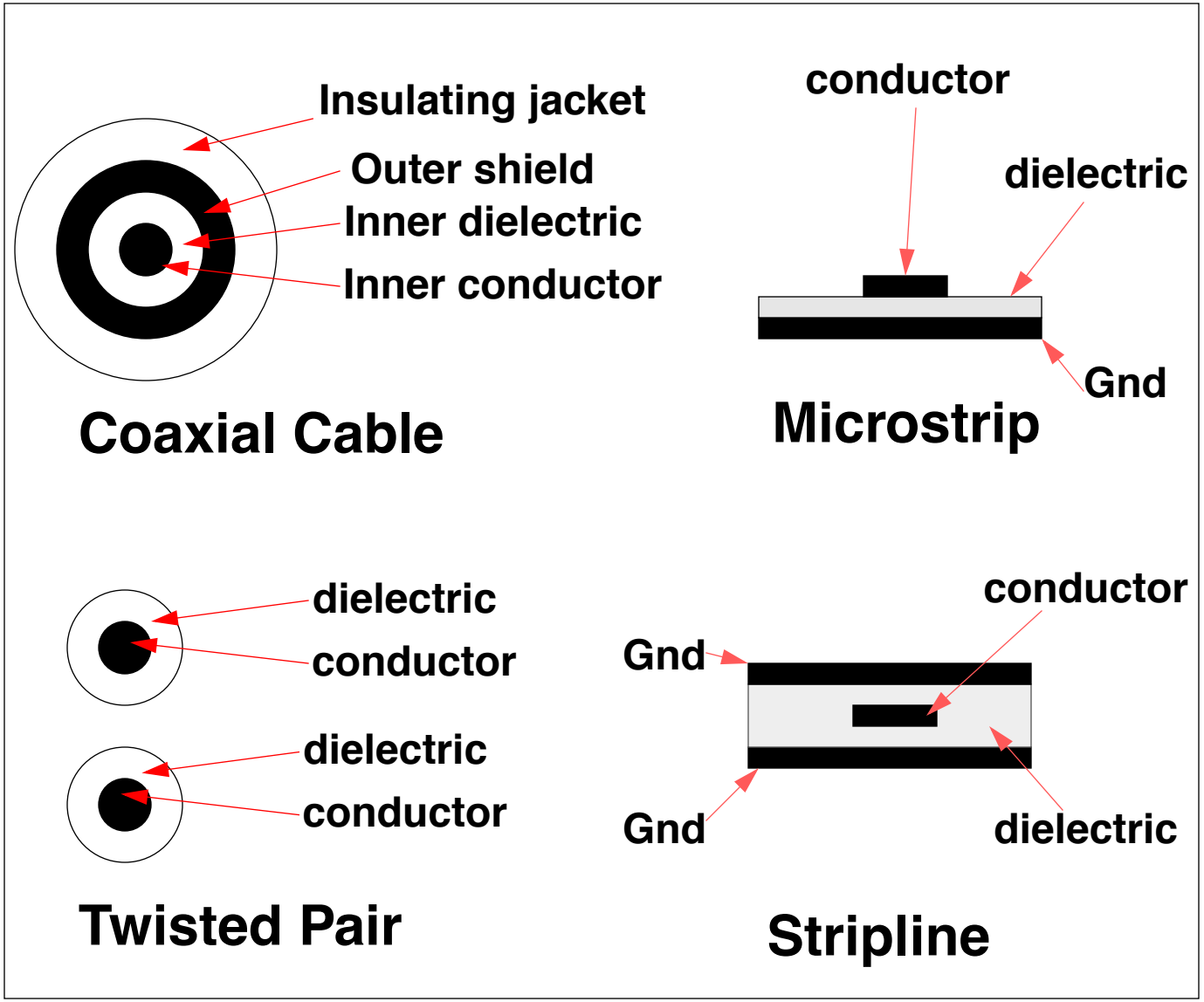
**IBM's 6-layer copper interconnect**

Bruce Jacob

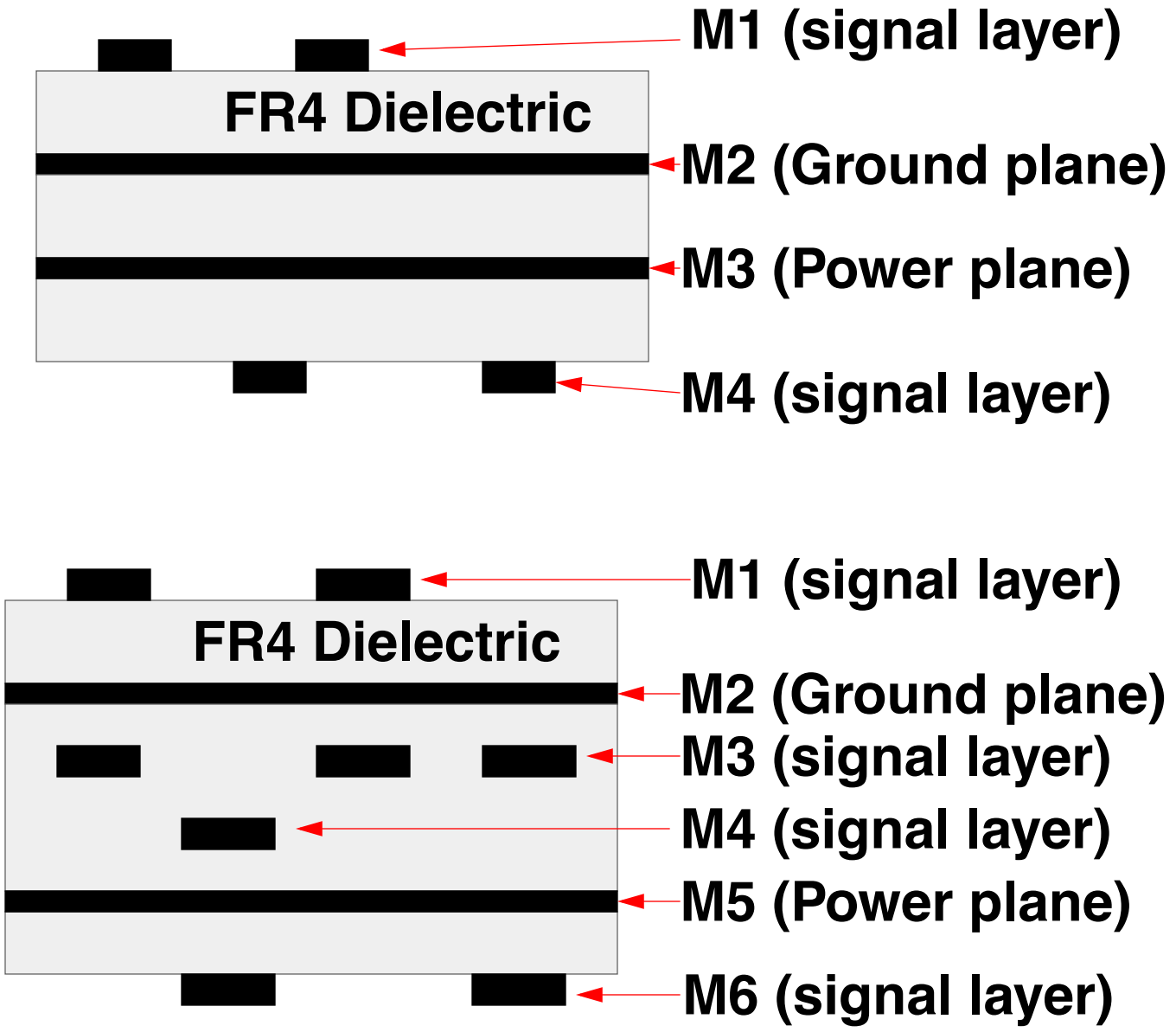
University of
Maryland
ECE Dept.



# Some Transmission Lines



# Cross Section of PCB Board



# Wires in Digital Systems

## Physically, wires are

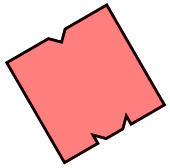
- **Stripguides on (and *in*) printed circuit-board cards, layed over & sandwiched between groundplanes**
- **Stripguides on ICs, layered atop each other**
- **Conductors in cables & cable assemblies**
- **Connectors**

## We tend to treat them as *IDEAL wires*

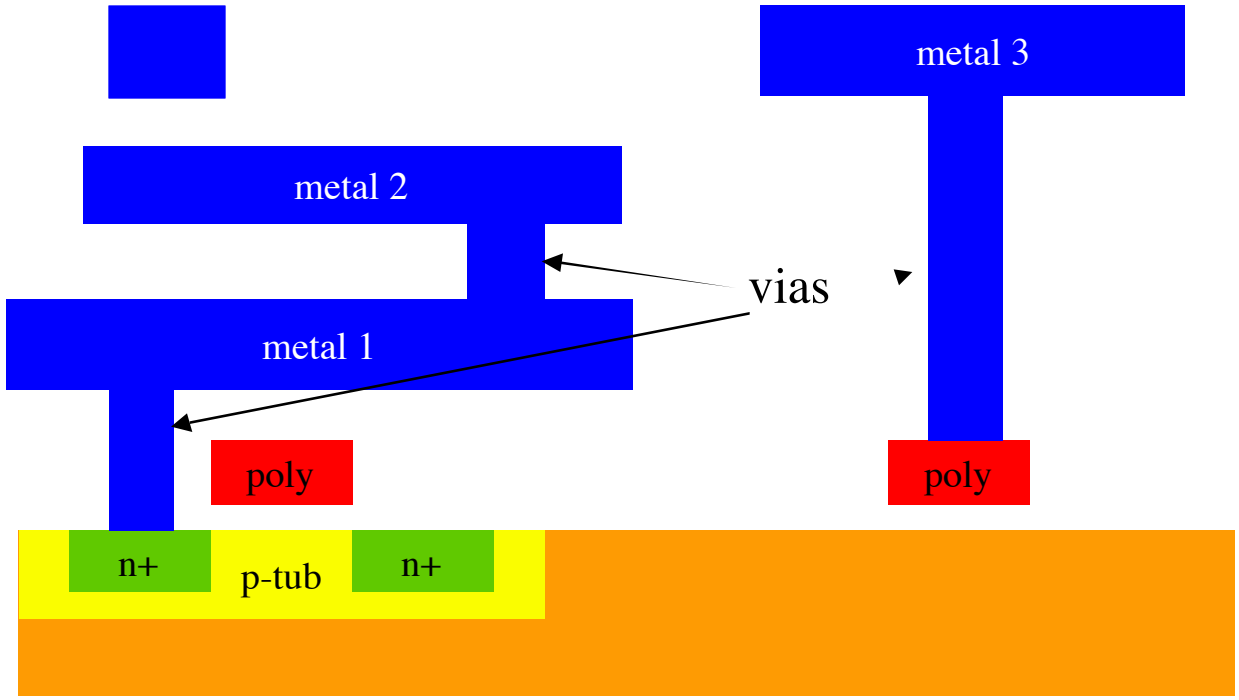
- **No delay (equipotential)**
- **No capacitance, inductance, or resistance**

## They are *NOT* ideal …

## To build reliable systems, must understand properties & behavior

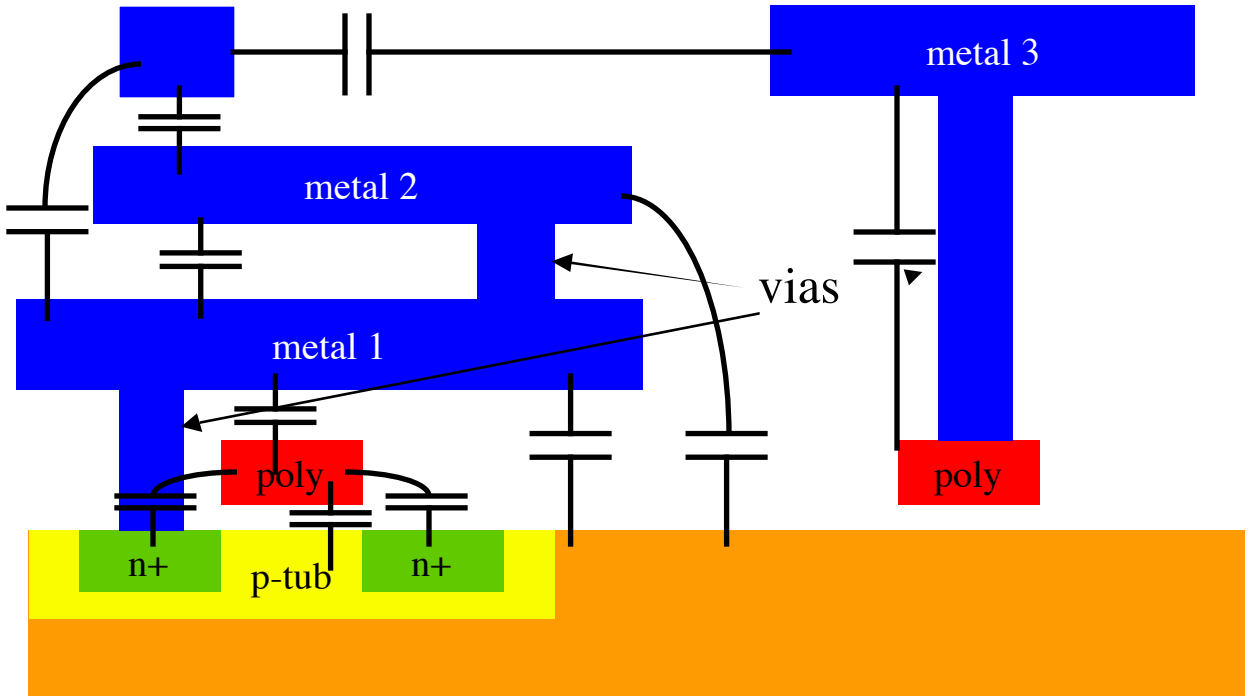


# Metal Layers in ICs



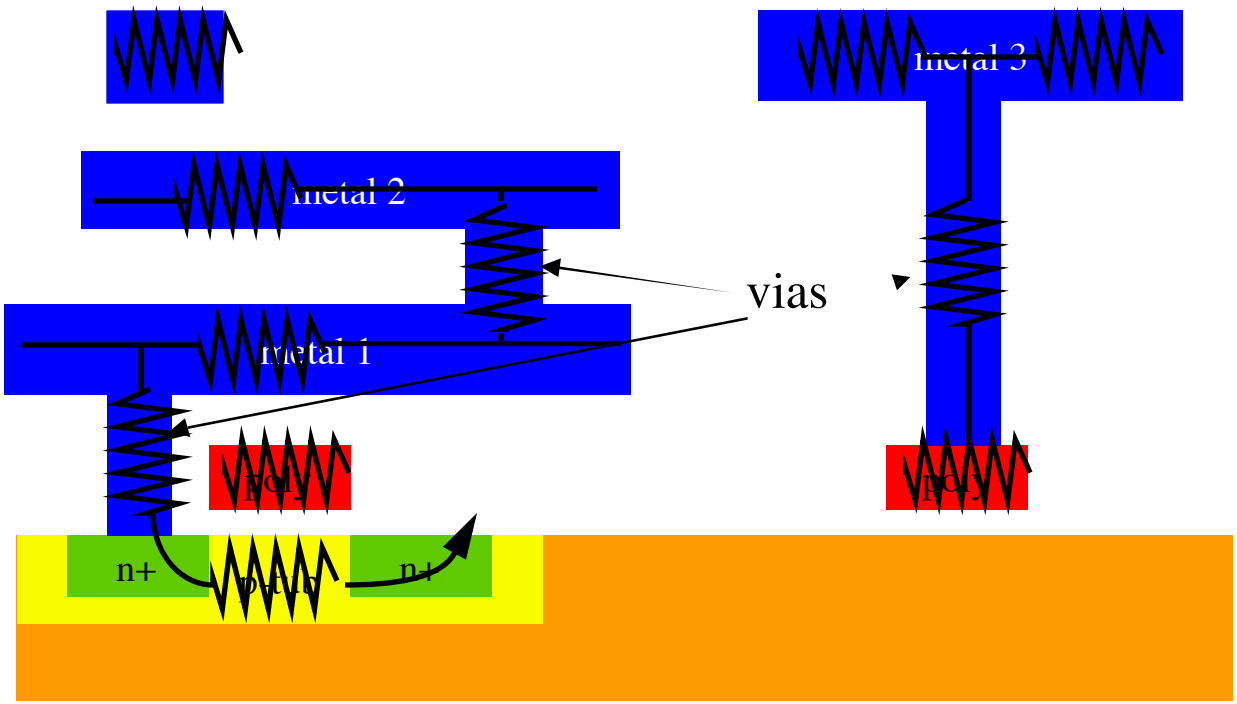

**Remember the *RC Constant* τ?**

# Metal Layers & Capacitances

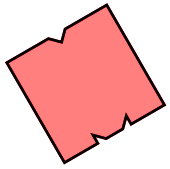



- **On-chip wires run in multiple layers with no explicit return planes (ground is used as implicit return)**
- **Thus, almost all capacitance of on-chip wire is to *other wires* (same plane, different plane, etc.)**
- **Capacitance of MOSFET scales with Vdd**

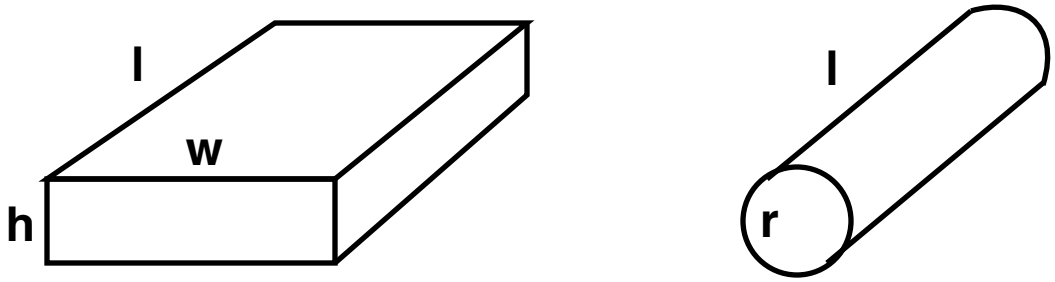
# Metal Layers & Resistances




- **Resistance of conductor proportional to length/width, depends on material (resistivity), causes delay & loss**
- **Resistance of wire scales with square root of signaling frequency (at high speeds) (“skin effect”)**
- **Process scaling tends to increase resistance**



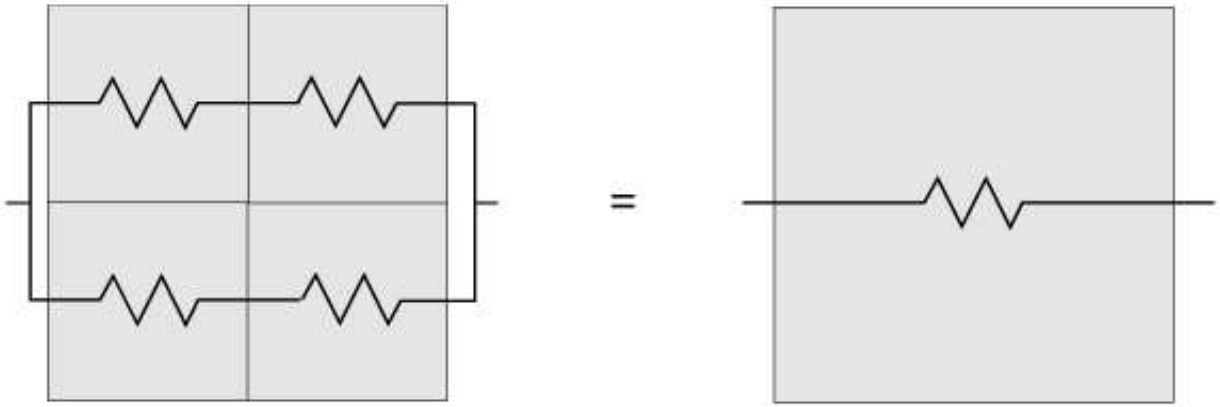
# Wire Resistance




- **$R = \rho l/A = \rho l/(wh)$ for rectangular wires (on-chip wires & vias, PCB traces)**
- **$R = \rho l/A = \rho l/(\pi r^2)$ for circular wires (off-chip, off-PCB)**

| Material | Resistivity ρ (Ω-m) |
|---|---|
| Silver (Ag) | 1.6 x 10-8 |
| Copper (Cu) | 1.7 x 10-8 |
| Gold (Au) | 2.2 x 10-8 |
| Aluminum (Al) | 2.7 x 10-8 |
| Tungsten (W) | 5.5 x 10-8 |

# Sheet Resistance



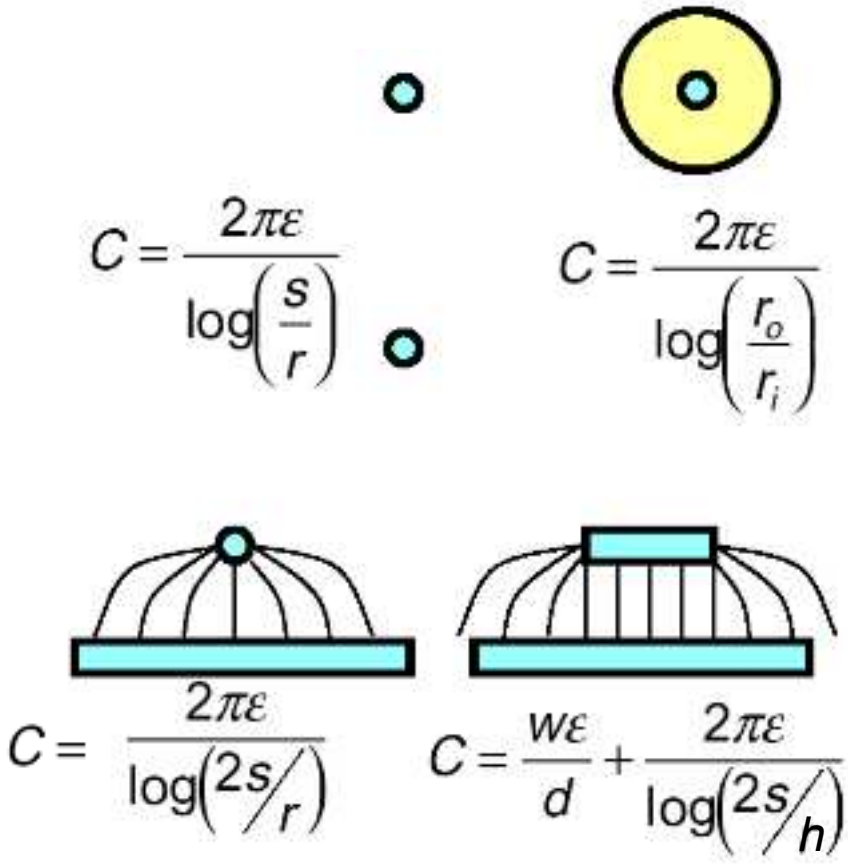
**$R = \rho l/(wh) = l/w \cdot \rho/h$ for rectangular wires**

**Sheet resistance $R_{sq} = \rho/h$**

| Material | Sheet resistance $R_{sq}$ (Ω/sq) |
|---|---|
| n, p well diffusion | 1000 to 1500 |
| n+, p+ diffusion | 50 to 150 |
| polysilicon | 150 to 200 |
| polysilicon with silicide | 4 to 5 |
| Aluminum | 0.05 to 0.1 |

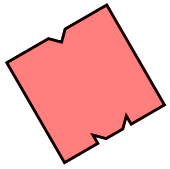
# Wire Capacitance

## Common wire cross-sections/permittivities:




| Material | $\varepsilon_r$ |
|---|---|
| Air | 1 |
| Teflon | 2 |
| Polymide | 3 |
| $SiO_2$ | 3.9 |
| Glass-epoxy (PCB) | 4 |
| Alumina | 10 |
| Silicon | 11.7 |

- **Permittivity $\varepsilon = \varepsilon_0\varepsilon_r$**
- **Permittivity of free space $\varepsilon_0 = 8.854 \times 10^{-12}$ F/m**



# Inductance

**When conductors of transmission line are surrounded by uniform dielectric, capacitance & inductance are related:**

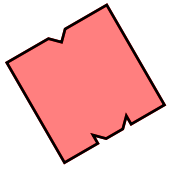
$$CL = \varepsilon\mu$$

**Inductive effects can be ignored**

- **if the resistance of the wire is substantial enough (as is the case for long Al wires with small cross section)**
- **if rise & fall times of applied signals are slow enough**

**So … inductance must be considered**

- **for off-chip signals (even power/ground)**
- **for future even-higher-speed on-chip signalling**



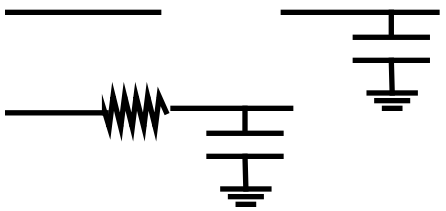
# Wires & Models

## Example Wires:

| Type | W | R | C | L |
|---|---|---|---|---|
| On-chip | 0.6 $\mu$m | 150k $\Omega$/m | 200 pf/m | 600 nH/m |
| PC Board | 150 $\mu$m | 5 $\Omega$/m | 100 pf/m | 300 nH/m |
| 24AWG pair | 511 $\mu$m | 0.08 $\Omega$/m | 40 pf/m | 400 nH/m |

## In a situation, use a *model* of wires that captures the properties we need:

- ideal, lumped L, R, or C
- LC, RC, RLC transmission line
- General LRCG transmission line



## Appropriate choice of model depends on signaling frequency

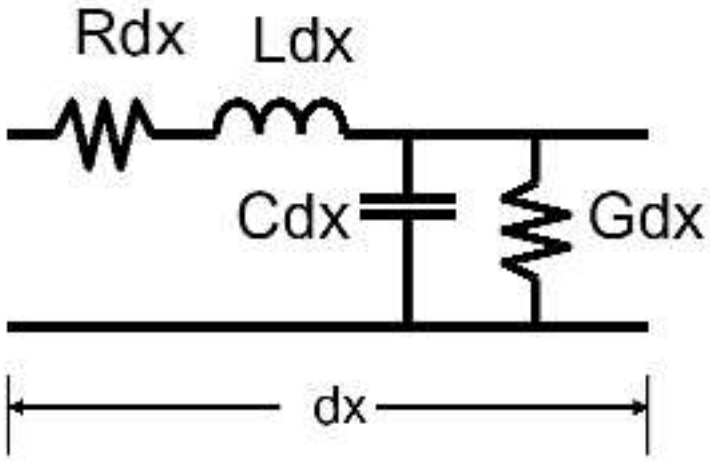
$$f_0 = \frac{R}{2\pi L}$$

# General LRCG Model

**Model an *infinitesimal* length of wire, dx, with lumped components**

**L, R, C, and G**

**(inductance, resistance, capacitance, and conductance)**




**Drop across R and L**

$$\frac{\partial V}{\partial x} = RI + L\frac{\partial I}{\partial t}$$

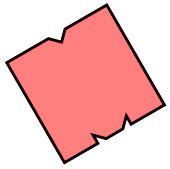
**Current into C and G**

$$\frac{\partial I}{\partial x} = GV + C\frac{\partial V}{\partial t}$$

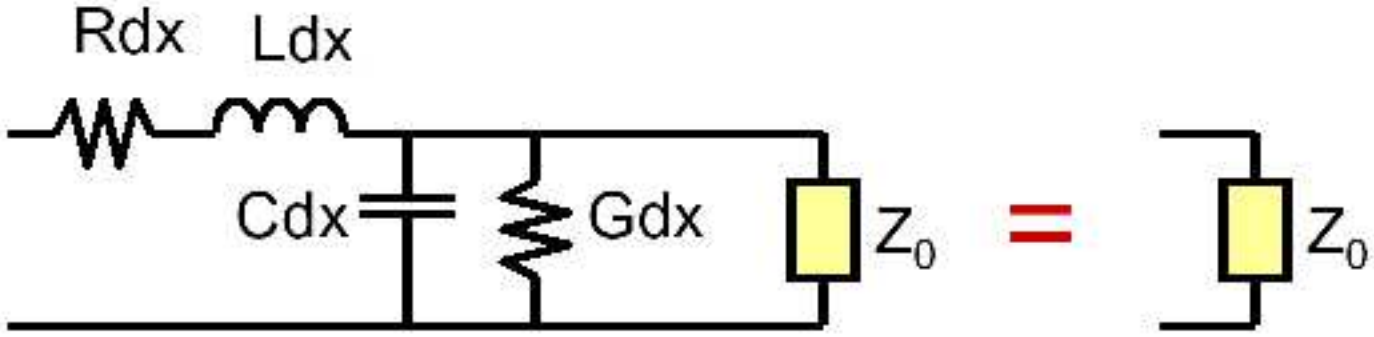
$$\frac{\partial^2 V}{\partial x^2} = RGV + (RC + LG)\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$

**For G=0:**

$$\frac{\partial^2 V}{\partial x^2} = RC\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$



# Impedance

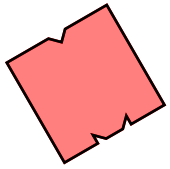



$$Z_0 = \left( \frac{R + Ls}{G + Cs} \right)^{\frac{1}{2}} \qquad s = 2\pi jf = j\omega \qquad Z_0 = \left( \frac{L}{C} \right)^{\frac{1}{2}}$$

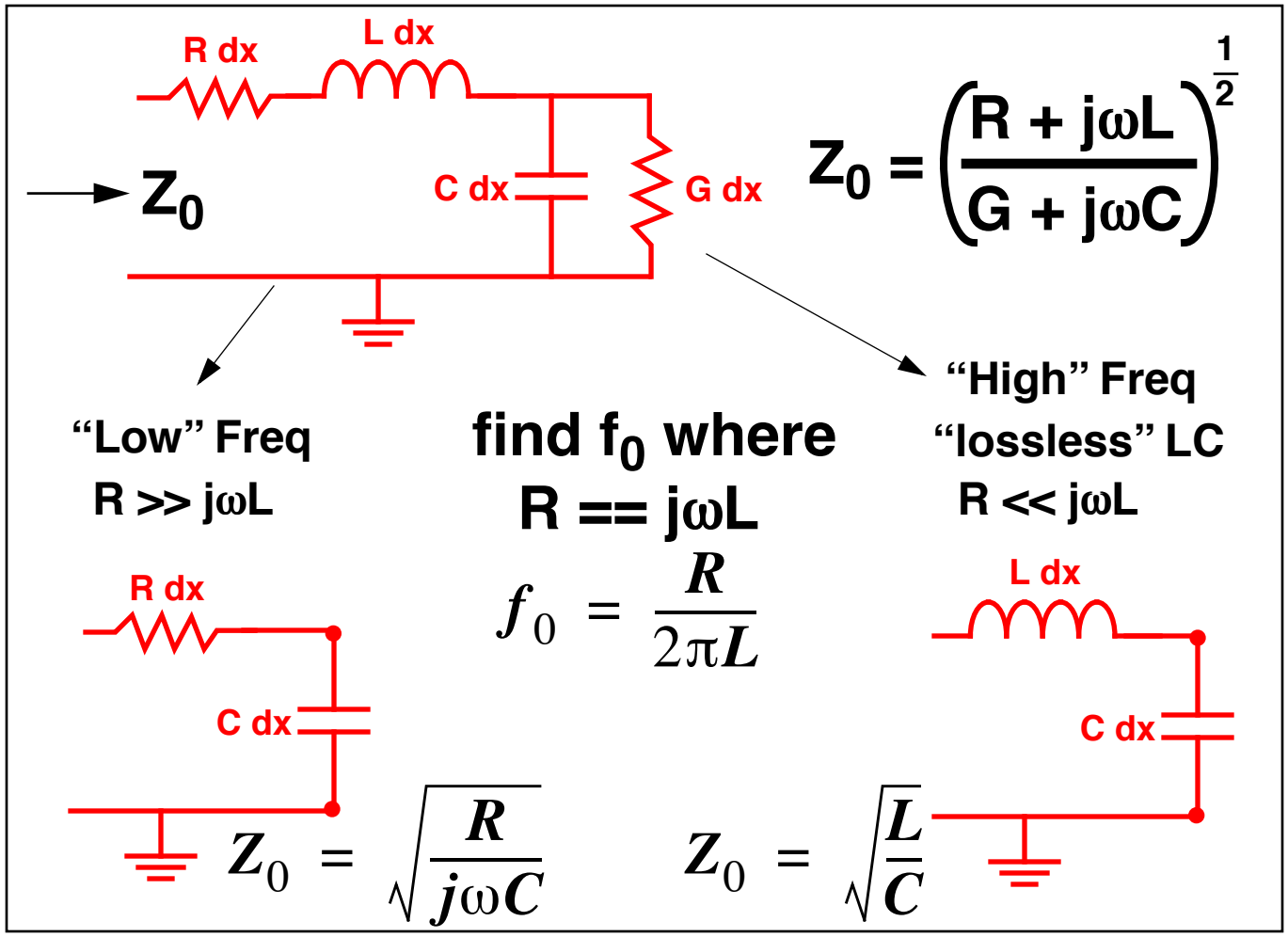
At high frequency (LC lines)

***(typical assumption: G = 0)***

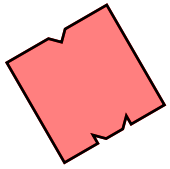
- **An infinite length of LRCG transmission line has *impedance $Z_0$***
- **Driving a line *terminated* into $Z_0$ is same as driving $Z_0$**
- **In general, $Z_0$ is complex and frequency-dependent**
- **For LC lines (operating at "high" frequencies), $Z_0$ is real-valued and independent of frequency**



# Cut-off Frequency $f_0$




- Transmission lines have characteristic frequency $f_0$
- Below $f_0$ ≈ RC model, Above $f_0$ ≈ LC model

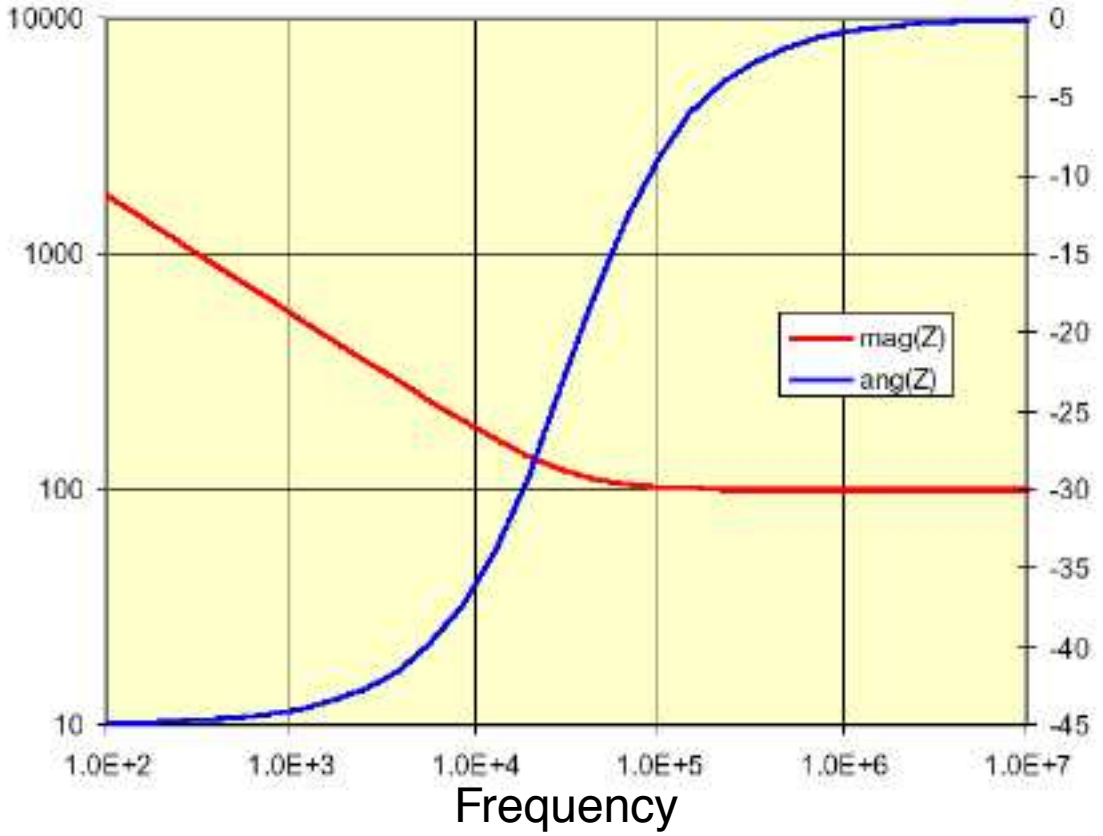


# Cut-off Frequency $f_0$

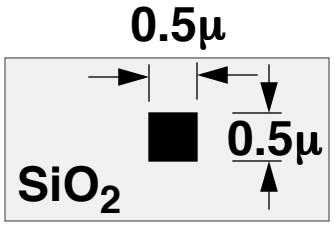
## Example, 24AWG Pair

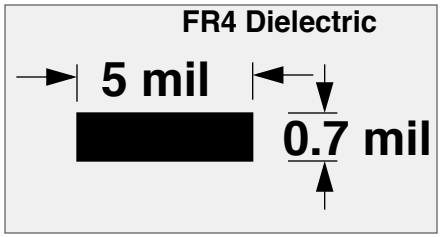
- $f_0$ = 33kHz
- Below $f_0$, line is RC
- Above $f_0$, line is LC

$$Z_0 = \left( \frac{.08 + 400 \times 10^{-9} \times 2\pi f j}{40 \times 10^{-9} \times 2\pi f j} \right)^{\frac{1}{2}}$$



# Cut-off Frequency $f_0$ II




L = 0.6 nH/mm
C = 73 nF/mm
$R_{dc}$ = 120Ω /mm
$f_0$ = 32 GHz

L = 0.5 nH/mm
C = 104 fF/mm
$R_{dc}$ = 0.008Ω /mm
$f_0$ = 2.5 MHz

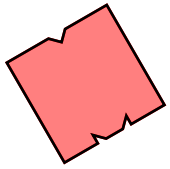
~RC Model for on chip interconnects

~LC Model for PC Board traces

$$Z_0 = \left(\frac{L}{C}\right)^{\frac{1}{2}} = \left(\frac{0.5\ \text{nH}}{0.1\ \text{pF}}\right)^{\frac{1}{2}} \sim= 70\ \Omega$$

1 mil = 0.001 inch

Example from Poulton 1999 ISSCC Tutorial



# RC Lines (low frequency)

$$\frac{\partial^2 V}{\partial x^2} = RGV + (RC + LG)\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$

## R >> jωL, governed by diffusion equation:

$$\frac{\partial^2 V}{\partial x^2} = RC\frac{\partial V}{\partial t}$$

## Signal diffuses down line, disperses:

R increases w/ length d

C increases with d

Delay & rise time both increase with RC, thus with $d^2$

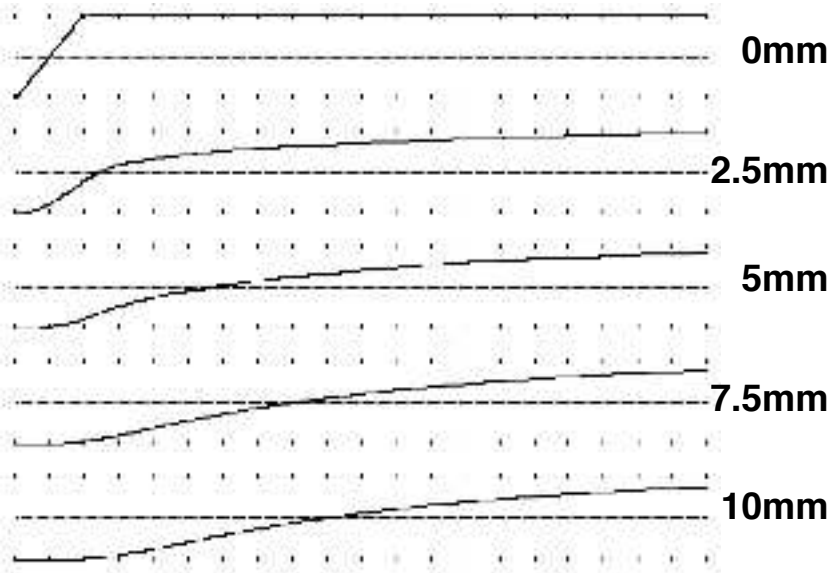
For a typical wire:

R = 150KΩ/m

C = 200pF/m

$\tau = RC = 30\ \mu s/m^2$

$= 30\ ps/mm^2$



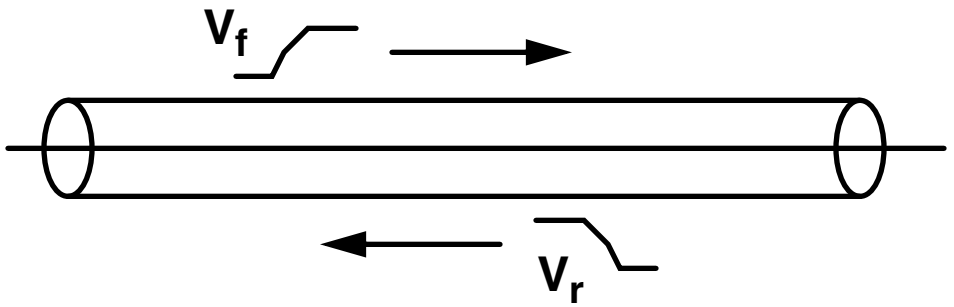
# LC Lines (high frequency)

$$\frac{\partial^2 V}{\partial x^2} = RGV + (RC + LG)\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$

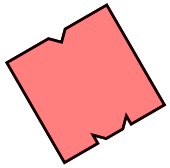
**R << jωL, governed by wave equation:**

$$\frac{\partial^2 V}{\partial x^2} = LC\frac{\partial^2 V}{\partial t^2} \qquad V_i(x, t) = \left(\frac{Z_0}{Z_0 + R_S}\right) V_S\left(t - \frac{x}{v}\right)$$

**Waveform on line is superposition of forward- and reverse-traveling waves:**




- **Waves travel with velocity v = $(LC)^{-1/2}$**
- **What happens when the wave gets to the end of line?**



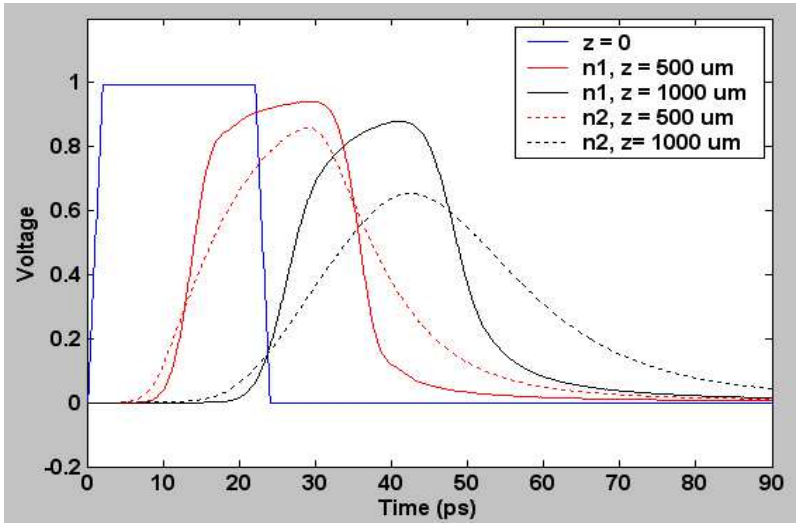
# RLC/G Lines (general case)

$$\frac{\partial^2 V}{\partial x^2} = RGV + (RC + LG)\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$

## Ignoring G, wave propagation equation:

$$\frac{\partial^2 V}{\partial x^2} = RC\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2}$$

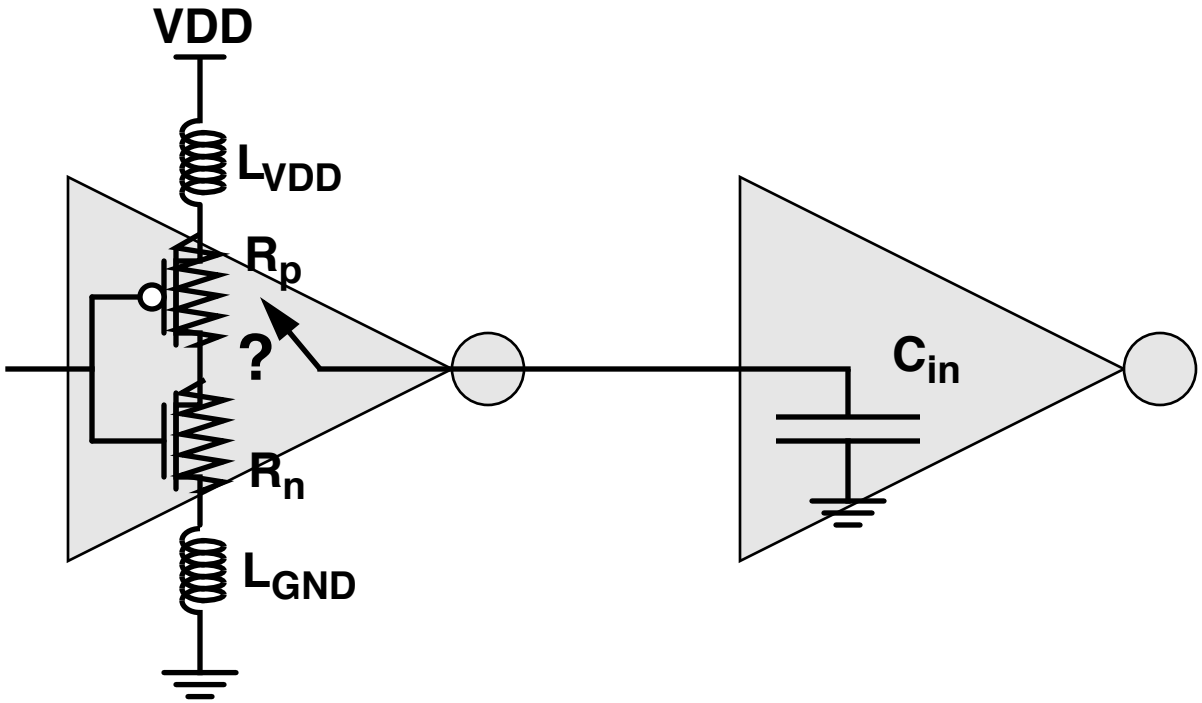
## Lossy transmission line, dispersive waves:




Substrate-doping-dependent dispersion of a picosecond-scale pulse in propagation of an on-chip transmission line

# RC vs. RLC

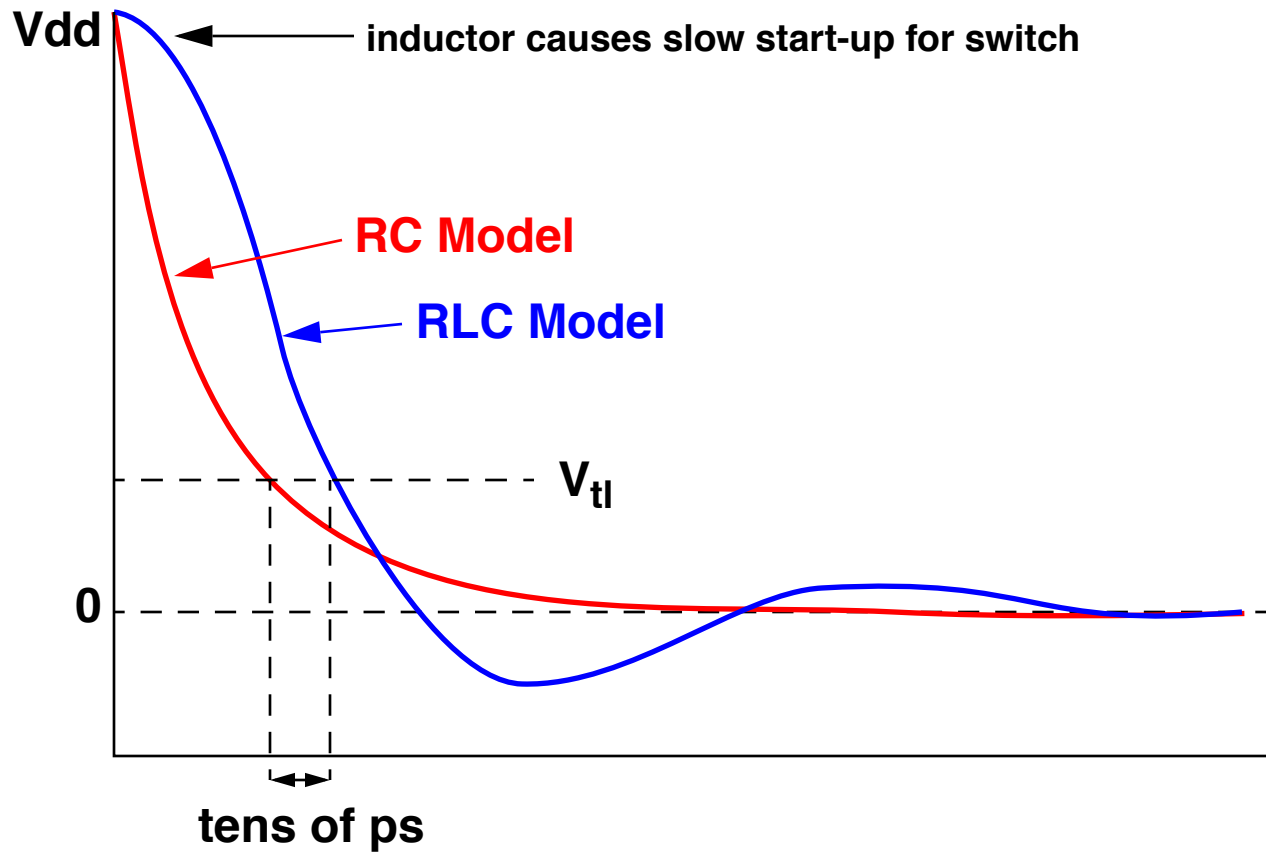
## Output response of inverter with step input:




**In reality, we have a non-zero inductance in series with the RC circuit. (Inductors and capacitors both "have memory")**

# RC vs. RLC

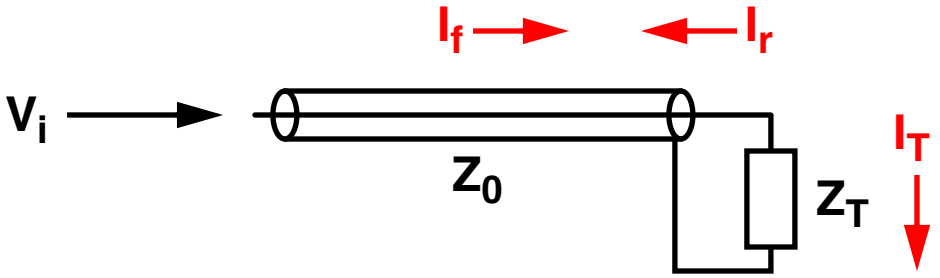
## Output response of inverter with step input:




## Result: slower reponse time, ringing

# Impedance and Reflections

## Terminating a Transmission Line:




## Telegrapher's Equations:

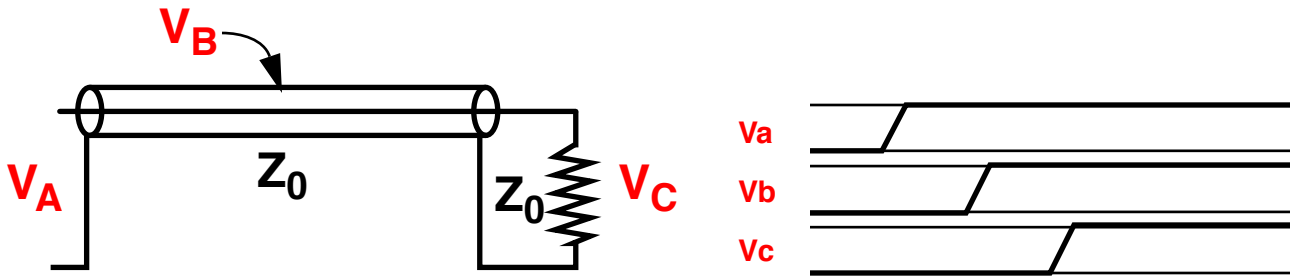
$$k_r = \frac{I_r}{I_i} = \frac{V_r}{V_i} = \frac{Z_T - Z_0}{Z_T + Z_0}$$

**Reflection coefficient $k_r$ may be complex for complex impedances $Z_T$ — i.e., the reflected wave may be phase-shifted from the incident wave.**

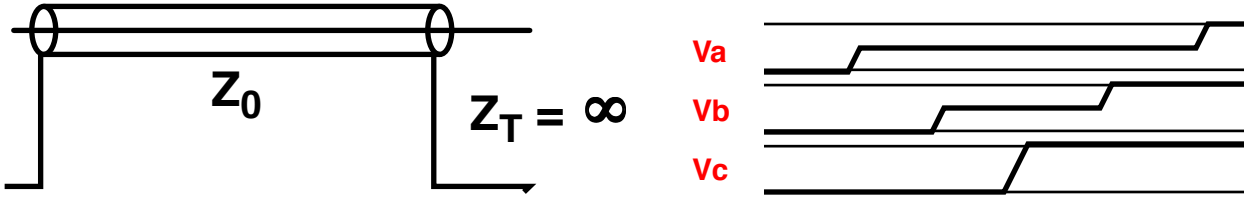
**For real-valued $Z_T$ the reflection coefficient is real, and the phase shift is either 0 ($k_r$ positive) or $\pi$ ($k_r$ negative).**

# Impedance and Reflections

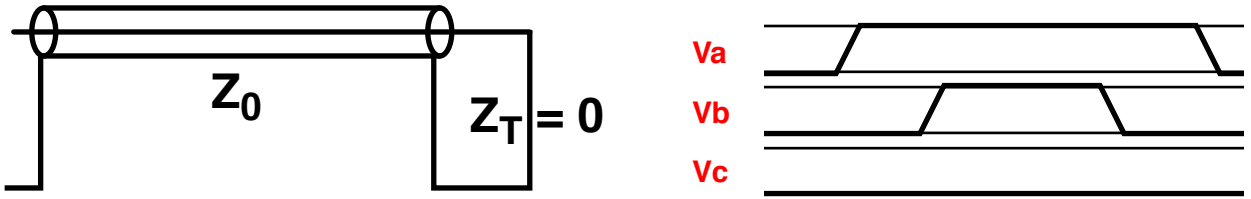
$$k_r = \frac{I_r}{I_i} = \frac{V_r}{V_i} = \frac{Z_T - Z_0}{Z_T + Z_0}$$



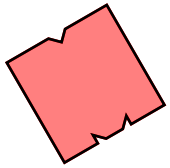

Matched Termination, $k_r = 0$



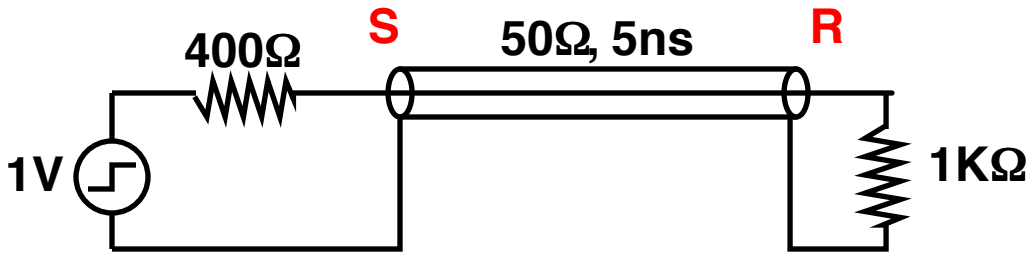

Open-Circuit Termination, $k_r = 1$



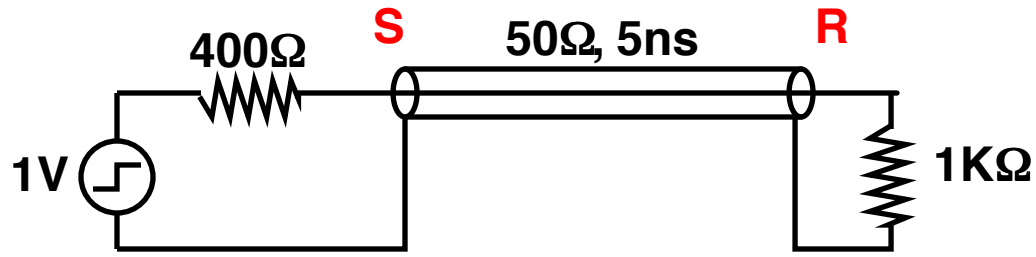

Short-Circuit Termination, $k_r = -1$



# Impedance and Reflections



# Impedance and Reflections

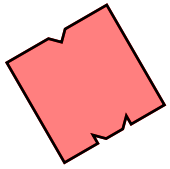



$$V_i = \left(\frac{Z_0}{Z_0 + R_S}\right) V_S = \left(\frac{50}{50 + 400}\right) 1V = 0.111V$$

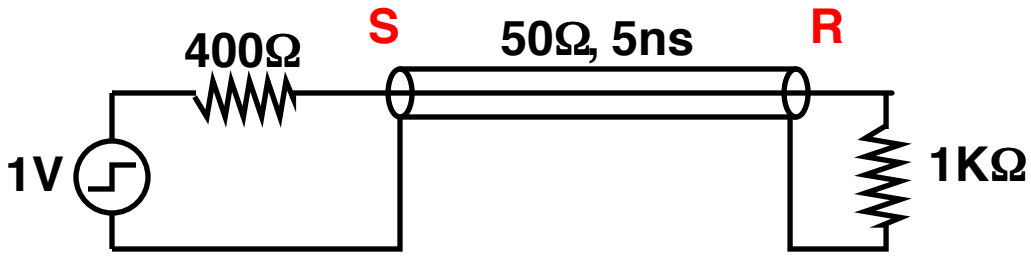
$$k_{rR} = \frac{Z_T - Z_0}{Z_T + Z_0} = \frac{1000 - 50}{1000 + 50} = 0.905$$

$$k_{rS} = \frac{Z_T - Z_0}{Z_T + Z_0} = \frac{400 - 50}{400 + 50} = 0.778$$

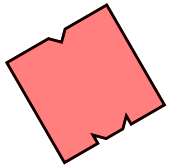
**Values are typical for 8-mA CMOS driver with 1kΩ pullup**



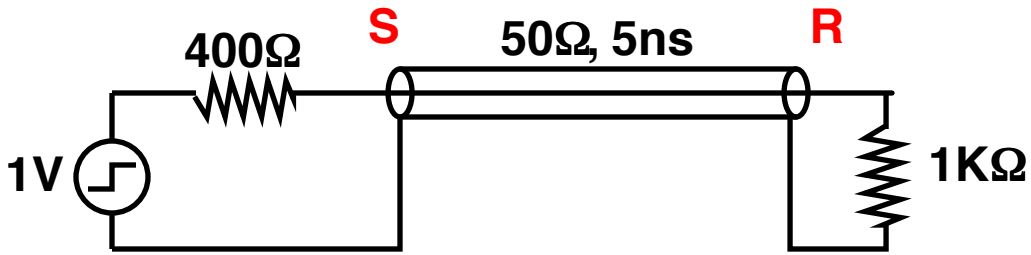
# Impedance and Reflections

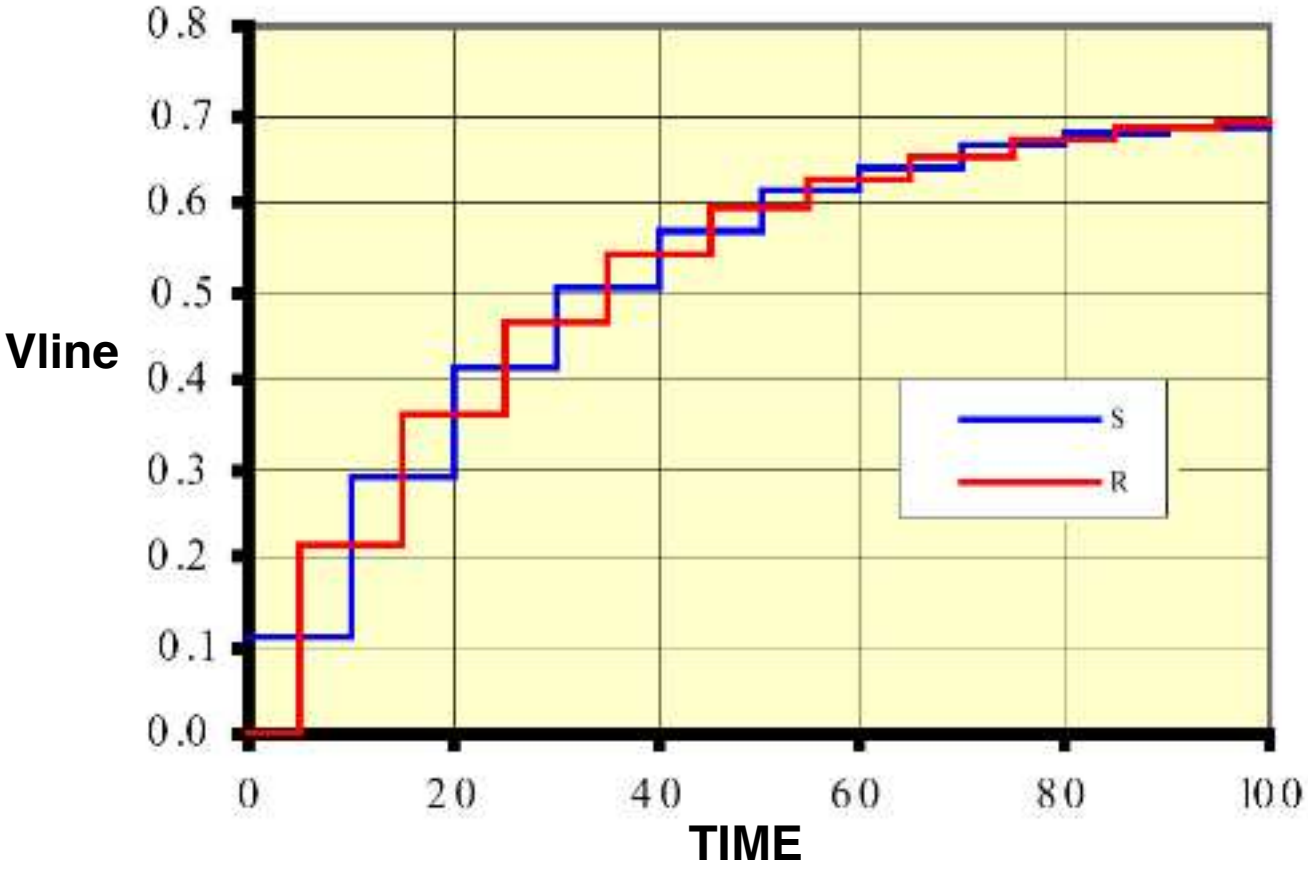



| | Vwave | Vline | time t |
|---|---|---|---|
| Vi1 | 0.111 | 0.111 | 0 |
| Vr1 | 0.101 | 0.212 | 5 |
| Vi2 | 0.078 | 0.290 | 10 |
| Vr2 | 0.071 | 0.361 | 15 |
| Vi3 | 0.055 | 0.416 | 20 |
| Vr3 | 0.050 | 0.465 | 25 |
| Vi4 | 0.039 | 0.504 | 30 |
| Vr4 | 0.035 | 0.539 | 35 |
| Vi5 | 0.027 | 0.566 | 40 |

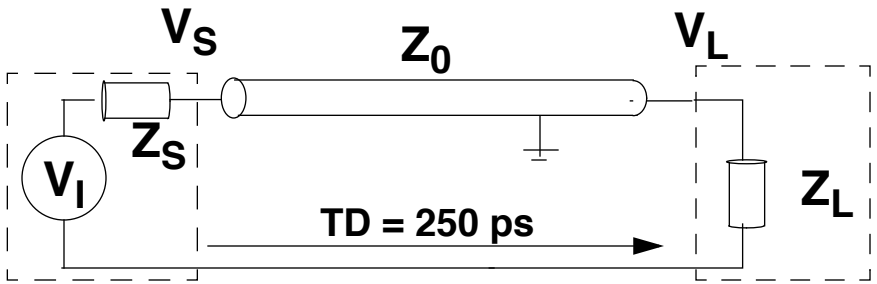


# Impedance and Reflections



# Reflections, $Z_S < Z_0$




$Z_S = 25\ \Omega$

$Z_0 = 50\ \Omega$

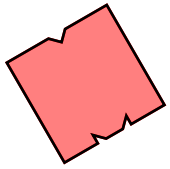
$Z_L = \text{inf}\ \Omega$

$V_I = 0\text{v} \rightarrow 2\text{v}$

$$V_S = V_I \frac{Z_S}{Z_S + Z_0} = 2 * \frac{50}{25 + 50} = 1.3333\ \text{V}$$

$$k_{r\ (\text{load})} = \frac{Z_L - Z_0}{Z_L + Z_0} = \frac{\text{inf} - 50}{\text{inf} + 50} = 1$$

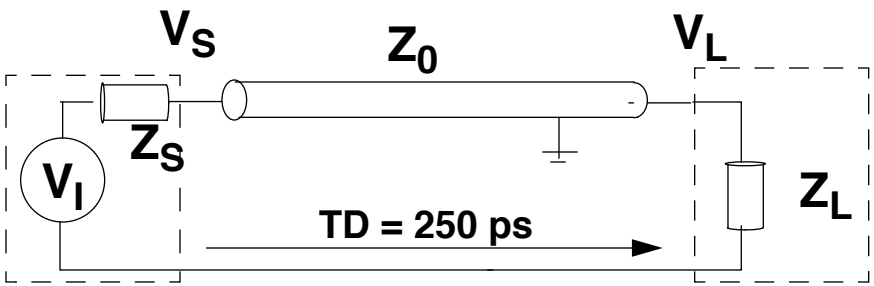
$$k_{r\ (\text{source})} = \frac{Z_S - Z_0}{Z_S + Z_0} = \frac{25 - 50}{25 + 50} = -0.3333$$

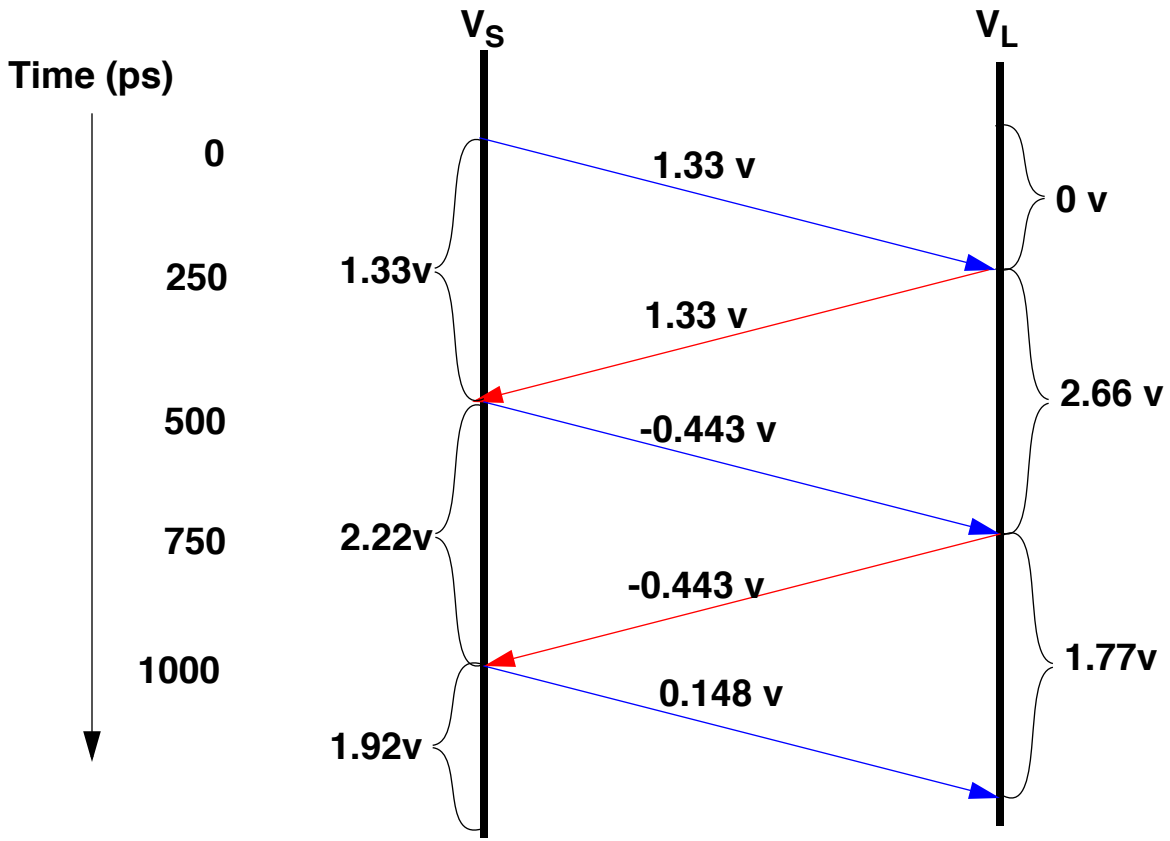


# Reflections, $Z_S < Z_0$

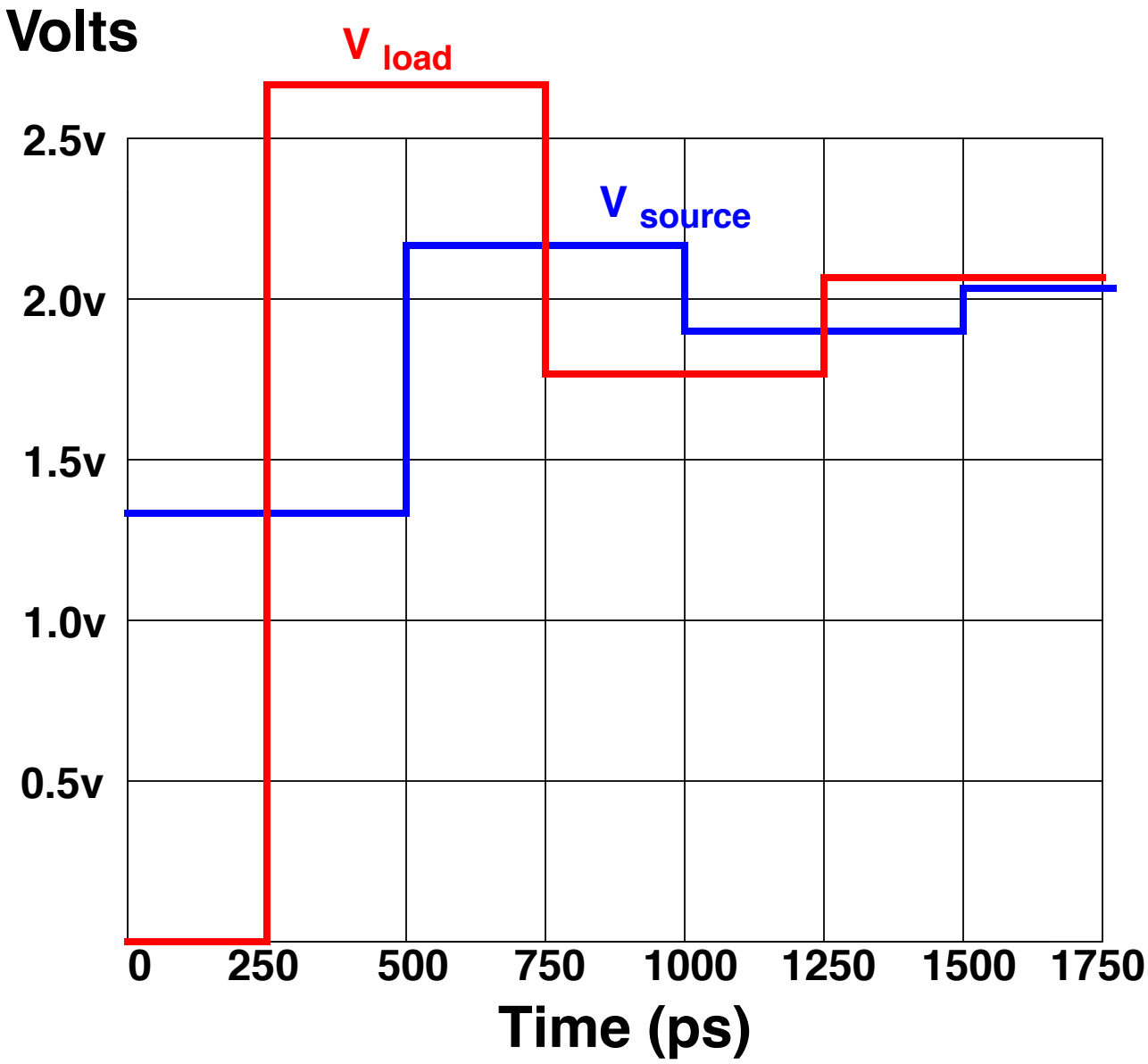
$k_{r\ (load)} = 1$

$k_{r\ (source)} = -0.3333$

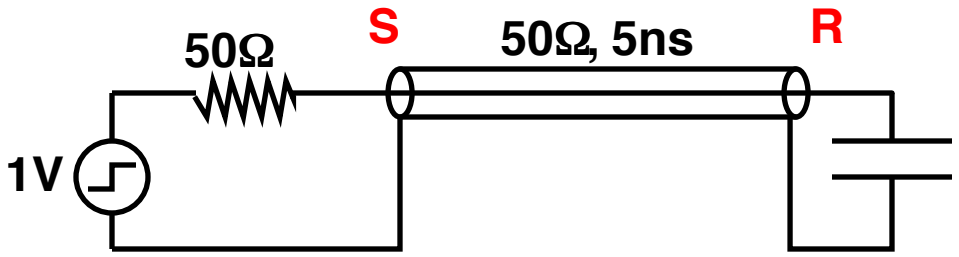


# Reflections, $Z_S < Z_0$

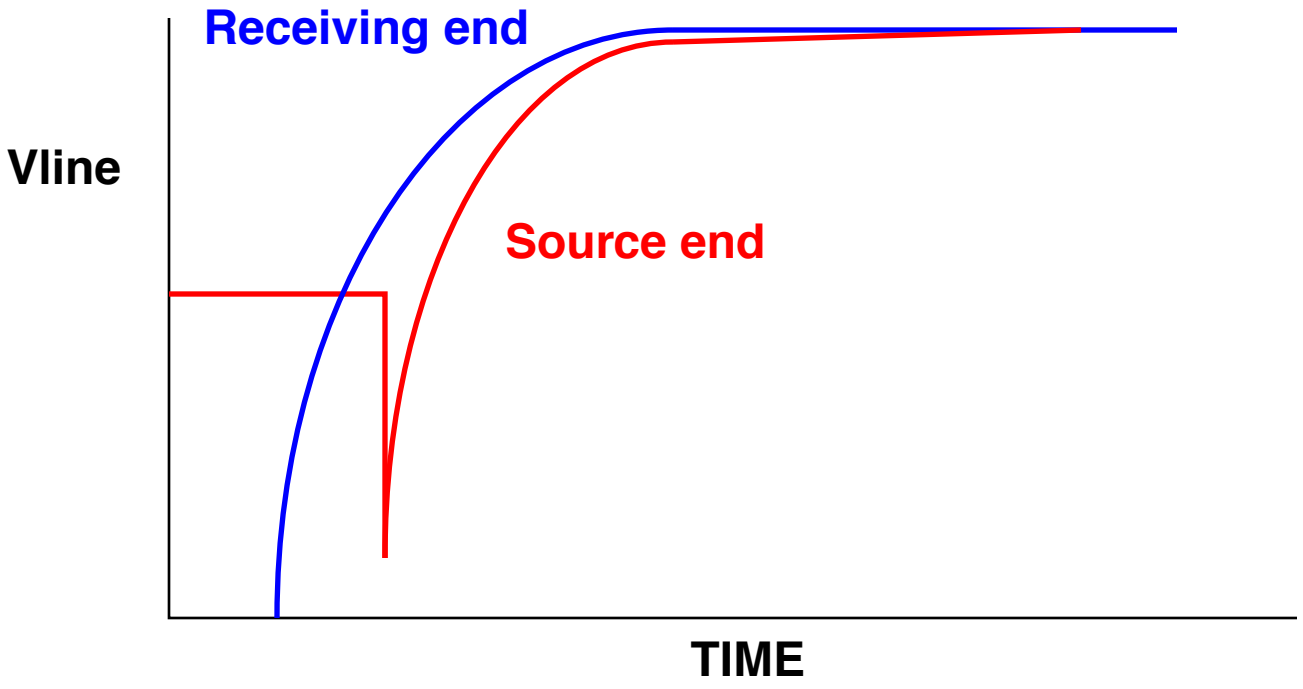


# Add In Capacitance …

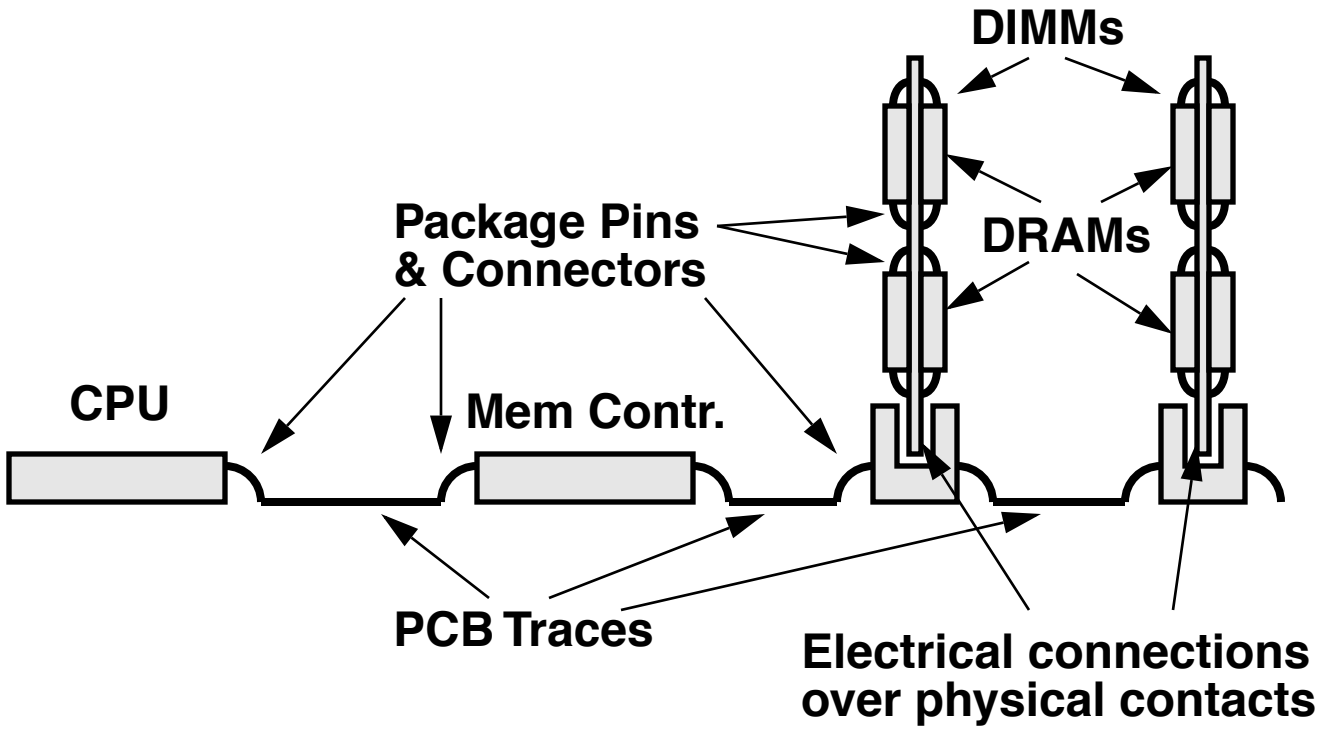



**What if we throw in a capacitor (i.e., reality?)**

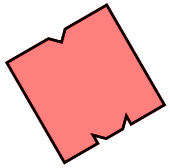
**Simple case: matched impedance at source end**



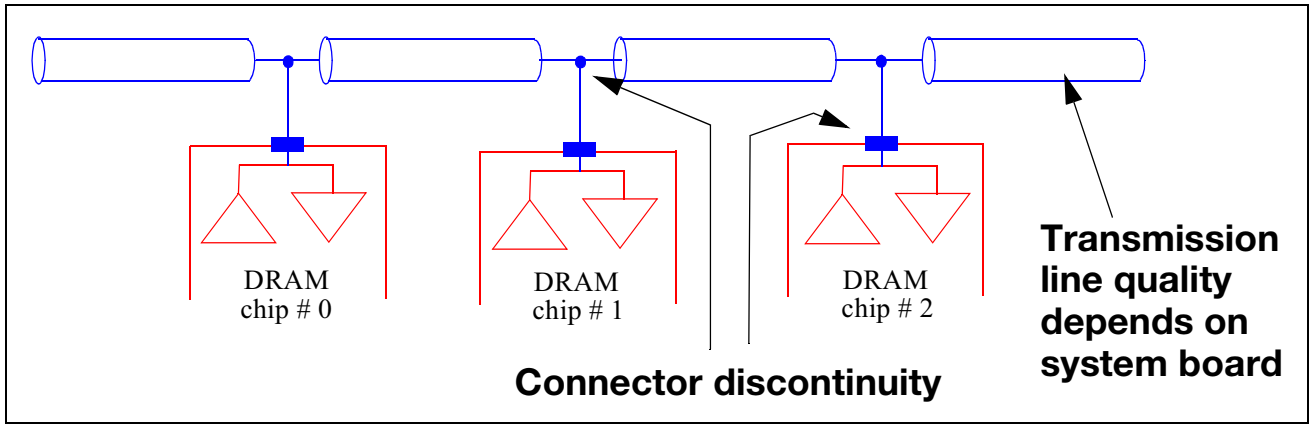
# Impedance and Reflections

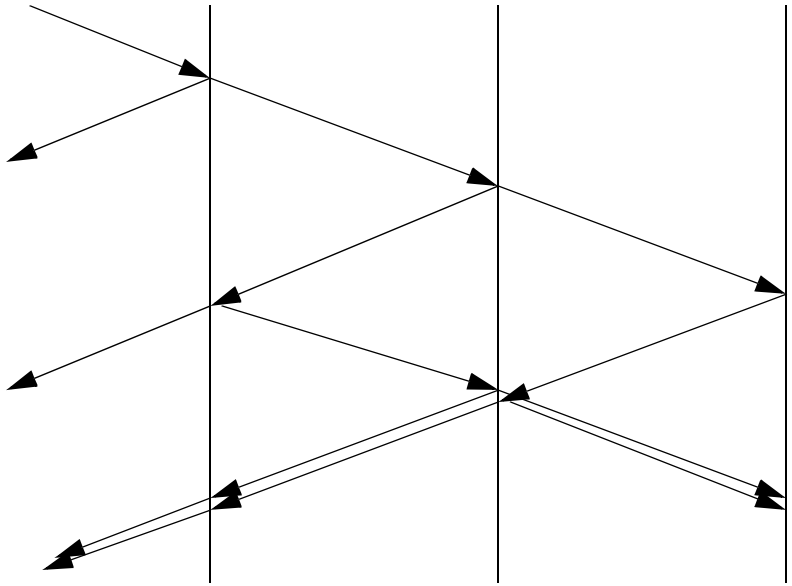



**Modern systems have MANY, MANY, MANY potential sources of impedance-mismatch and/or reflections**

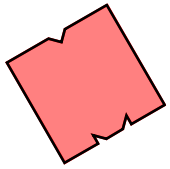


# Multidrop Bus I

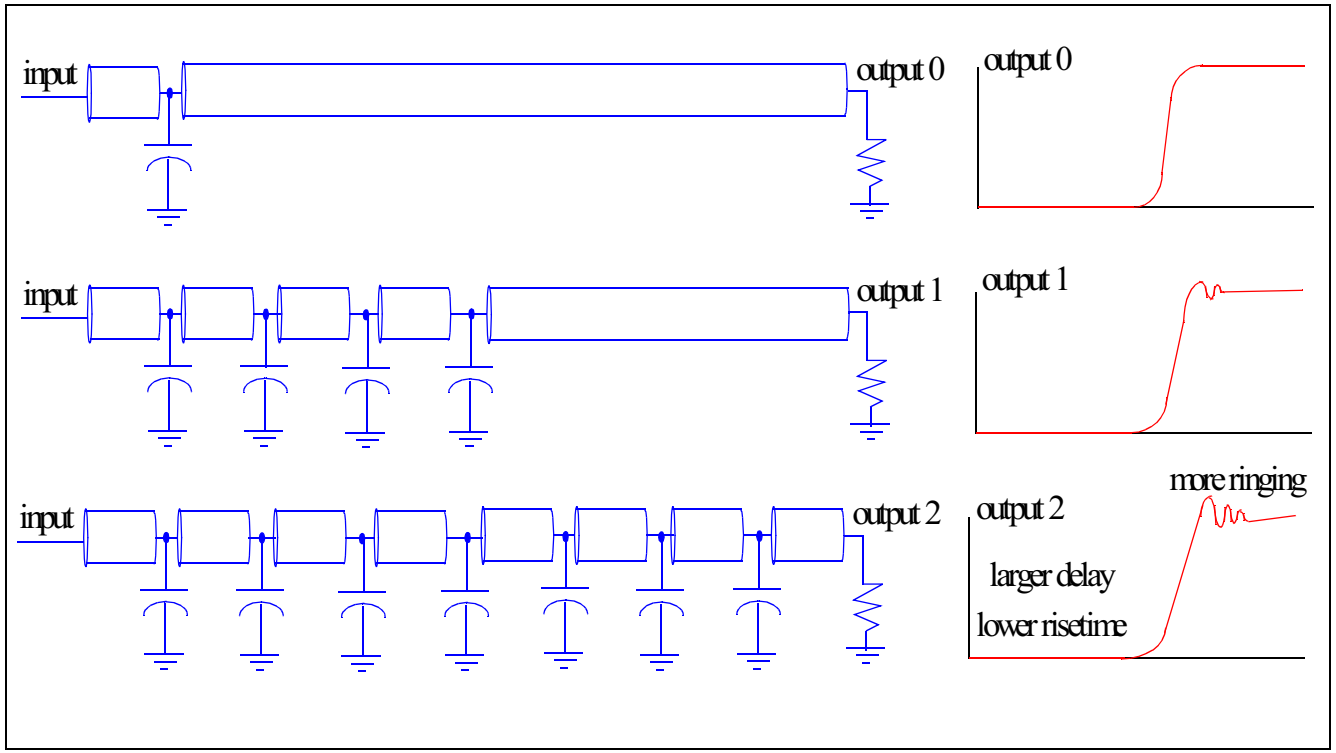


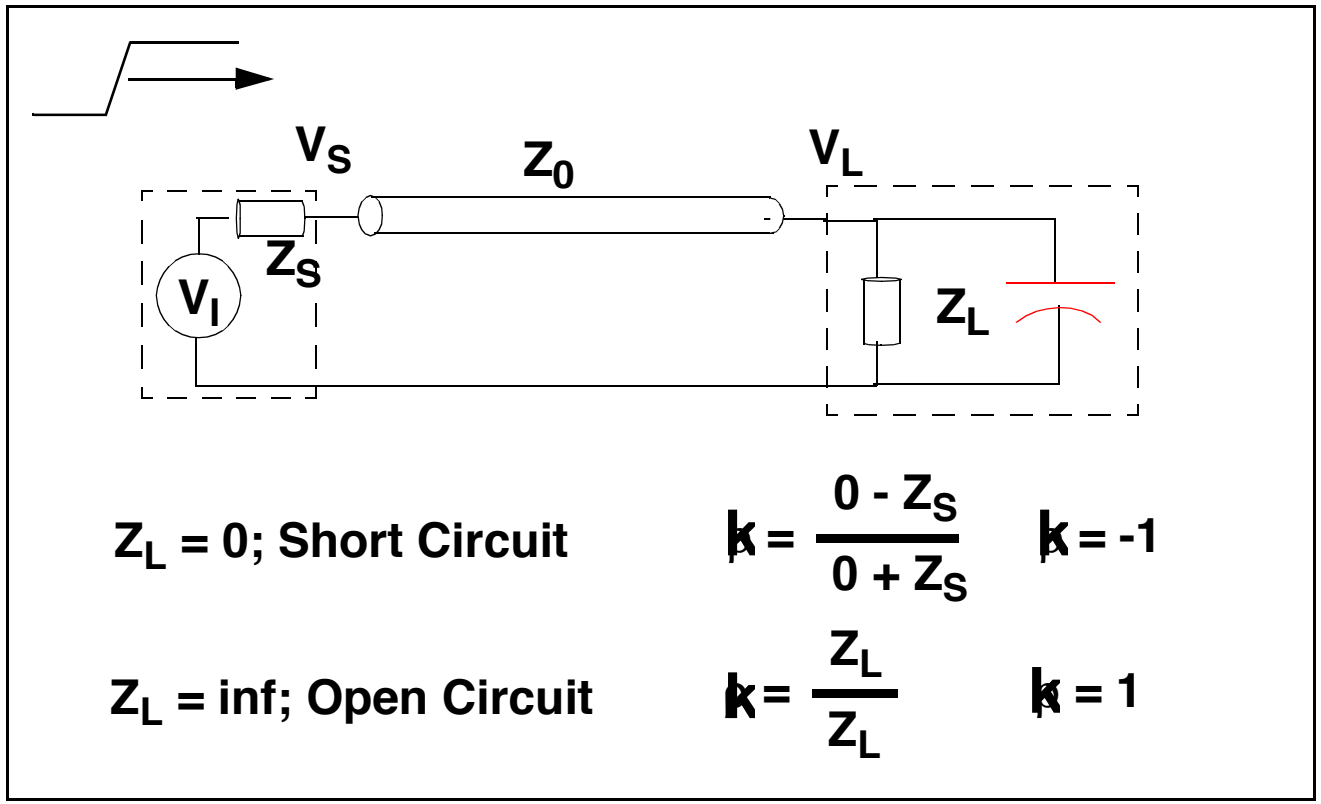
**Each impedance discontinuity is a reflective interface**



# Multidrop Bus II



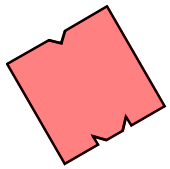
# Capacitive Termination I



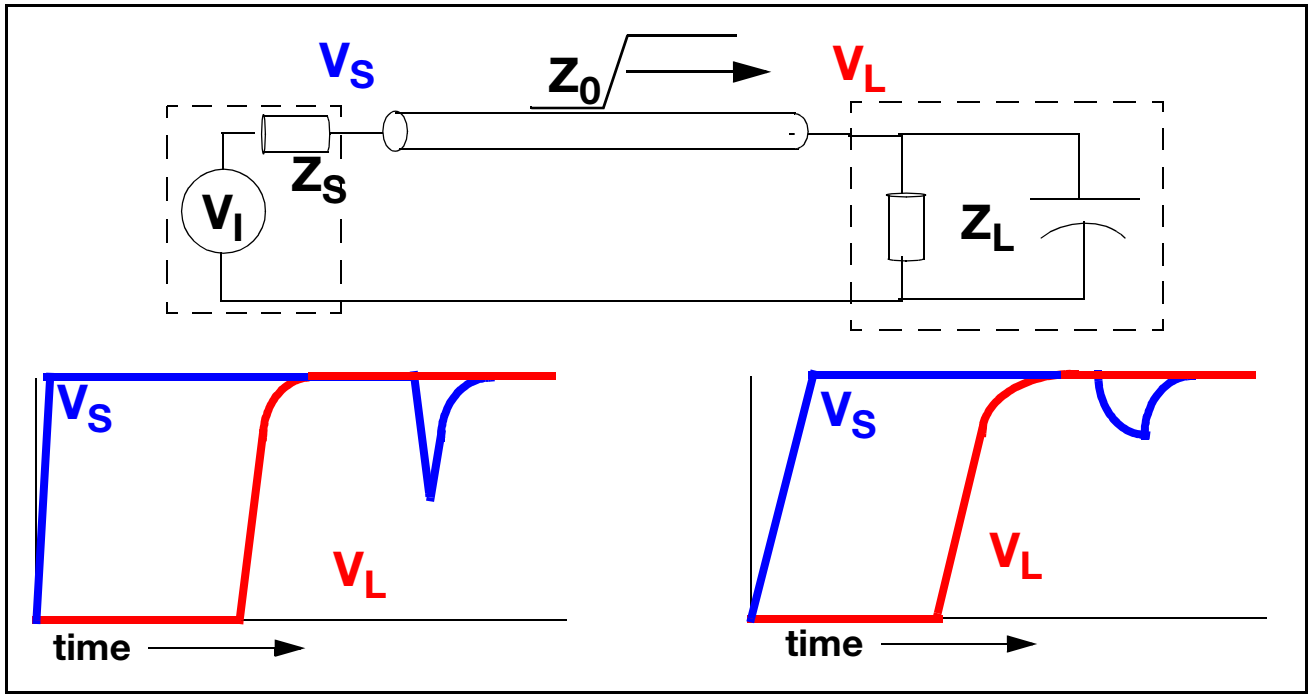

$Z_L = 0$; Short Circuit $\quad k = \frac{0 - Z_S}{0 + Z_S} \quad k = -1$

$Z_L = inf$; Open Circuit $\quad k = \frac{Z_L}{Z_L} \quad k = 1$

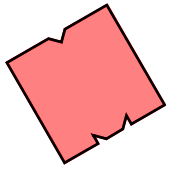
- Capacitors behave like short circuit when not charged. Once charged, behaves like open circuit.



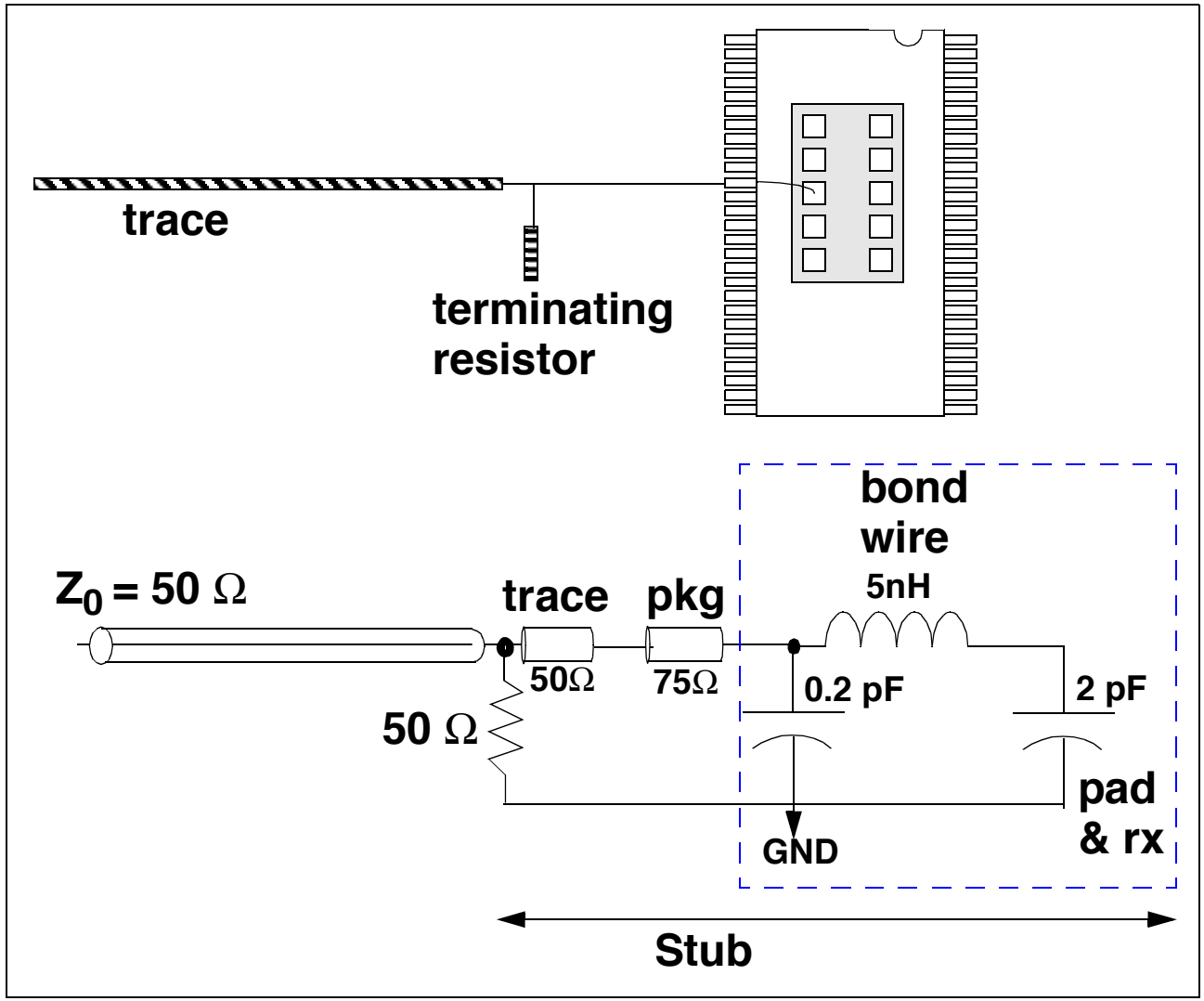
# Capacitive Termination II




- When "shorted", $V_S$ is reflected with the same magnitude back toward source.
- The "magnitude" here is the "magnitude" of rise time.
- To minimize glitch, limit slew rate.

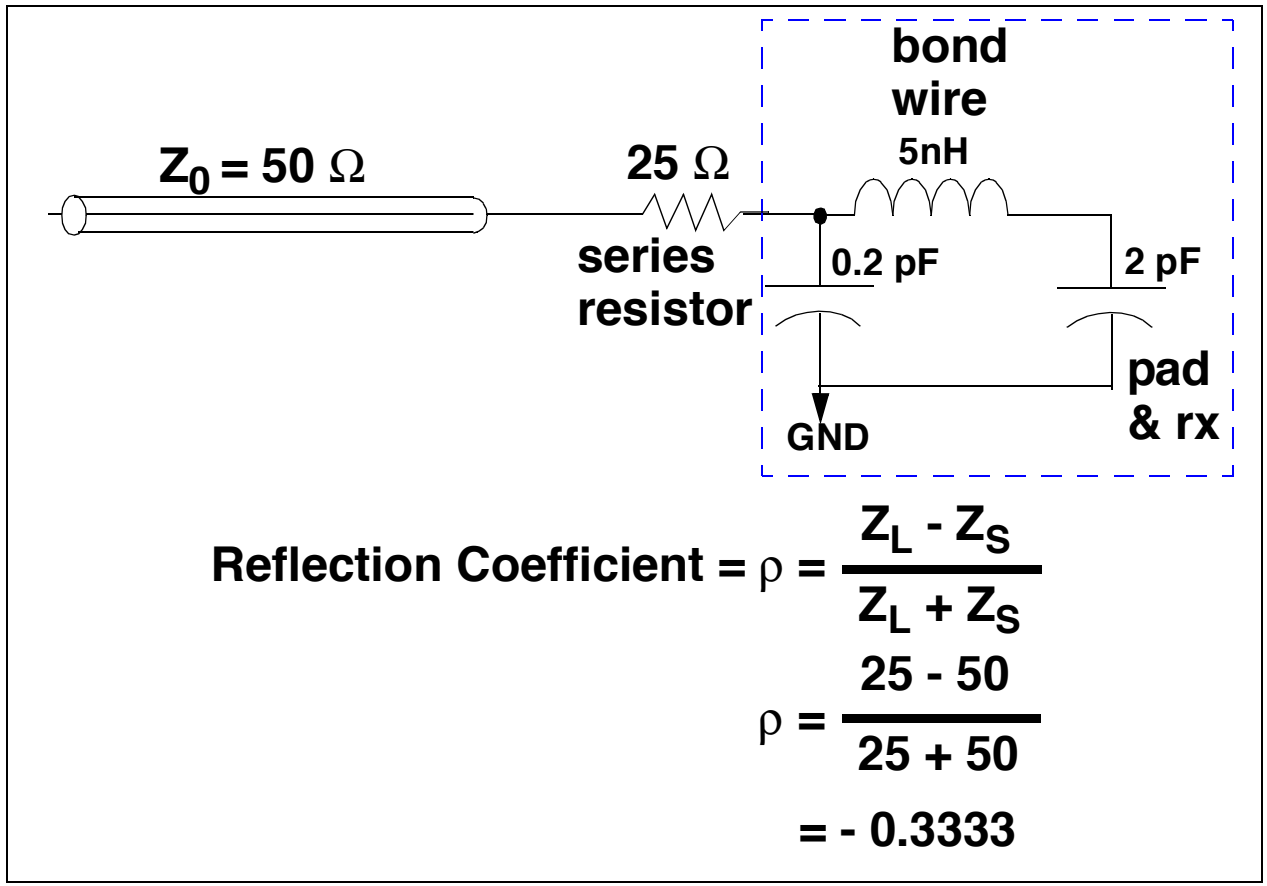


# What is a Stub?




- Signal path inside of package also significant

# Series Stub Terminated Logic



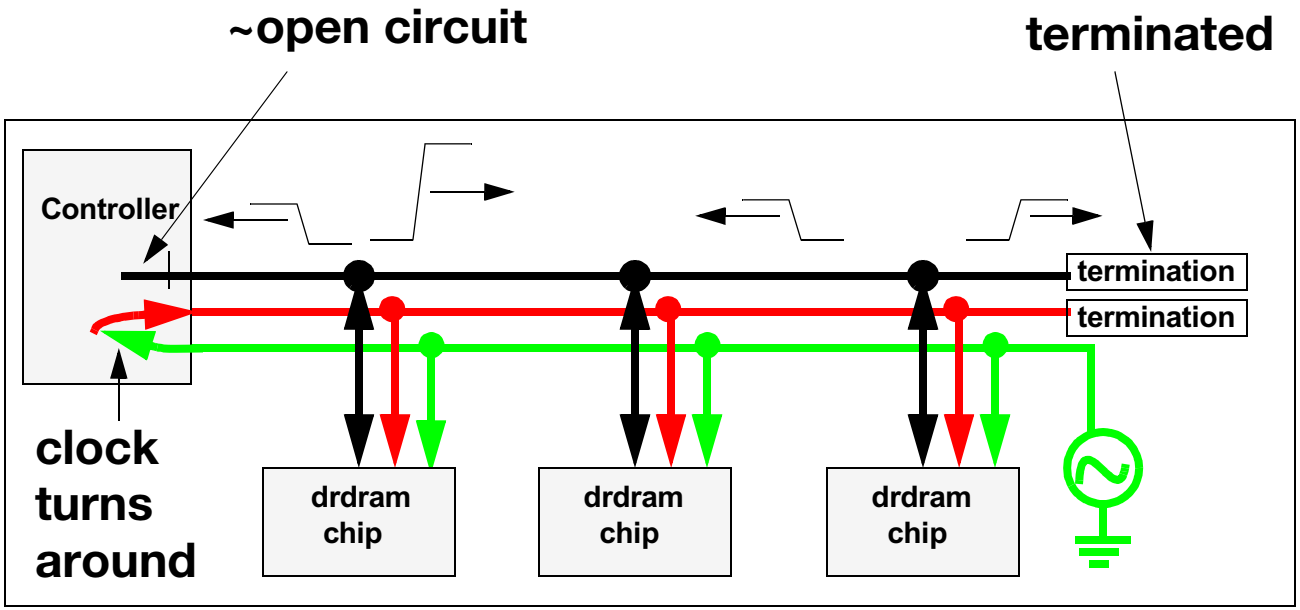

$$\text{Reflection Coefficient} = \rho = \frac{Z_L - Z_S}{Z_L + Z_S}$$

$$\rho = \frac{25 - 50}{25 + 50}$$

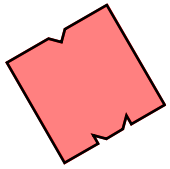
$$= -0.3333$$

- Series resistor isolates stub from line
- reduces ringing
- reduces power

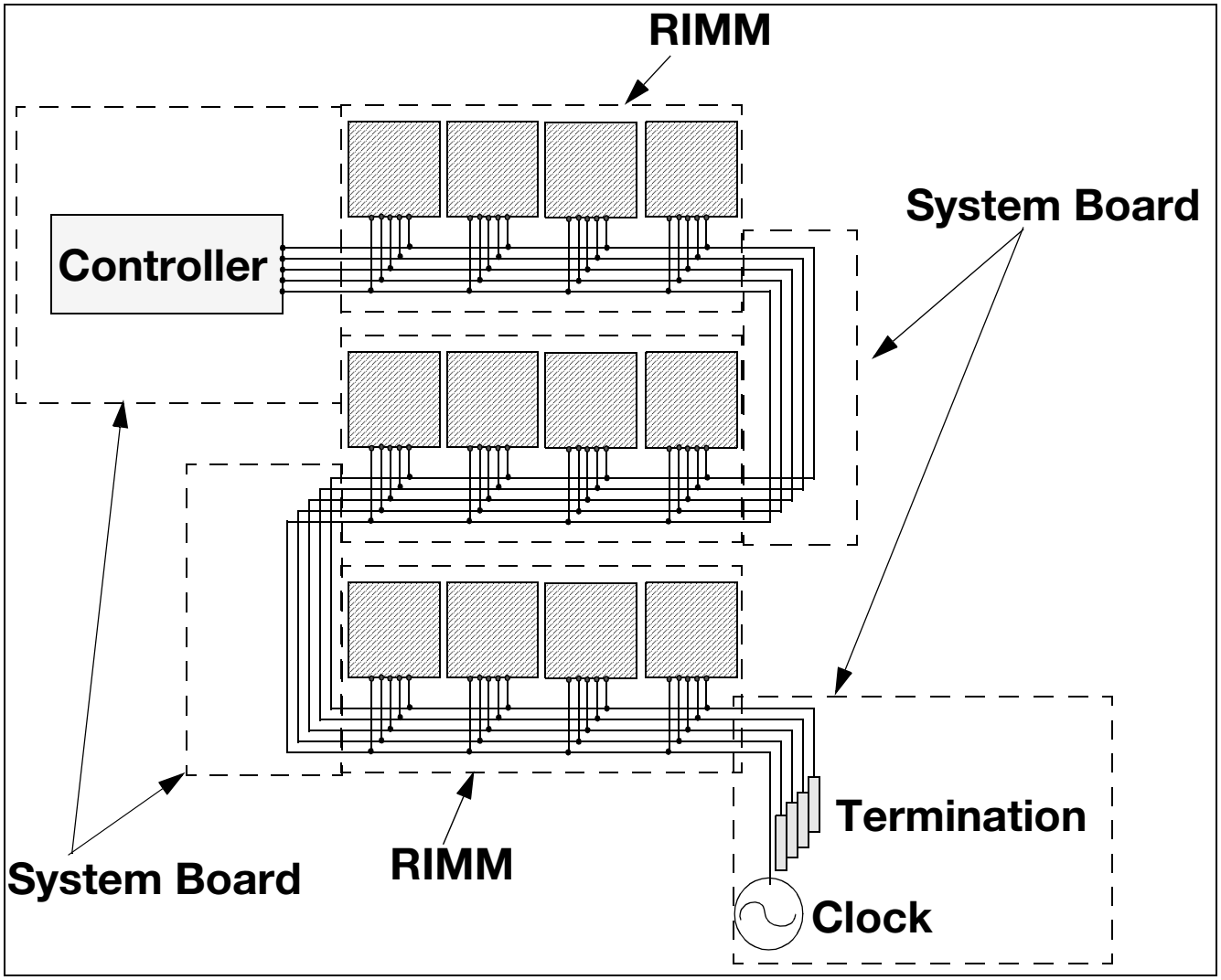
# Rambus Signalling



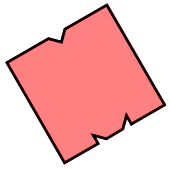

- Current mode
- Controller sends full height signal swings
- DRAM chips send half height signal swings
- Reflection off of controller driver
  creates full height signal swing
- Current control and voltage slewrate control



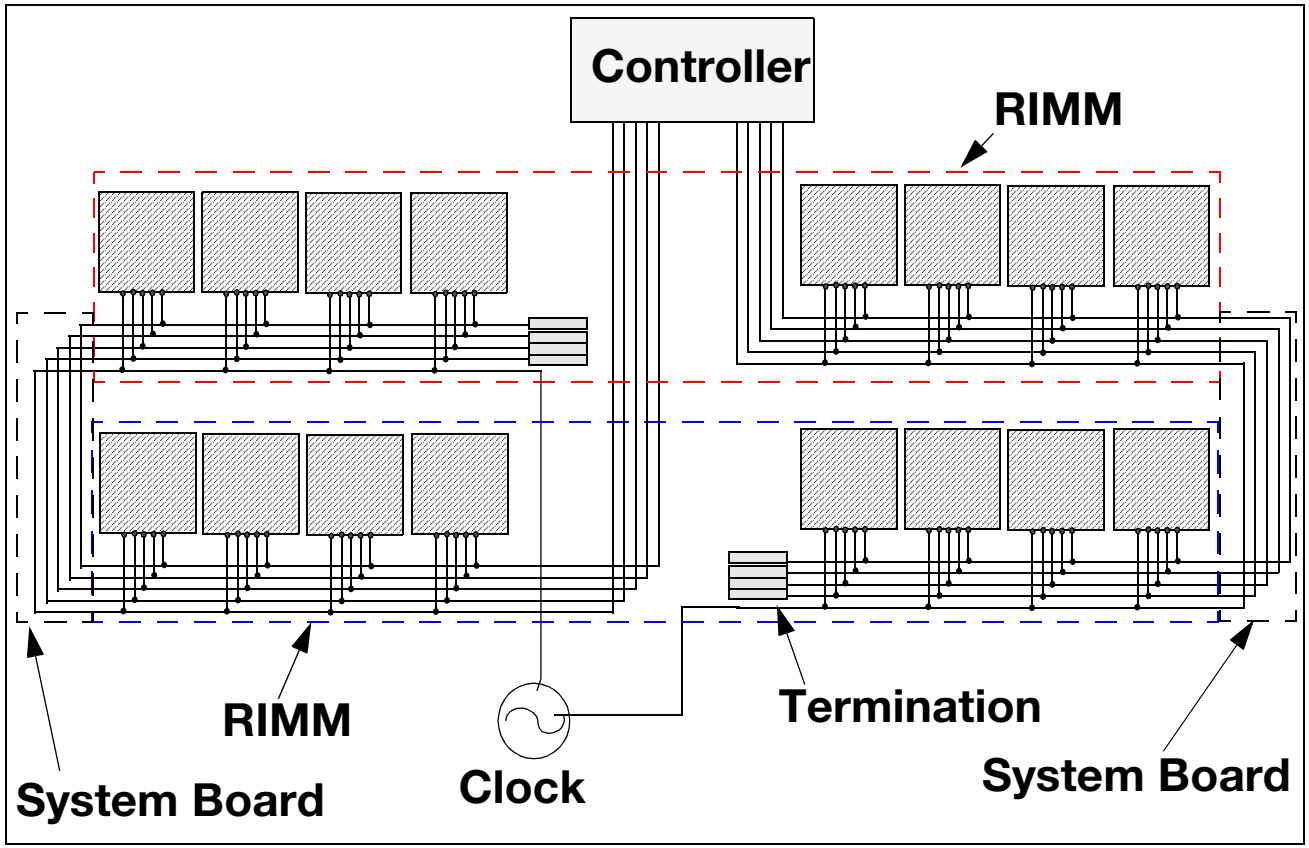
# Rambus Channel - 16 bit



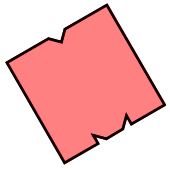

- RIMM #3 dropped from spec to solve signalling problems
- Need Continuity RIMM



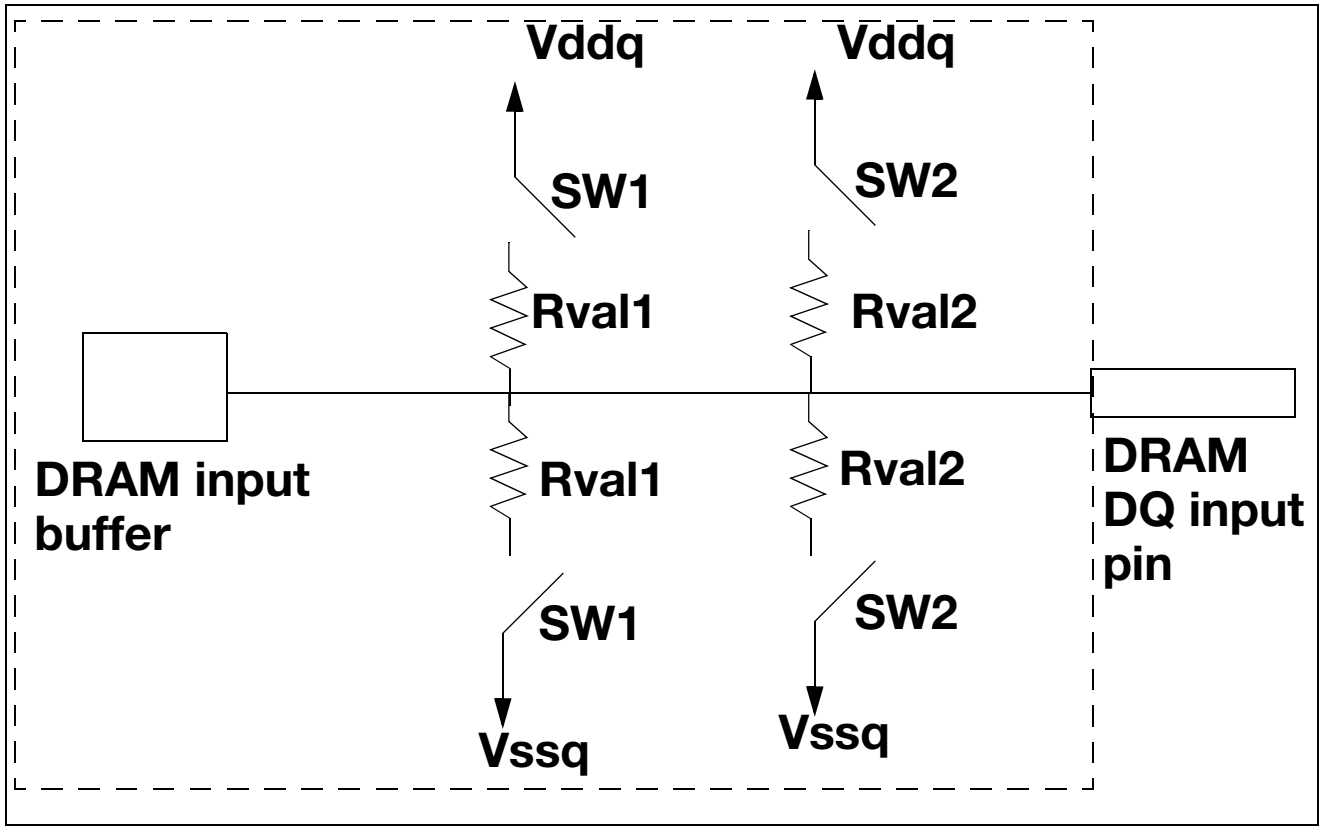
# Rambus Channel - 32/64 bit



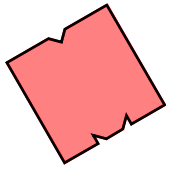

- Termination brought onto RIMM
- Connector interface count reduced from 4 to 3
- No need for special terminated RIMM and non-terminated RIMM



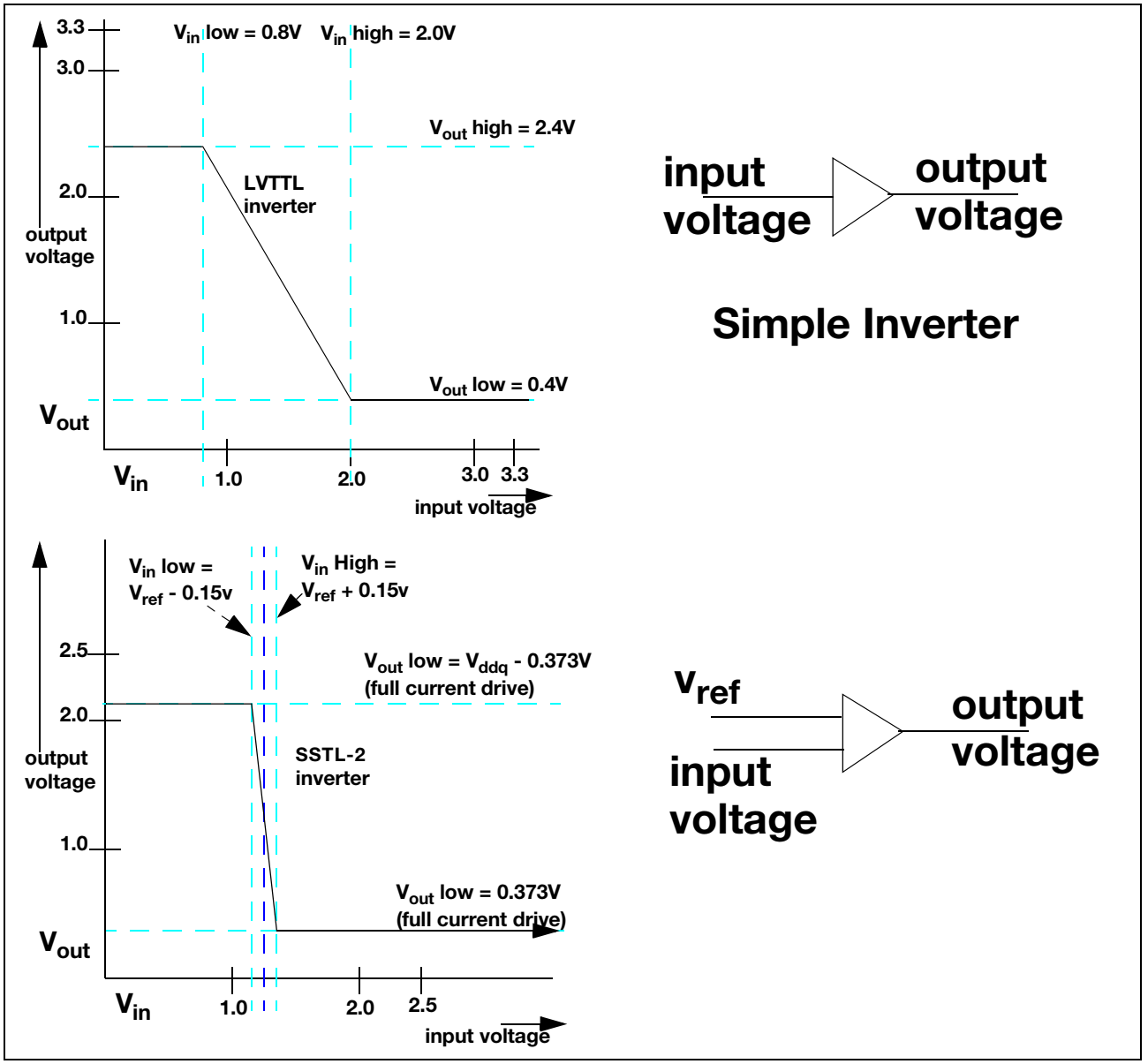
# ODT: On Die Termination

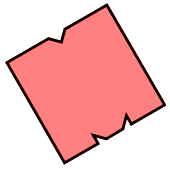



- Active, dynamic termination, depending on R/W, and number of loads on electrical bus
- Can be turned on/off in 2 cycles, off in 2.5 cycles
- Designed into DDR II

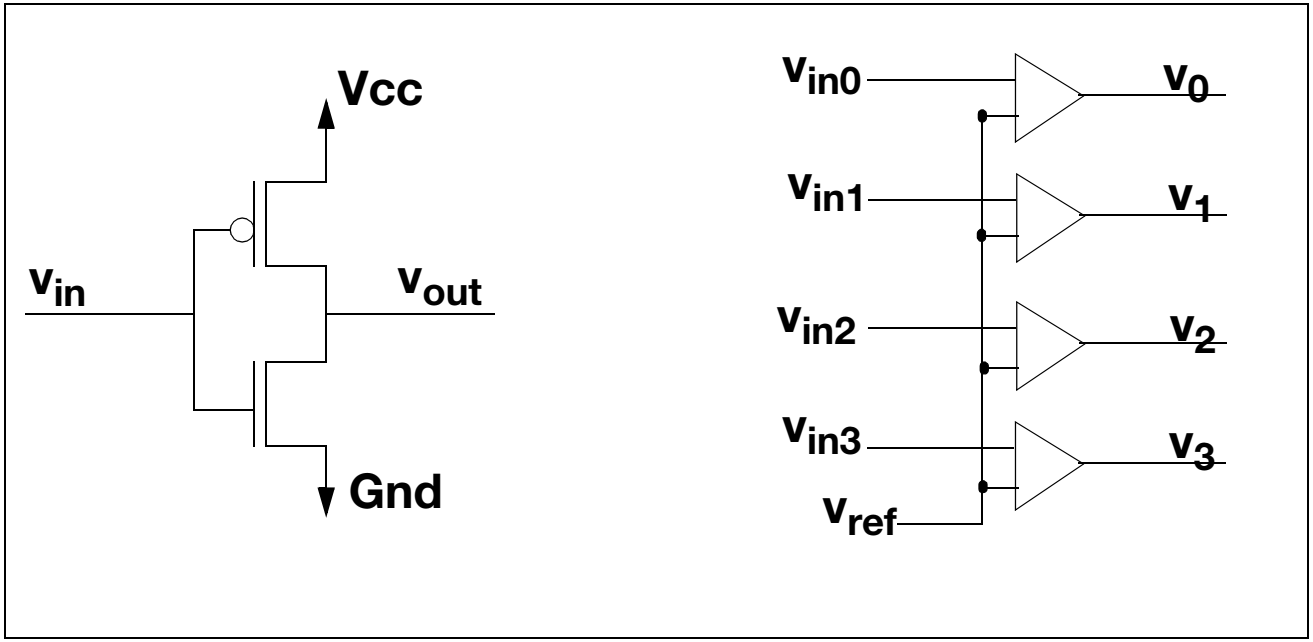


# Input Buffering - References I

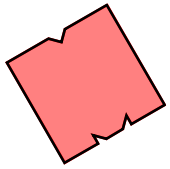


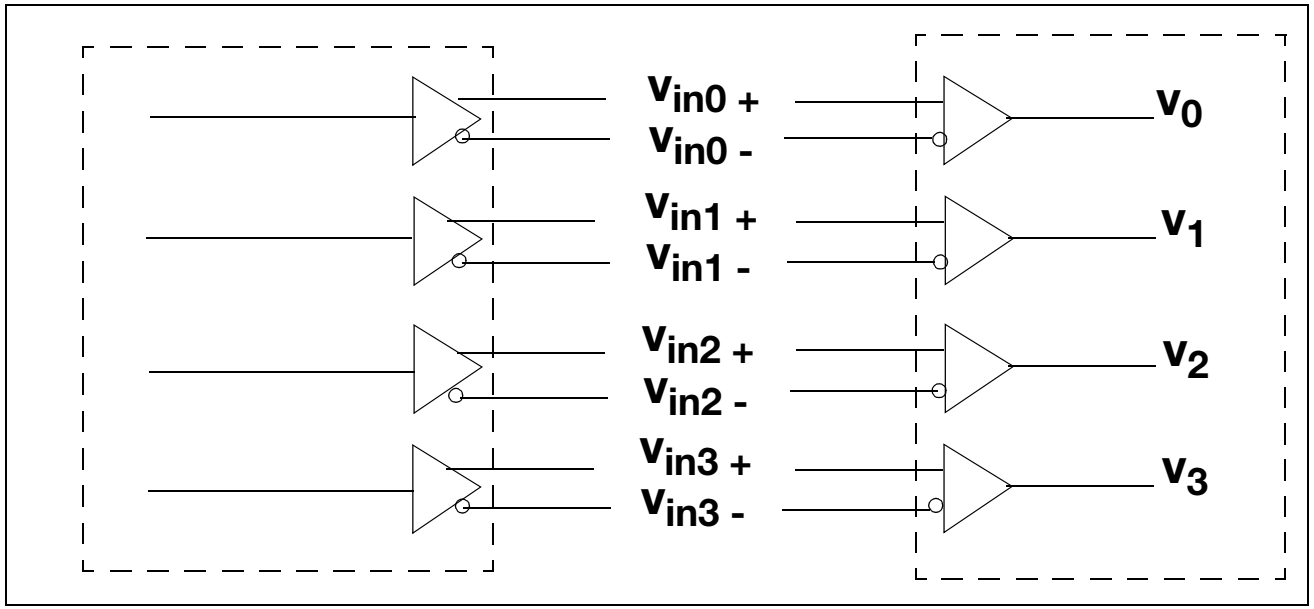
# Input Buffering - References II



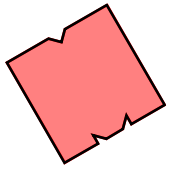

- Inverters aren't very good for high frequency signalling
- Local or remote reference?



# Differential Signalling

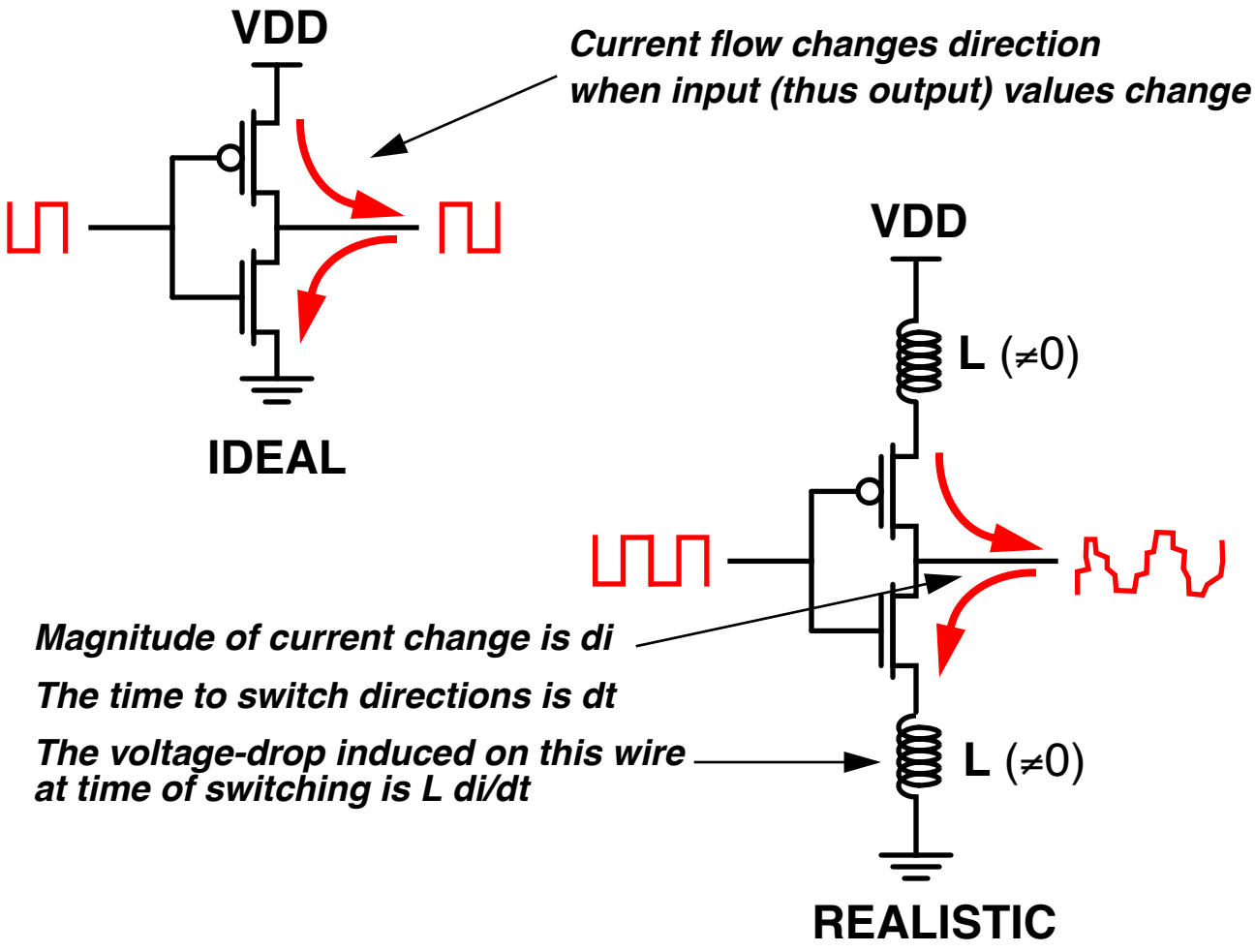



- Send the signal and its complement
- The complement signal path is the current return path
- Signal pairs must be routed closely together
- Signalling scheme can reject common mode noise quite well



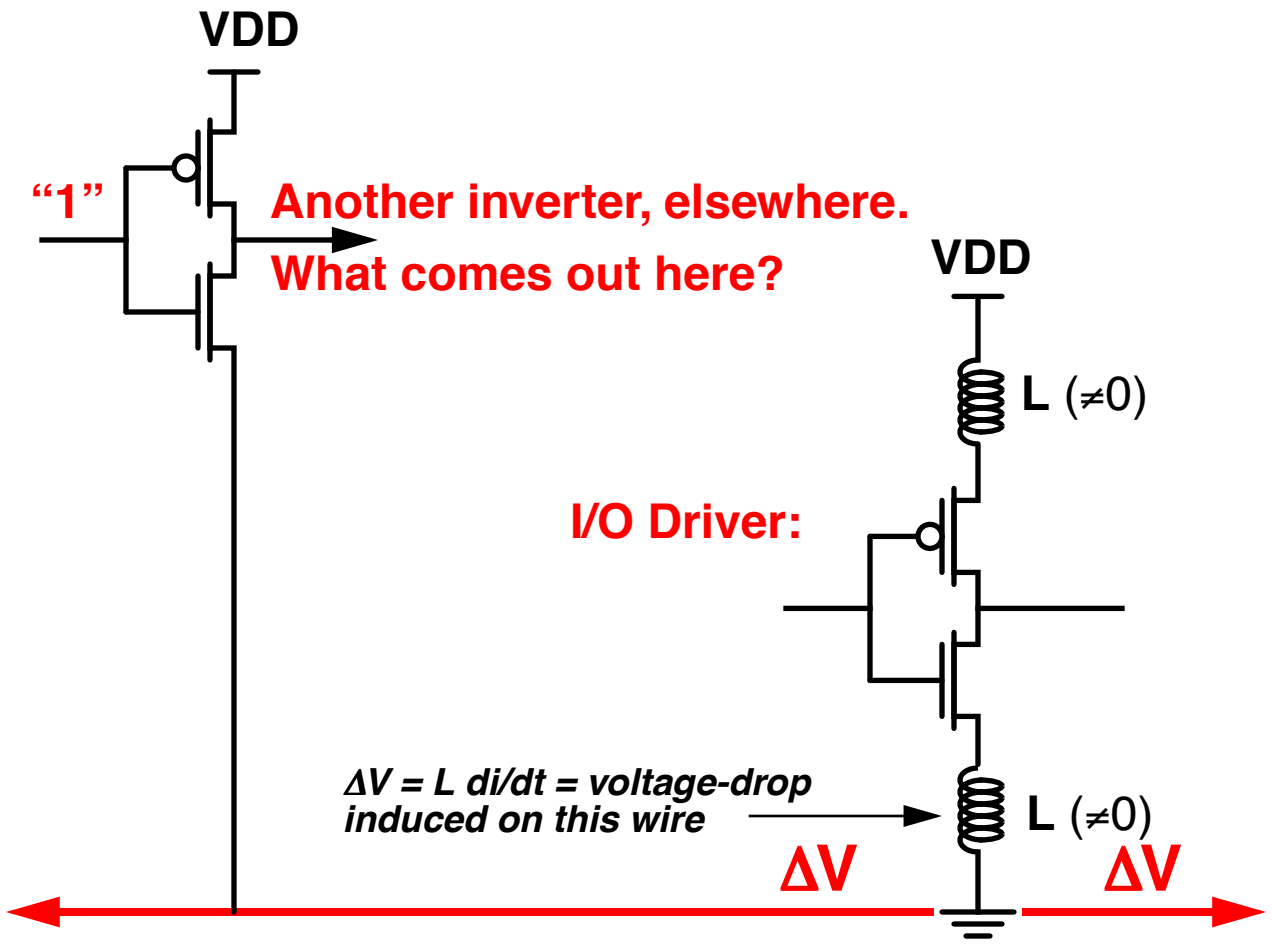
# Inductive Noise

## L di/dt noise (ground bounce):

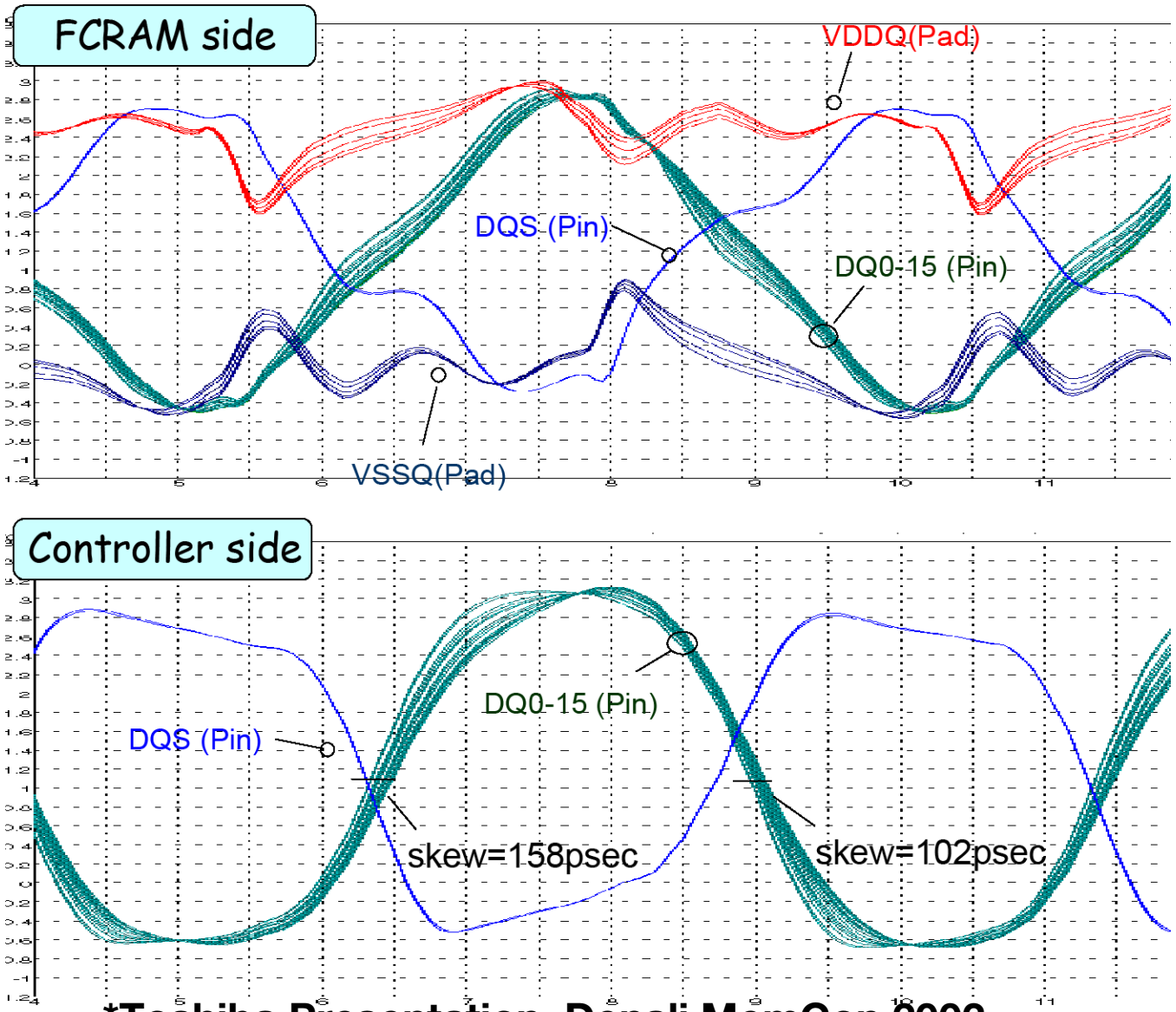


# Inductive Noise

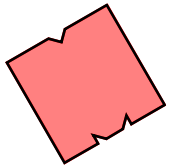
## L di/dt noise (ground bounce):



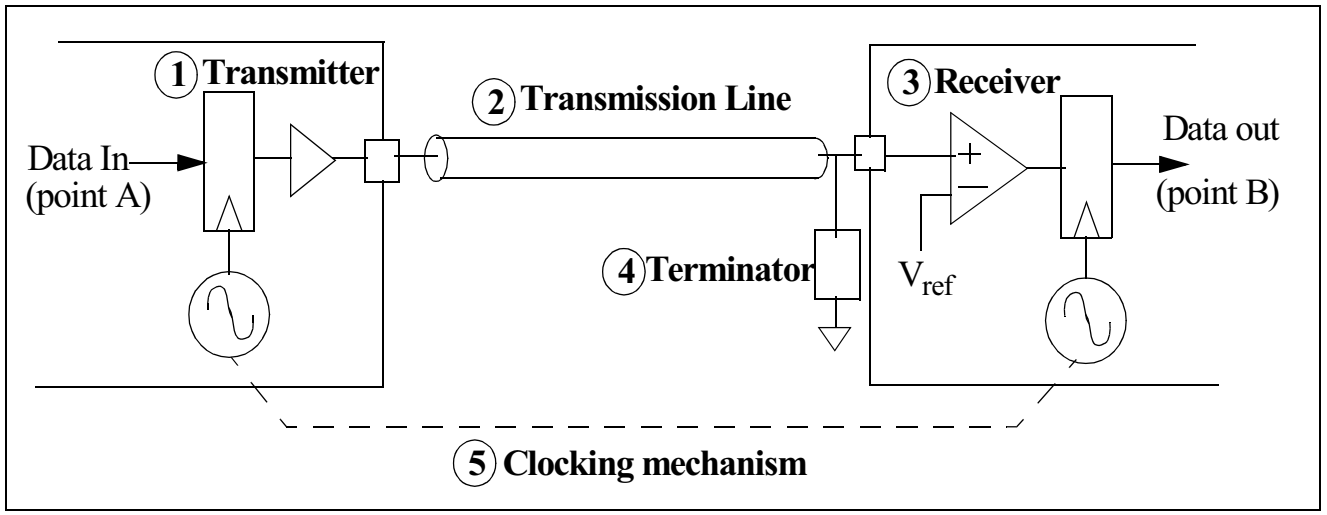
# Real vs Ideal World




*Toshiba Presentation, Denali MemCon 2002



# A Signaling System




**1. Transmitter: Encodes data as current/voltage level onto the line**

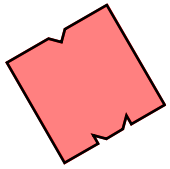
**2. Transmission Line: Deliver data from transmitter to receiver**

**3. Receiver: Compare against reference to extract data**

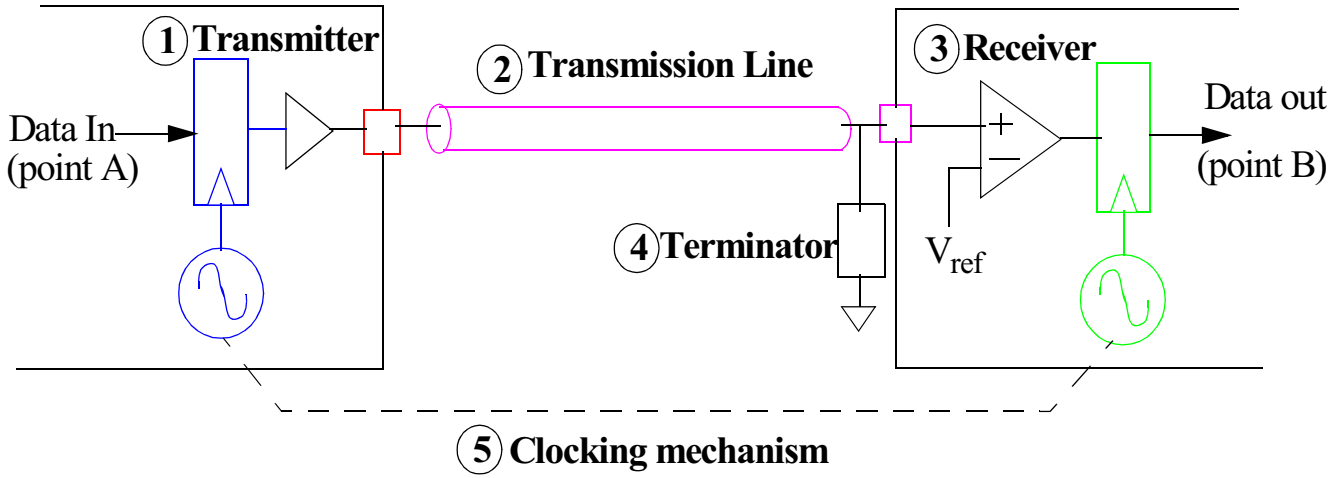
**4. Terminator: Remove signal from line, once they're received**

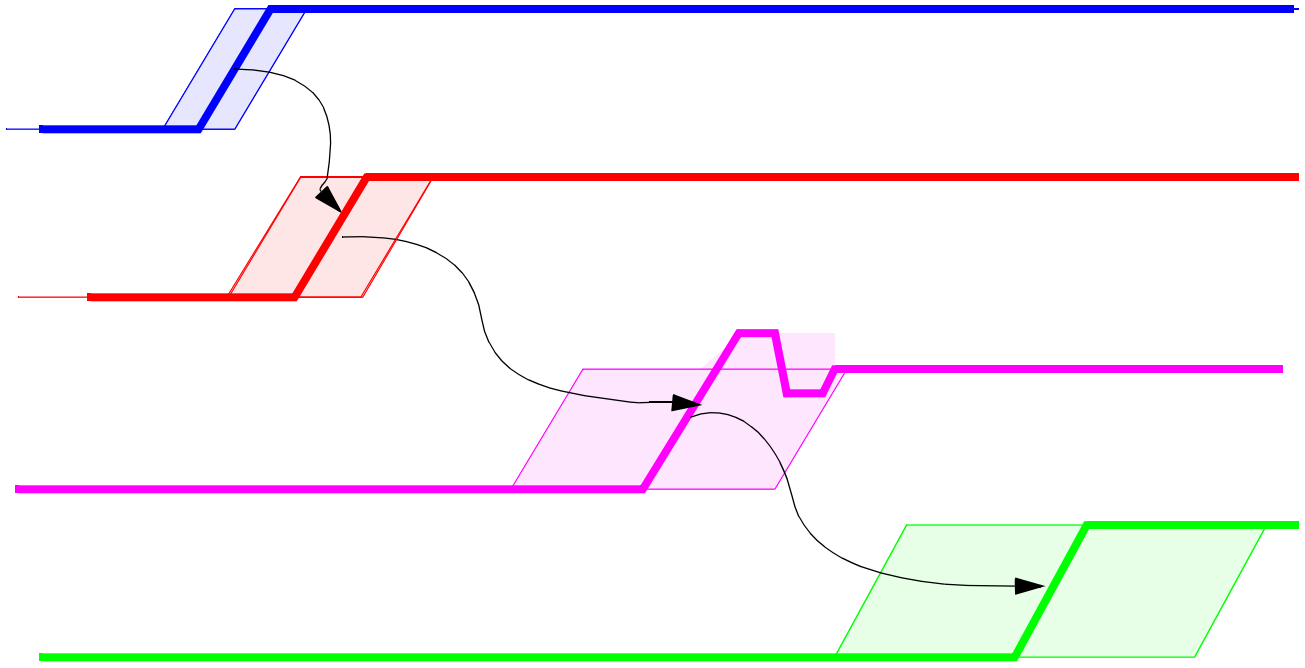
**5. Clock: Tells transmitter when to send, receiver when to sample signal**

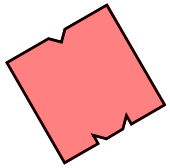
* Poulton ISSCC 1999 Signaling Tutorial



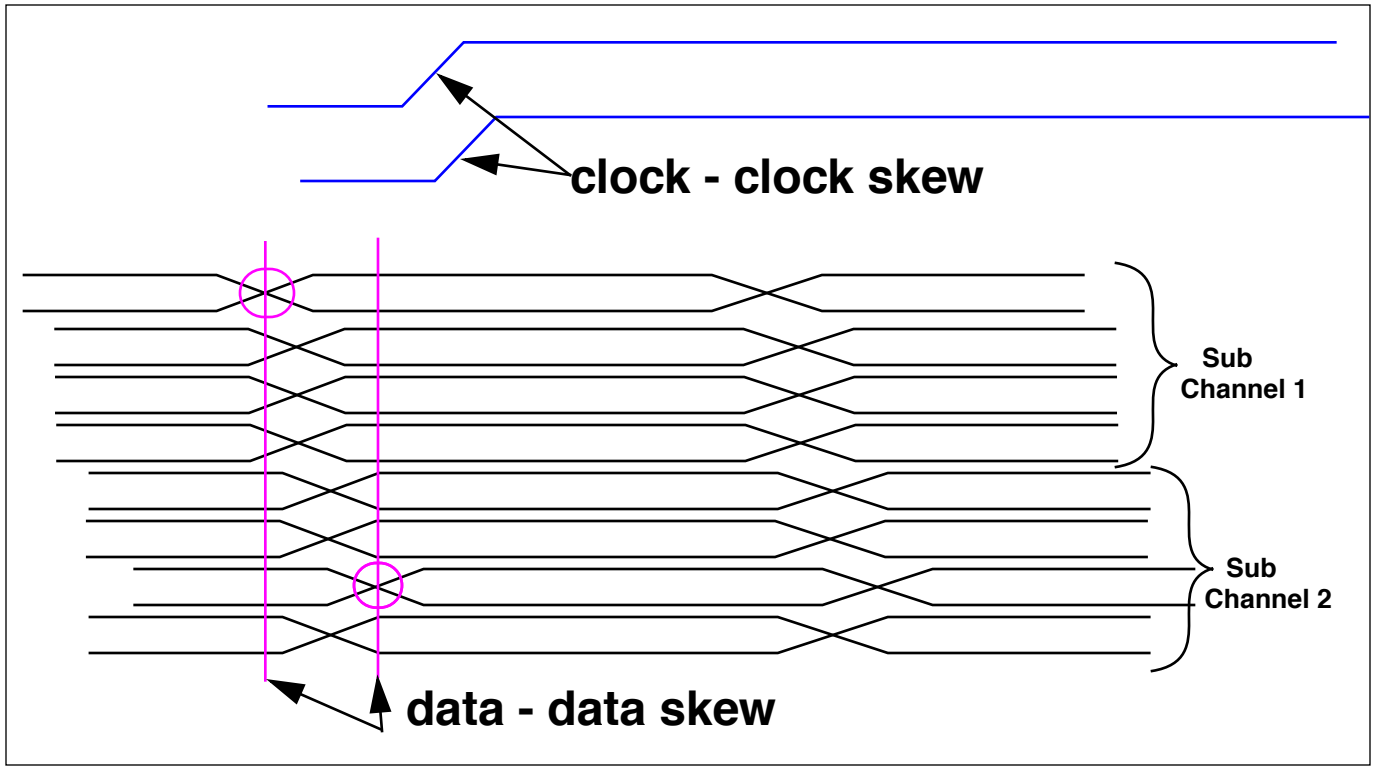
# Timing Budget Visualized



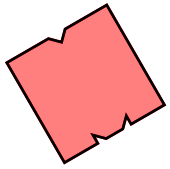





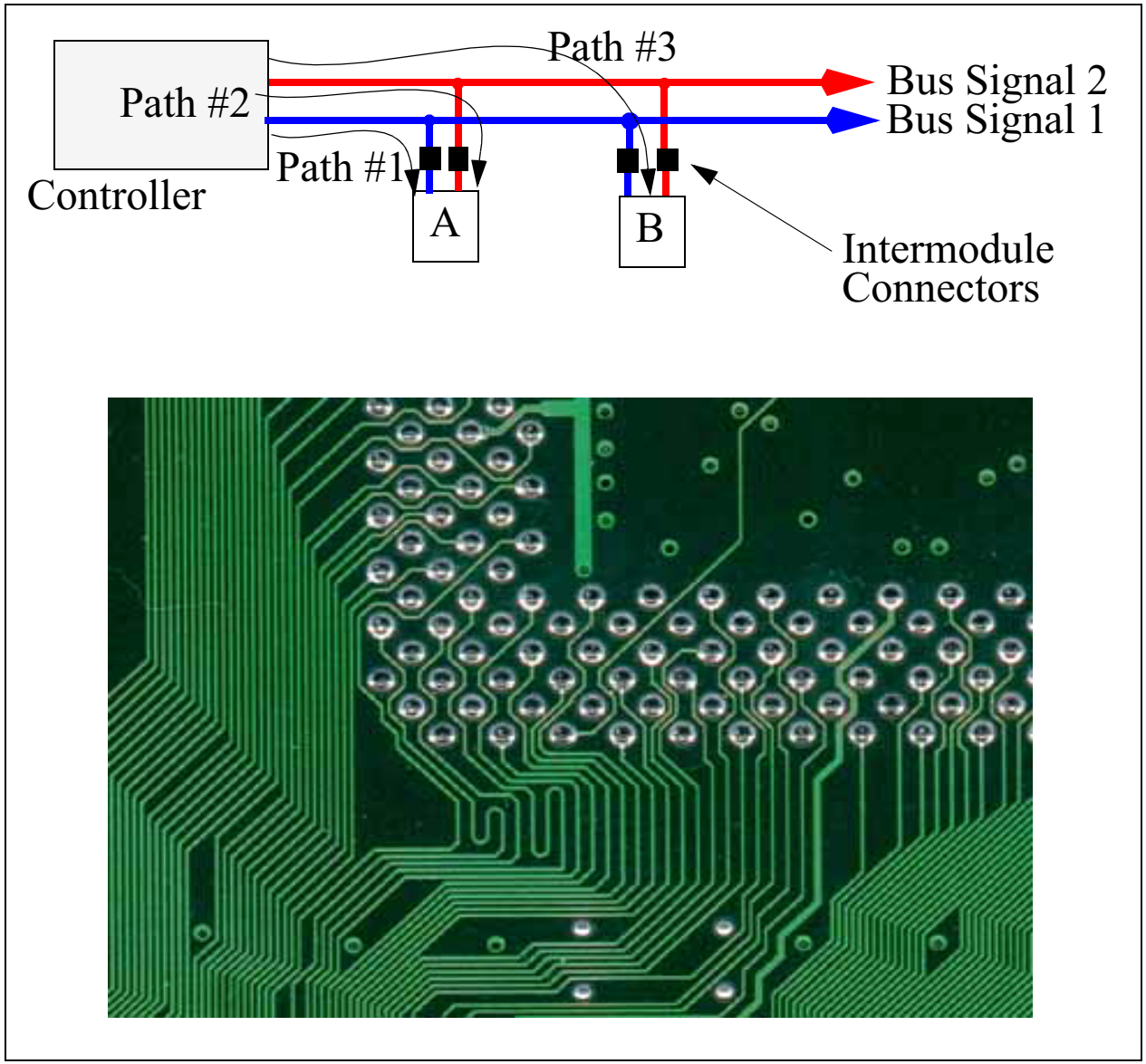
# Definitions: *Skew*



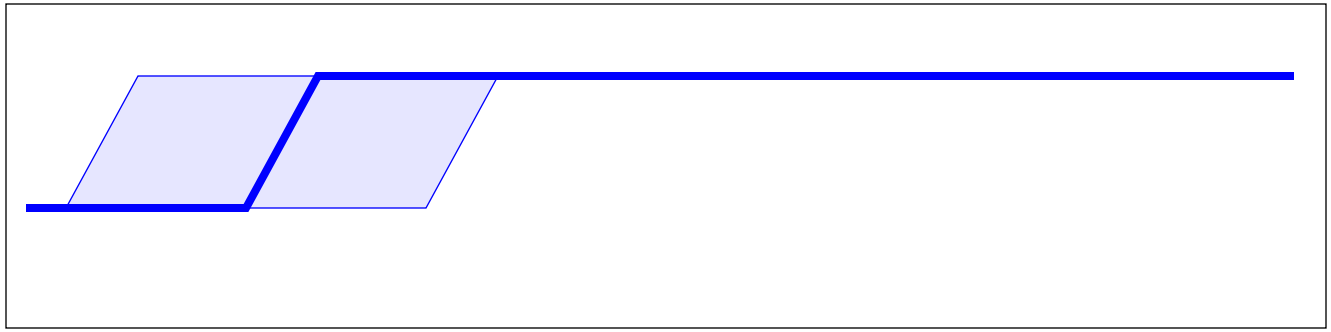

- Static timing displacement from ideal design
- Caused by differences in signal path characteristics
- Total timing budget must take data-data skew, data-clock skew as well as clock-clock skew into cycle budget consideration



# Path Length Differential

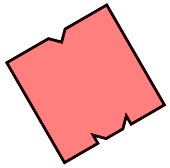


# Definitions: *Jitter*

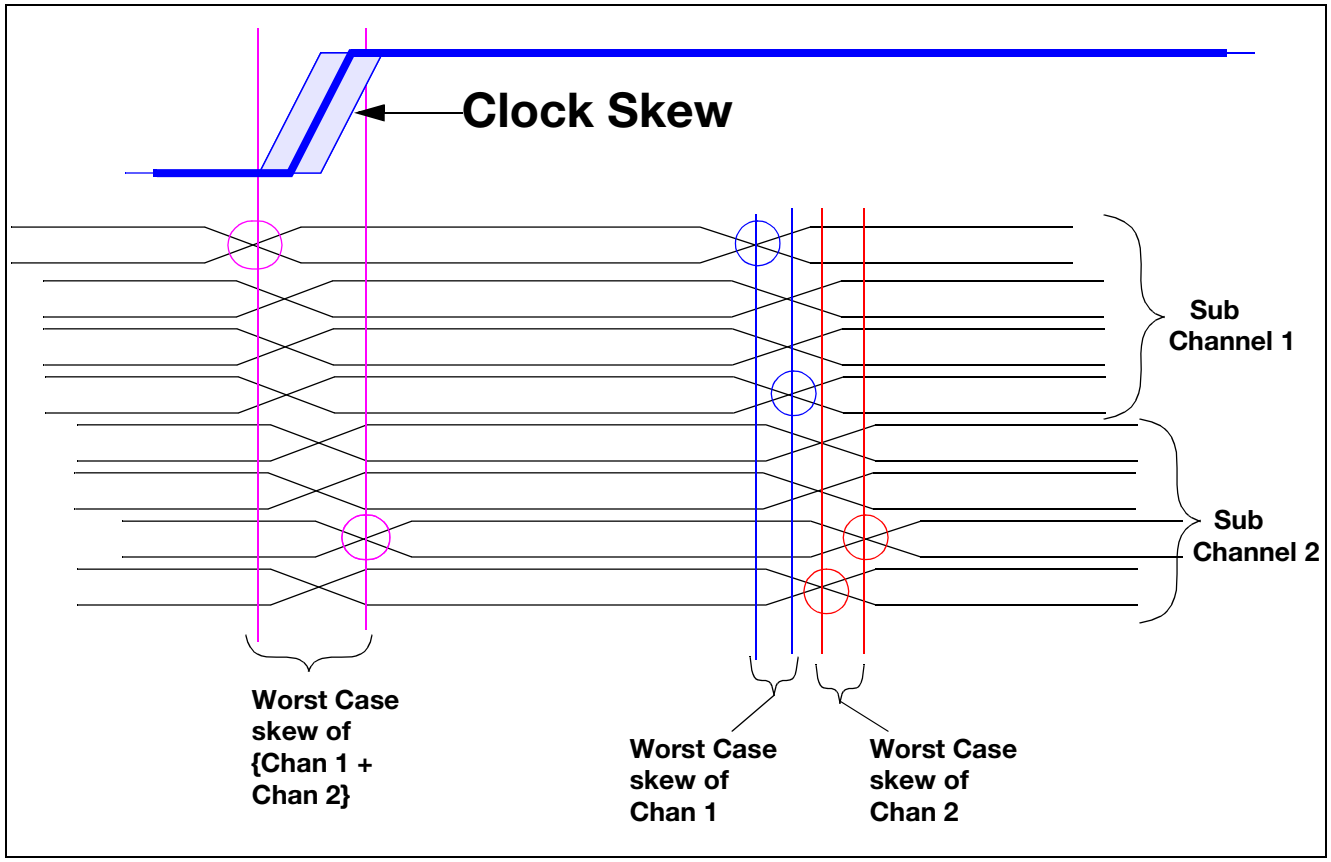


- Dynamic timing displacement from nominal timing characteristics

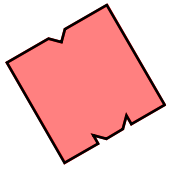
- Magnitude and offset of timing displacement could depend on: previous signal state(s), current signal state(s), supply voltage level(s), crosstalk, variations in thermal characteristics. Perhaps even phases of the moon (not proven).



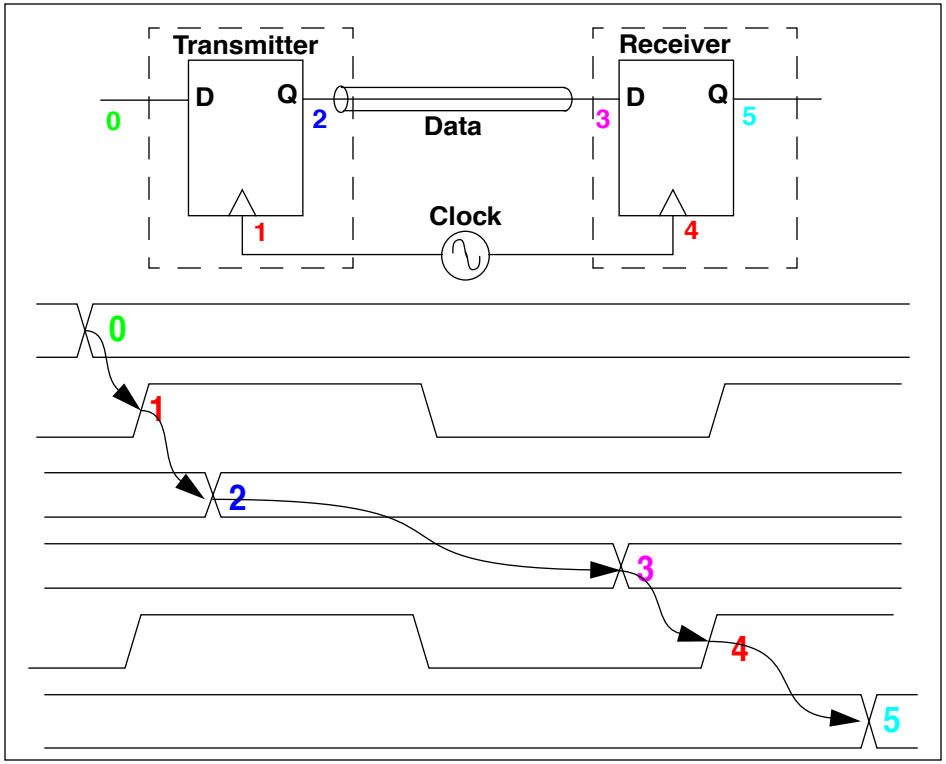
# Uncertainty == Bad



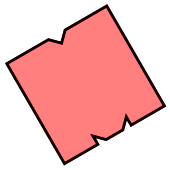

**Worst case skew must be considered in system timing**



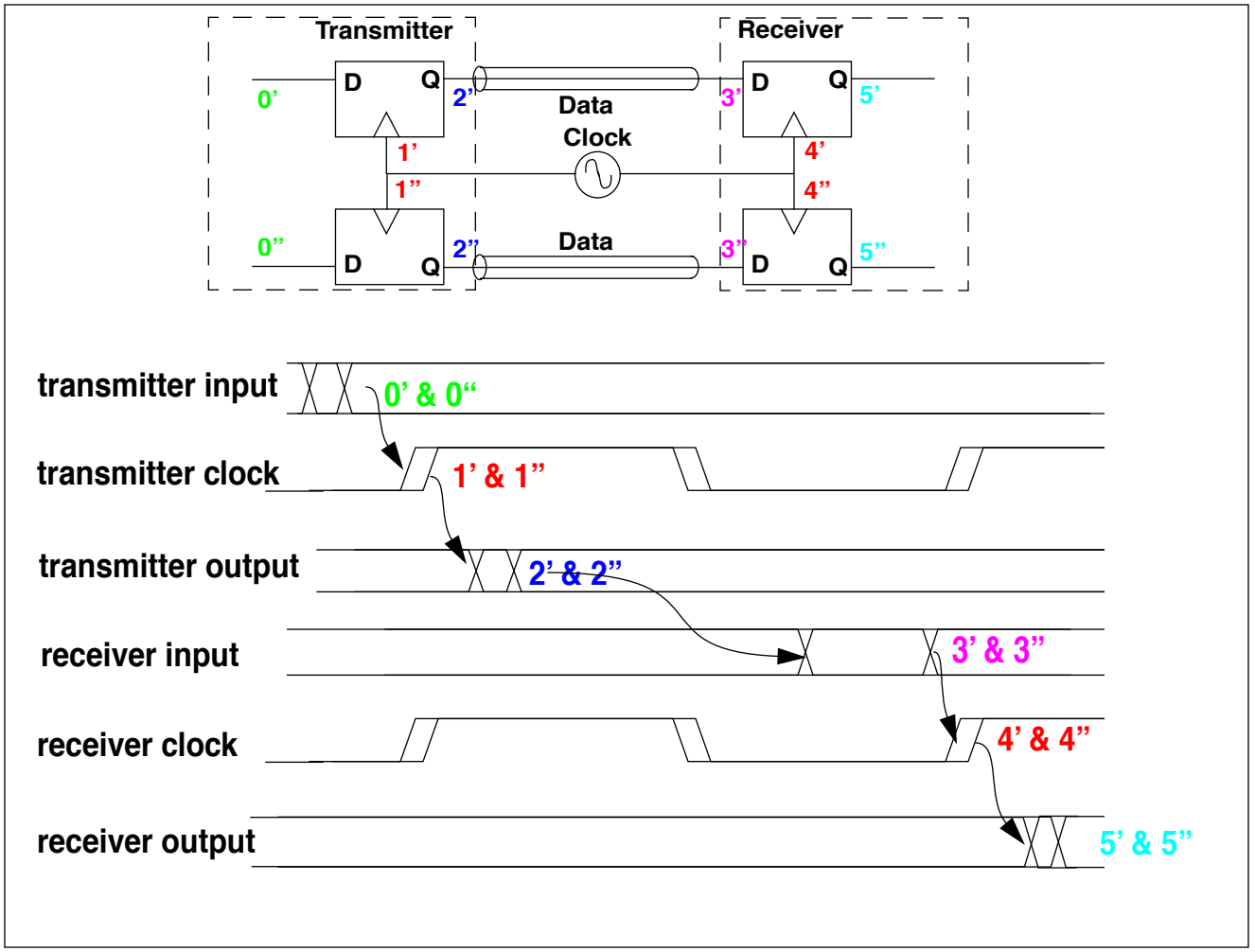
# Global Clock I



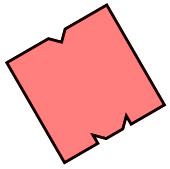

**0: Assume data is stable for setup time before clock edge**
**1: Rising edge of transmitter clock**
**2: Transmitter begins to drive data (perhaps through logic)**
**3: Signal reaches input of receiver.**
**4: Rising edge of receiver clock**
**5: Receiver latches data and drives internal signal lines**



# Global Clock II: Parallel Data




- Skew and jitter eats into timing budget
- Luckily, uncertainty does not accumulate beyond latches

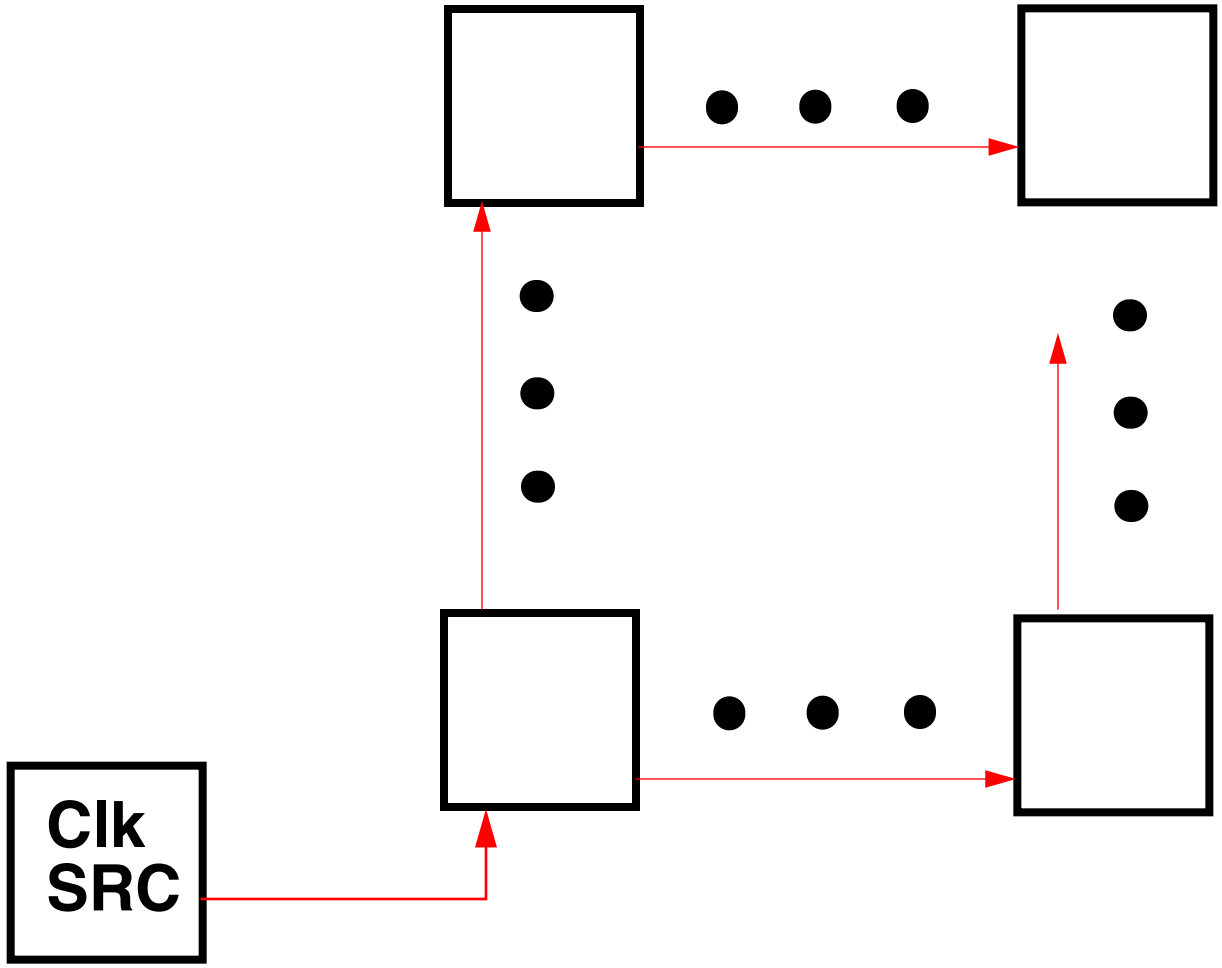


Bruce Jacob

University of
Maryland
ECE Dept.



# What Needs To Be Done?




**By the way: “global” clock scheme**

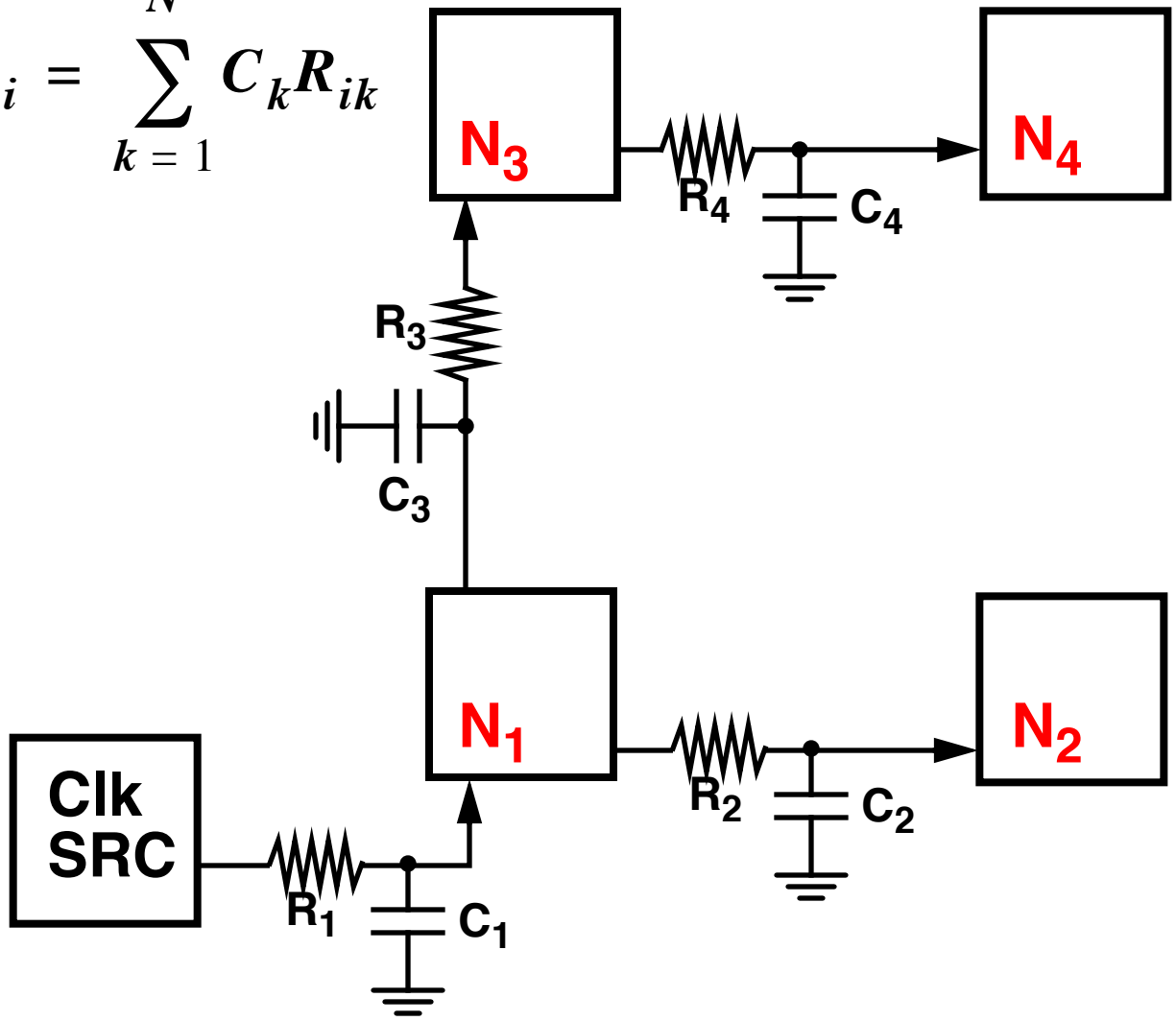
Bruce Jacob

University of
Maryland
ECE Dept.



# Background: Elmore Delays

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$



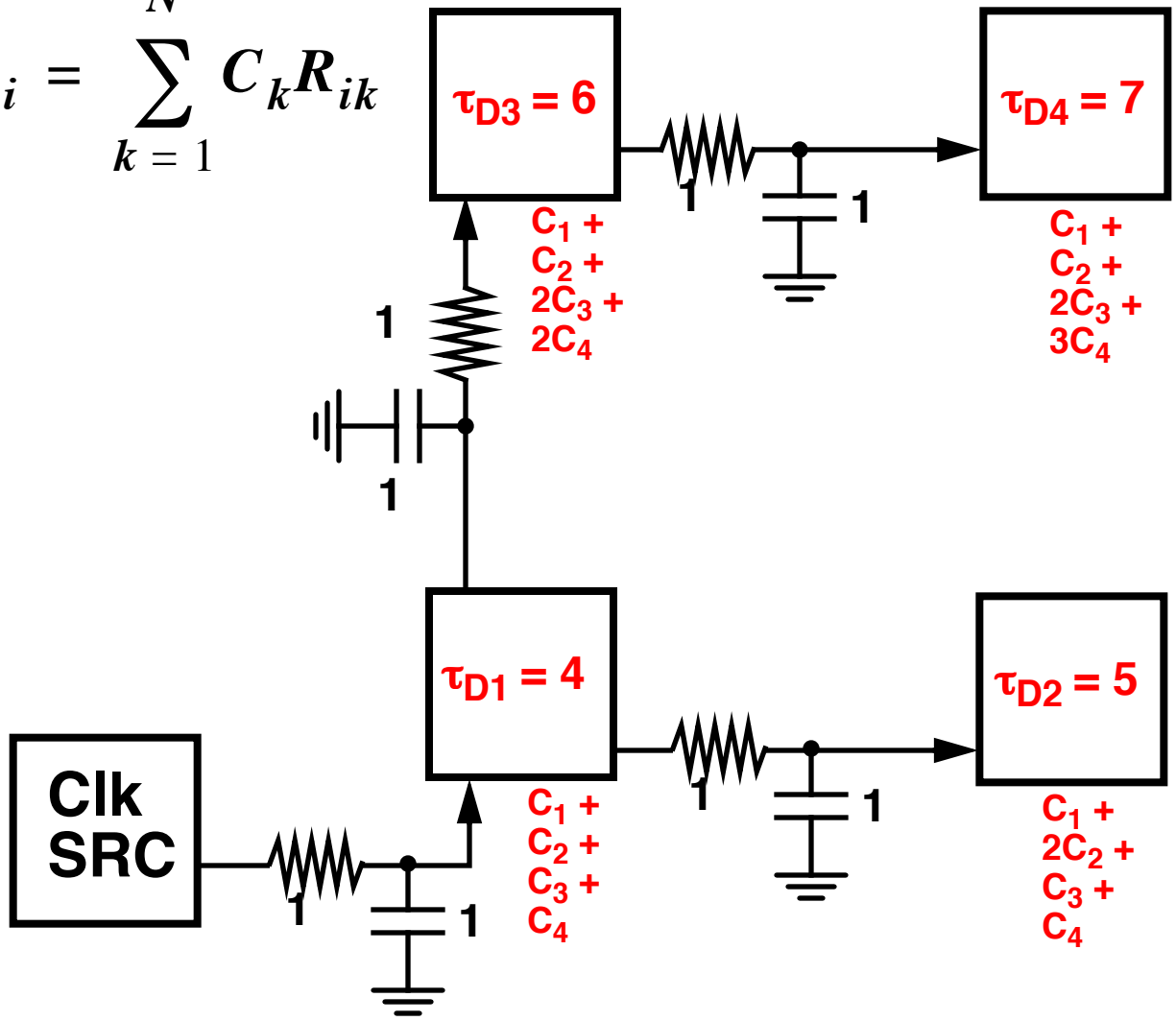
Bruce Jacob

University of
Maryland
ECE Dept.



# Background: Elmore Delays

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

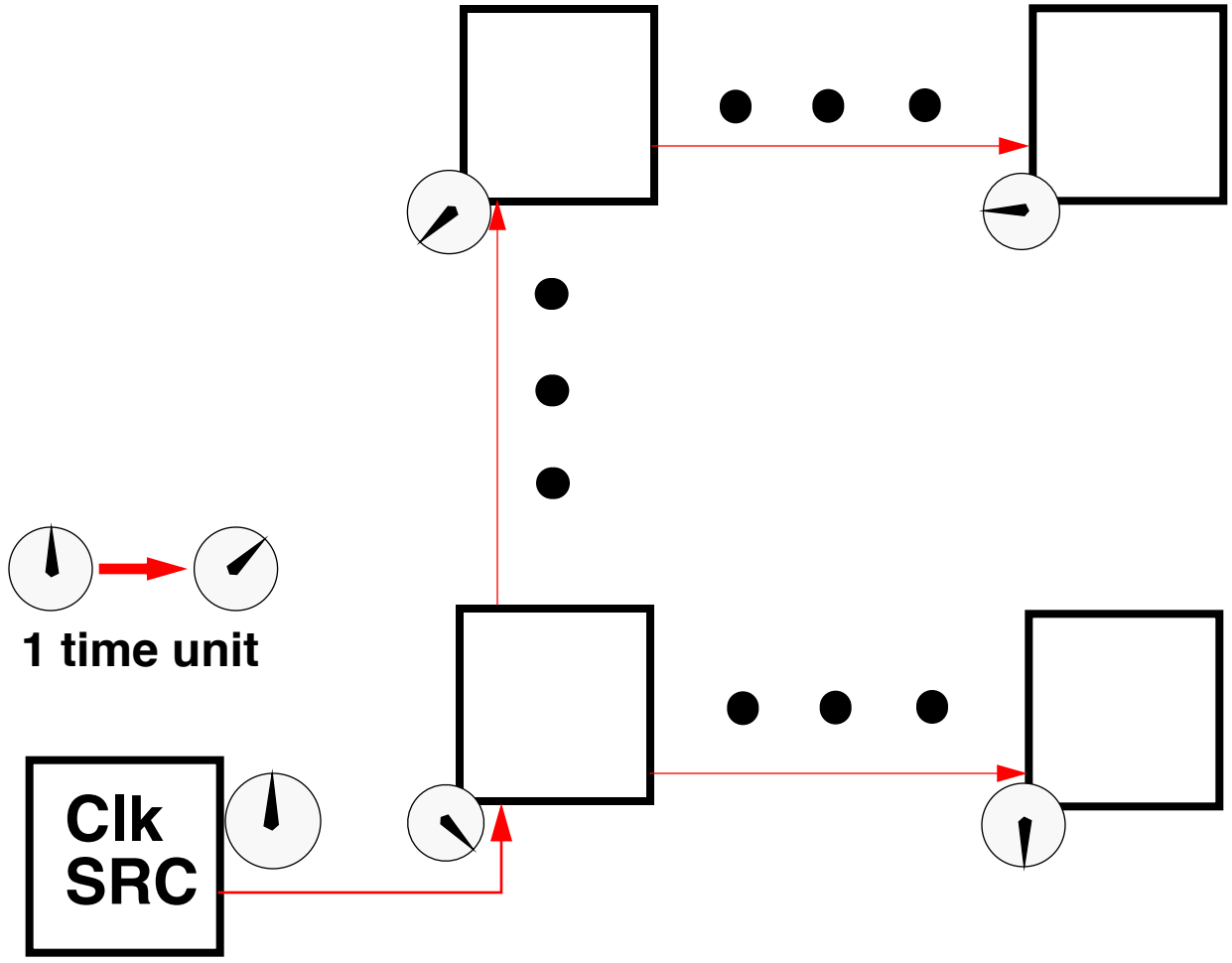


Bruce Jacob

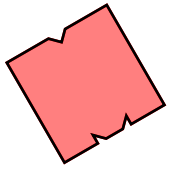
University of
Maryland
ECE Dept.



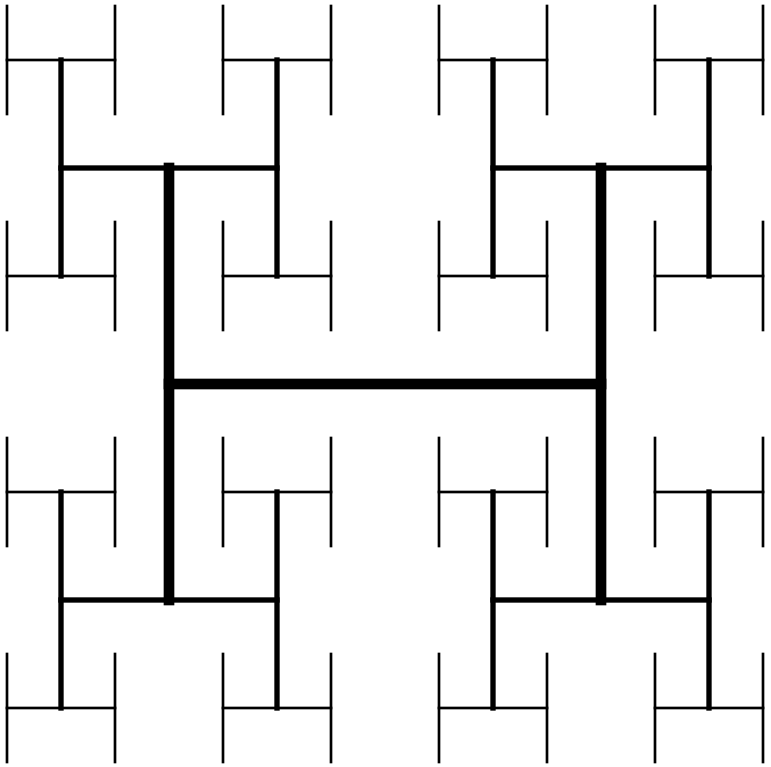
# End Result: *Clock Skew*




**In general, even nearby clocks not in synch**

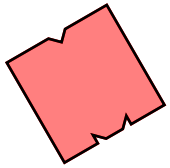


# Clock Tree I



**Clock distribution is a *major* design problem!**

- **Every branch sees same wire length and capacitance**
- **The clock skew is theoretically zero**
- **The sub-blocks should be small enough s.t. the skew within the block is tolerable**
- **Essential to consider clock distribution *early* in the design process**

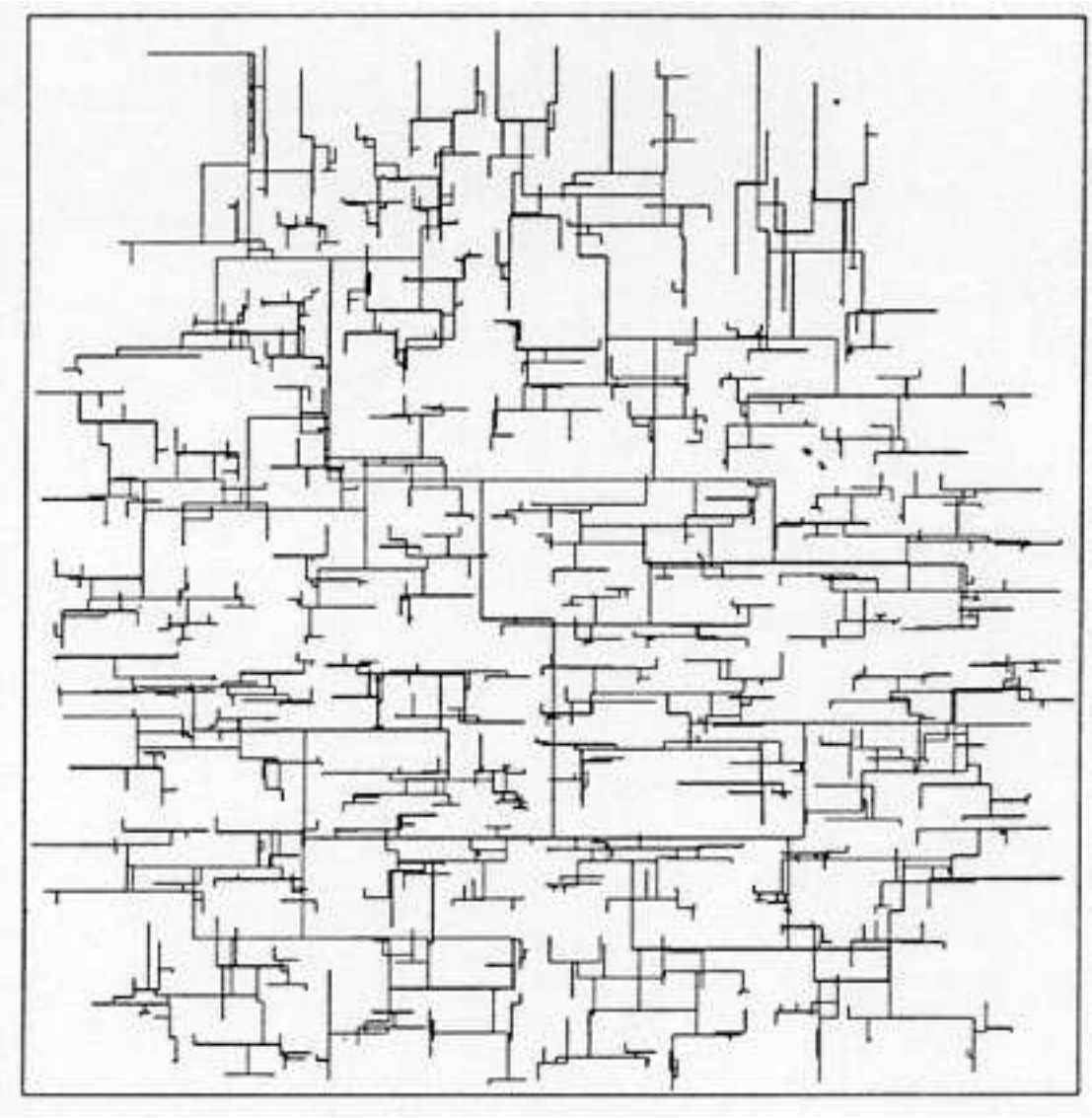


Bruce Jacob

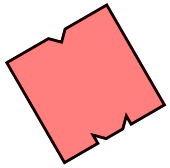
University of
Maryland
ECE Dept.



# Clock Tree I



**An on-chip clock tree**

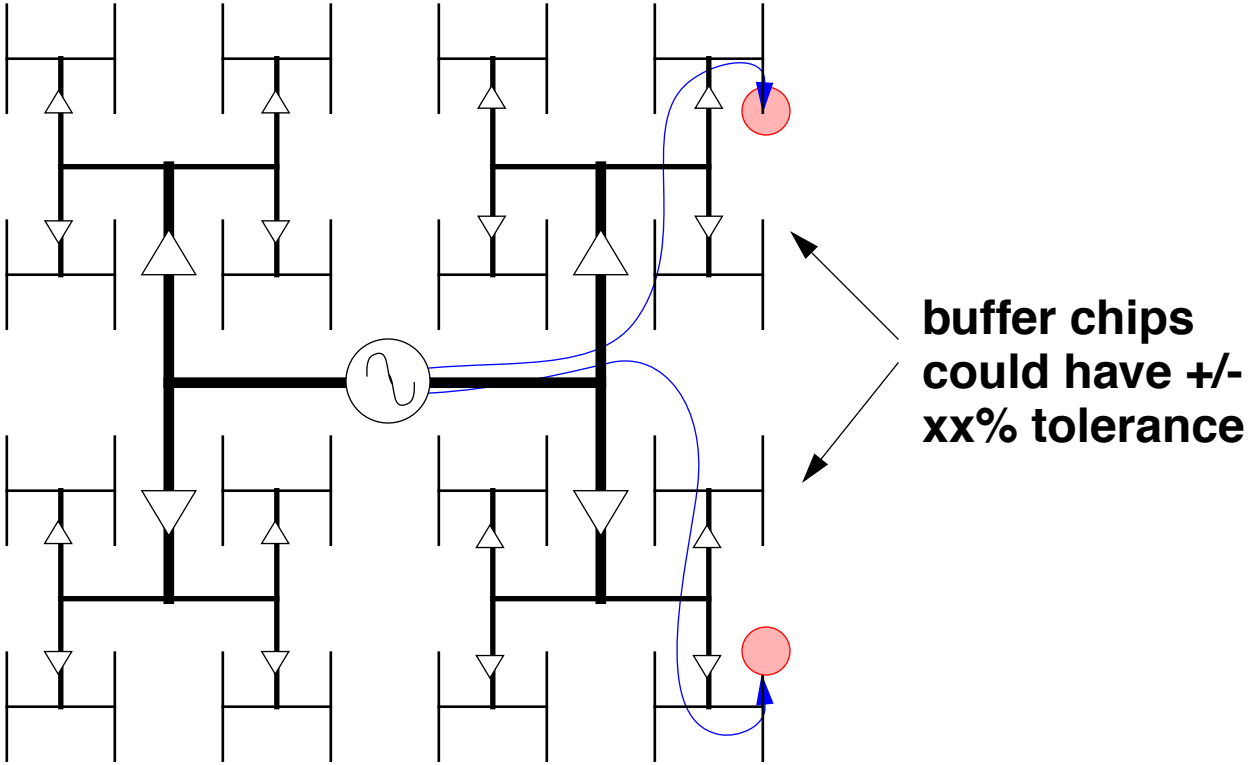


Bruce Jacob

University of
Maryland
ECE Dept.

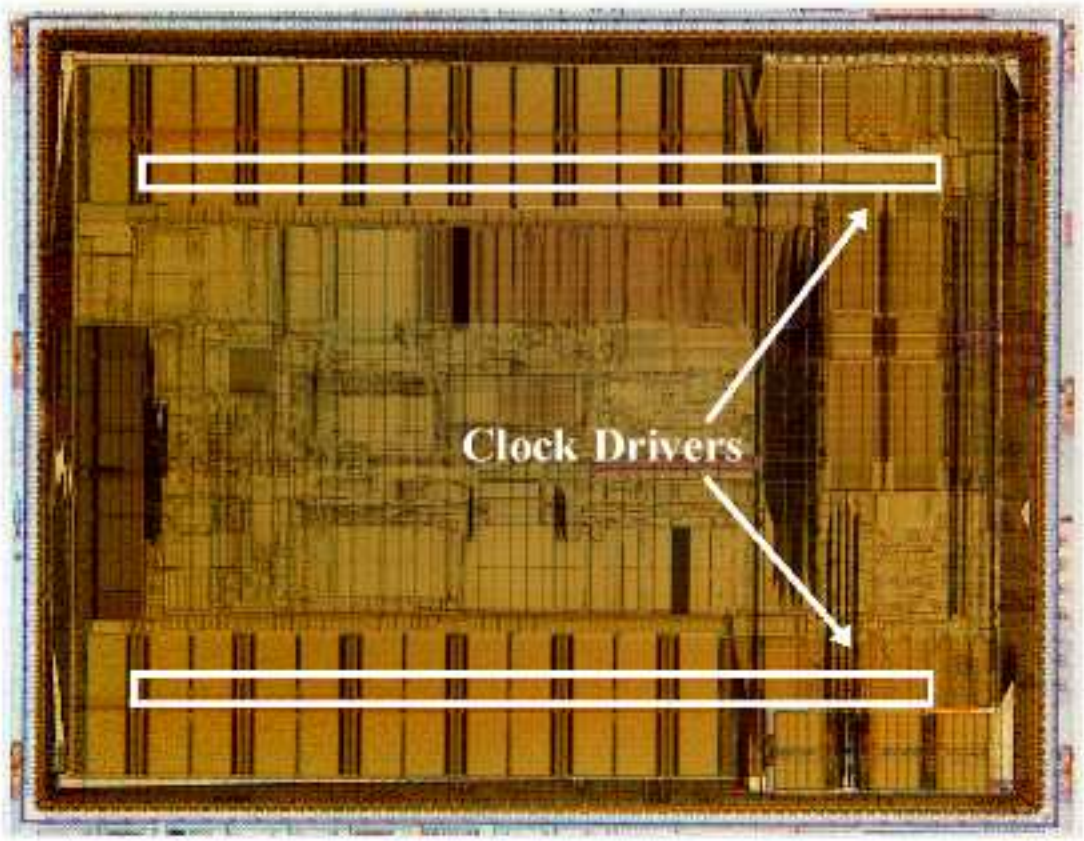


# Clock Tree II (I with buffers)

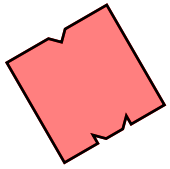



- Large synchronous systems require all components (chips or registers) to be driven by clock signal.
- Clock signal paths and buffers could introduce both skew and jitter at each stage
- Jitter and skew are additive with larger systems. More buffering, more skew and jitter.

# DEC Alpha 21164




**9.3 M Transistors, 4 metal layers, 0.55$\mu$m**
**Clock Freq: 300 MHz**
**Clock Load: 3.75 nF**
**Power in Clock = 20W (out of 50W)**
**Two Level Clock Distribution:**
**• Single 6-stage driver at center**
**• Secondary buffers drive left and right side**
**Max clock skew less than 100psec**
**• Routing the clock in the opposite direction**
**• Proper timing**

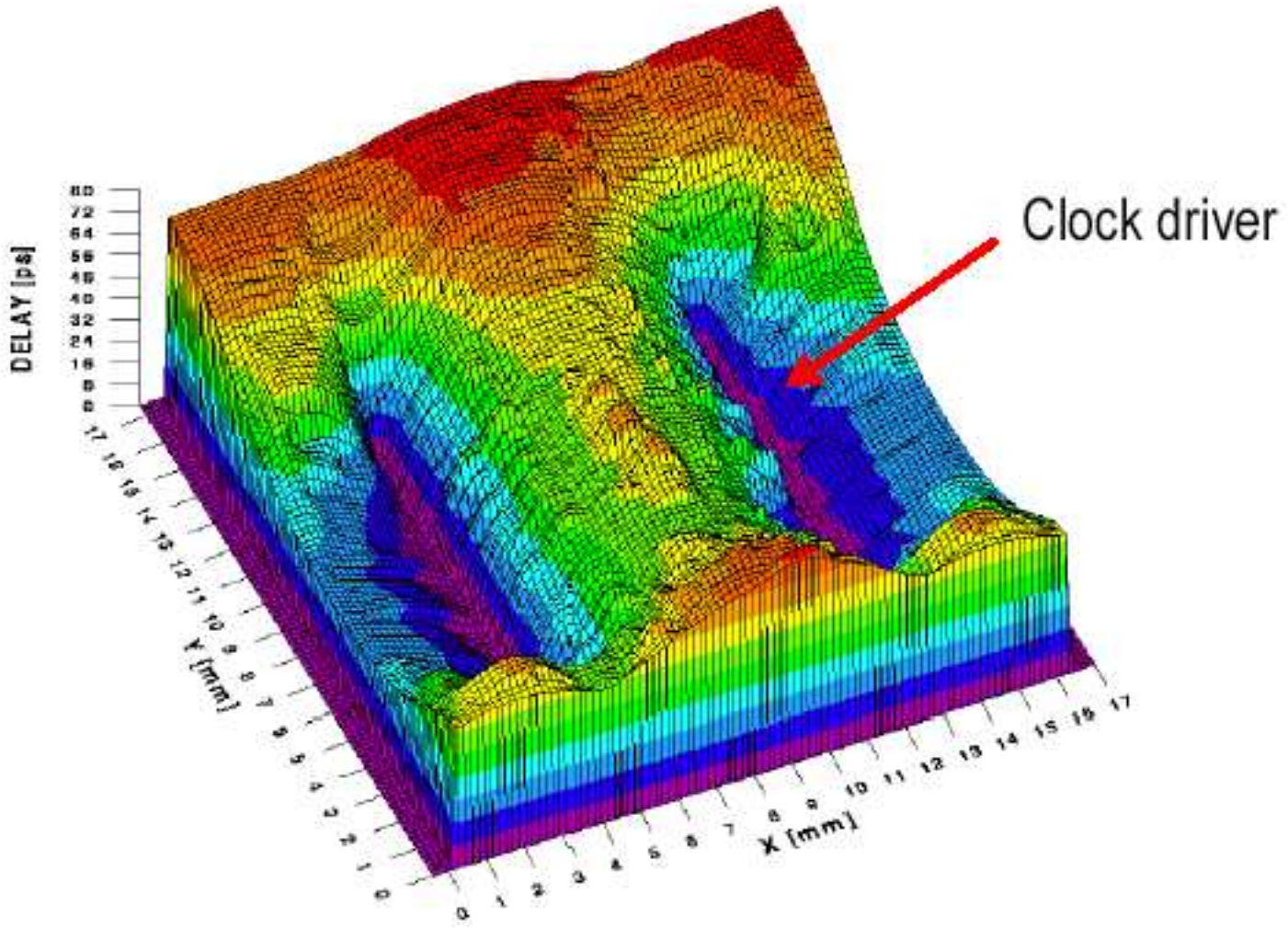


Bruce Jacob

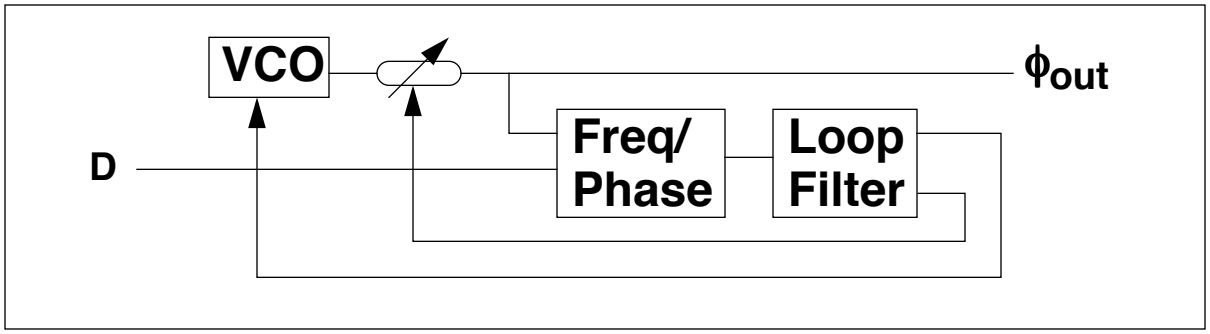
University of
Maryland
ECE Dept.



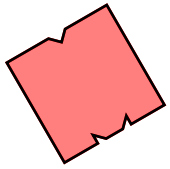
# Clock Skew in Alpha



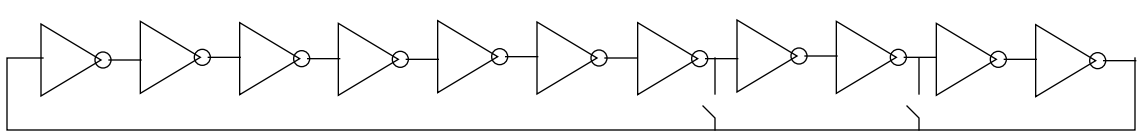
# Phase Locked Loop




- Given a data signal, recover the frequency and phase of the data signal, generate local reference clock $\phi_{out}$
- Local reference clock may be frequency multiple of input clock
- PLL depends on data input to provide "enough" signal transitions to lock onto, else PLL could lose coherency.
- Modern processors utilize PLL's for frequency multiplication

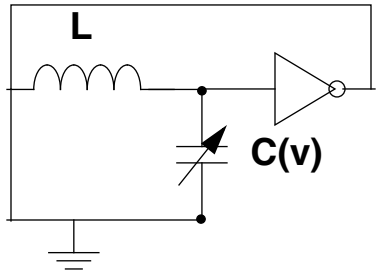


# Voltage Controlled Oscillator



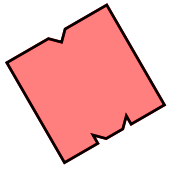
Ring Oscillator

- VCO may be designed from ring oscillator where voltage controls the number of (odd) stages of inverters in the feedback ring




LC Oscillator (conceptual illustration)

- VCO may be designed from resonant oscillator where voltage controls capacitance in LC circuit.

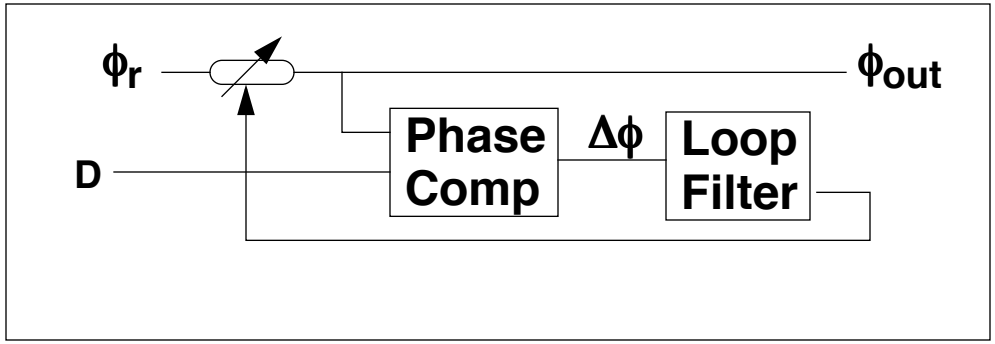


Bruce Jacob

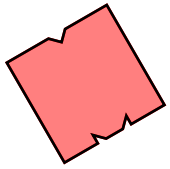
University of
Maryland
ECE Dept.



# Delay Locked Loop




- Given a data signal and reference clock, compare and adjust phase of local clock signal by $\Delta\phi$
- Unlike PLL, requires reference clock
- Hence, no need to "recover" clock signal with VCO
- Modern DRAM with dual edged clocking utilizes DLL's for phase compensation. (gets you 90 degrees)

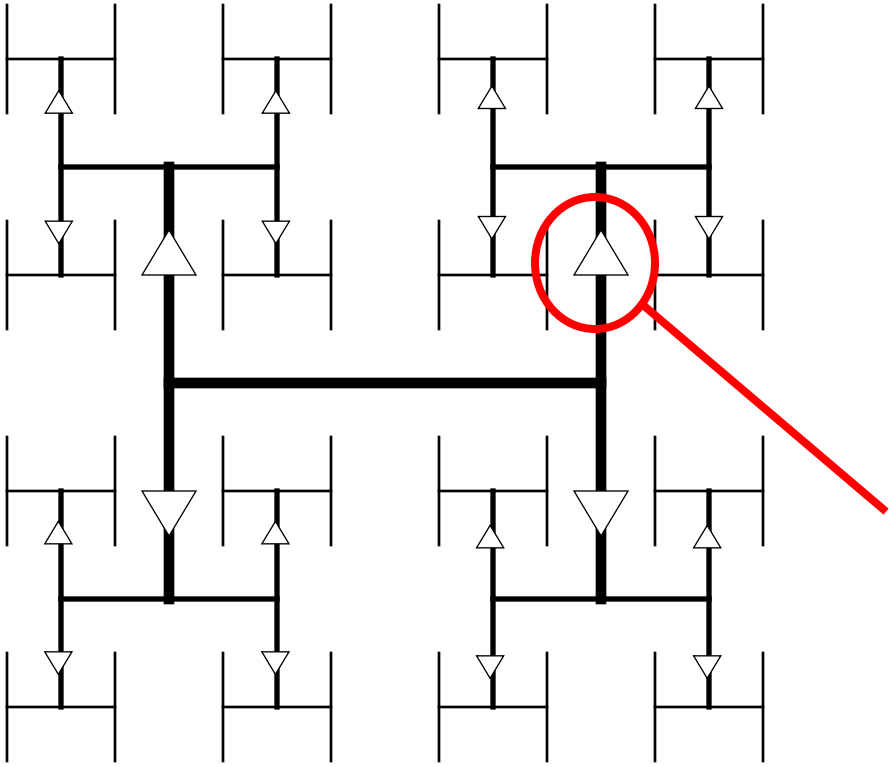


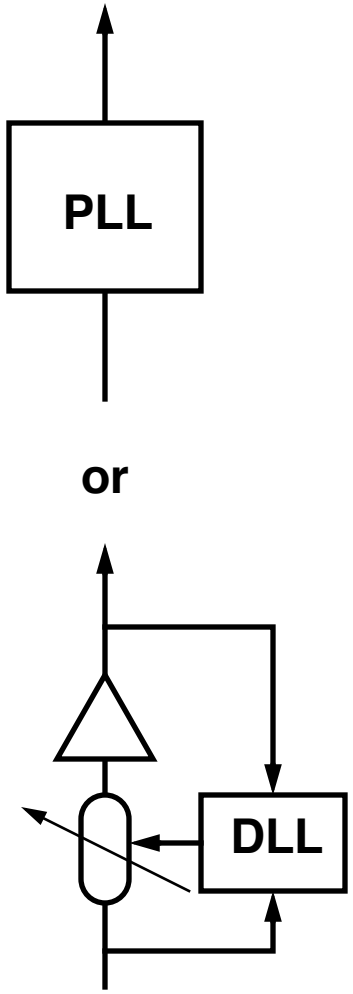
Bruce Jacob

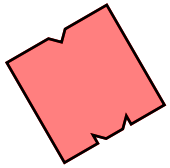
University of
Maryland
ECE Dept.



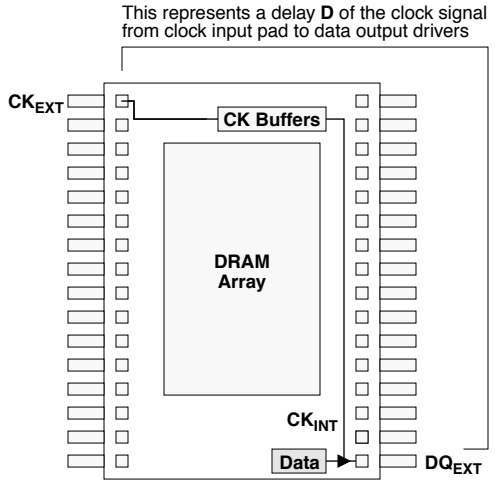
# Zero-Skew Clock Distribution

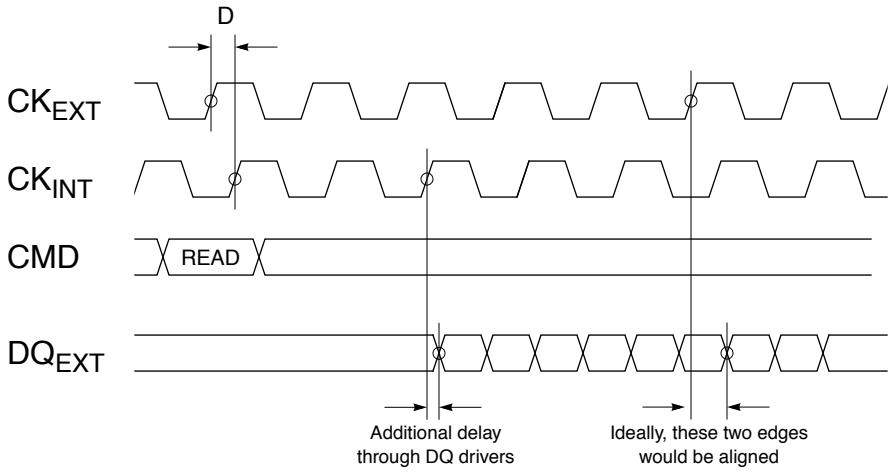


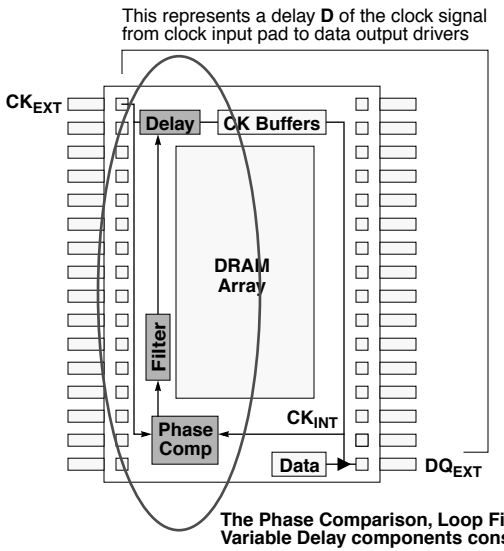


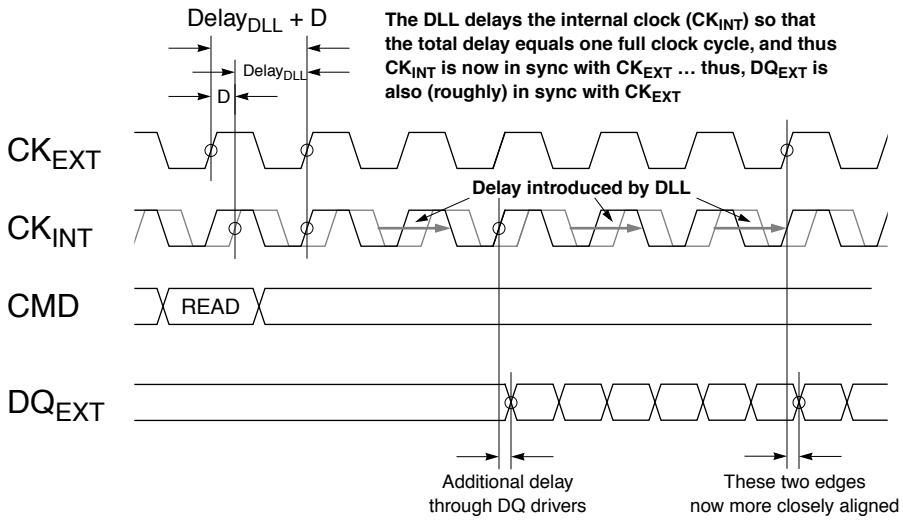
# DLL in DDR SDRAM



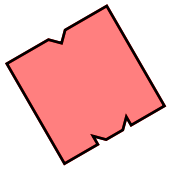



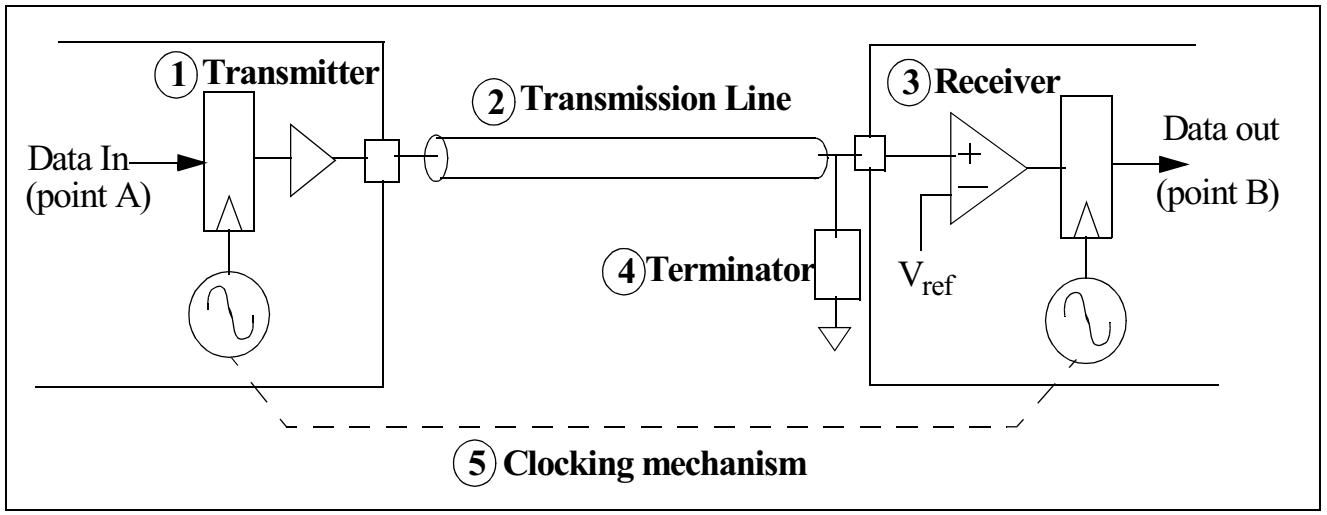

The Phase Comparison, Loop Filter, and Variable Delay components constitute a DLL




**The DLL delays the internal clock ($CK_{INT}$) so that the total delay equals one full clock cycle, and thus $CK_{INT}$ is now in sync with $CK_{EXT}$ … thus, $DQ_{EXT}$ is also (roughly) in sync with $CK_{EXT}$**



# A Signaling System




**1. Transmitter: Encodes data as current/voltage level onto the line**

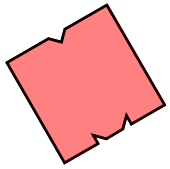
**2. Transmission Line: Deliver data from transmitter to receiver**

**3. Receiver: Compare against reference to extract data**

**4. Terminator: Remove signal from line, once they're received**

**5. Clock: Tells transmitter when to send, receiver when to sample signal**

* Poulton ISSCC 1999 Signaling Tutorial

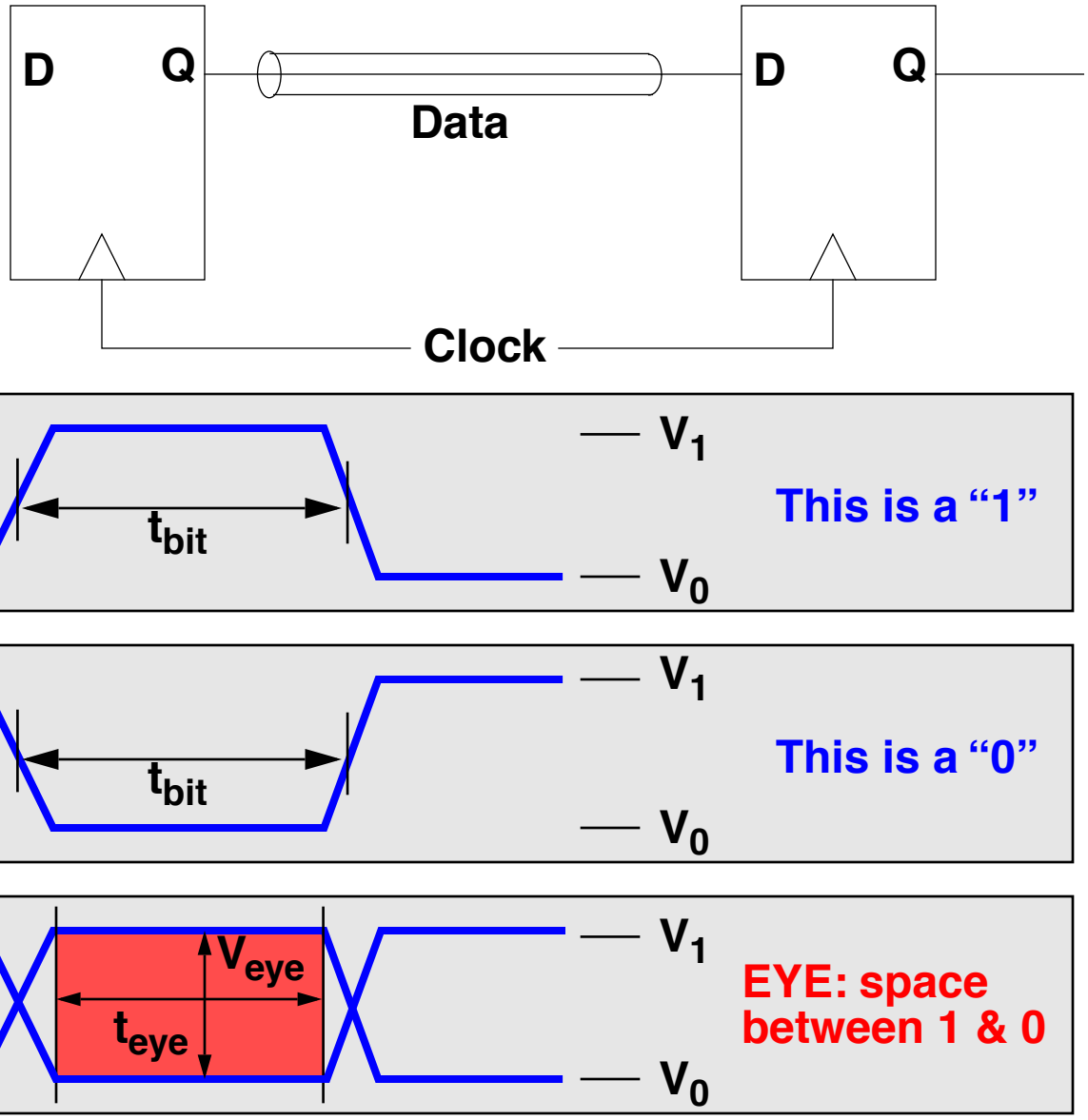


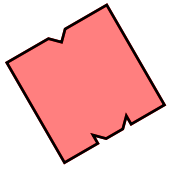
Bruce Jacob

University of
Maryland
ECE Dept.



# Idea: Detect “0” vs. “1”



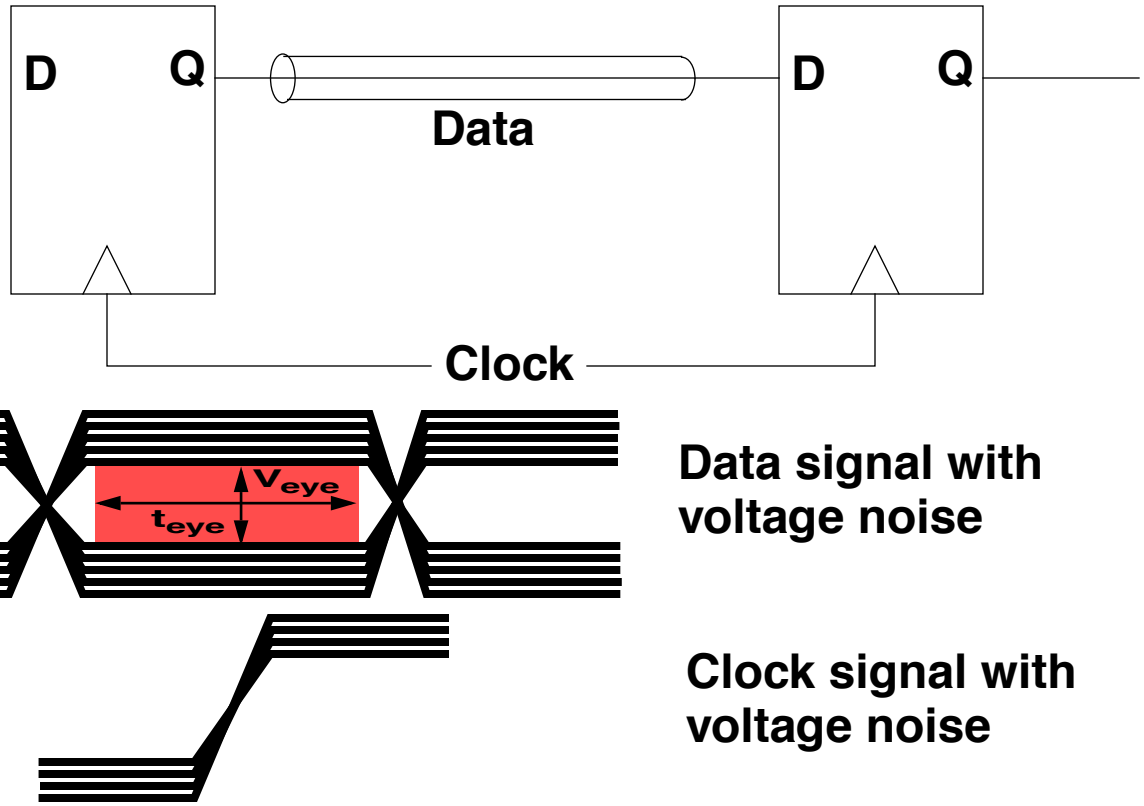



Bruce Jacob

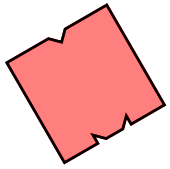
University of
Maryland
ECE Dept.



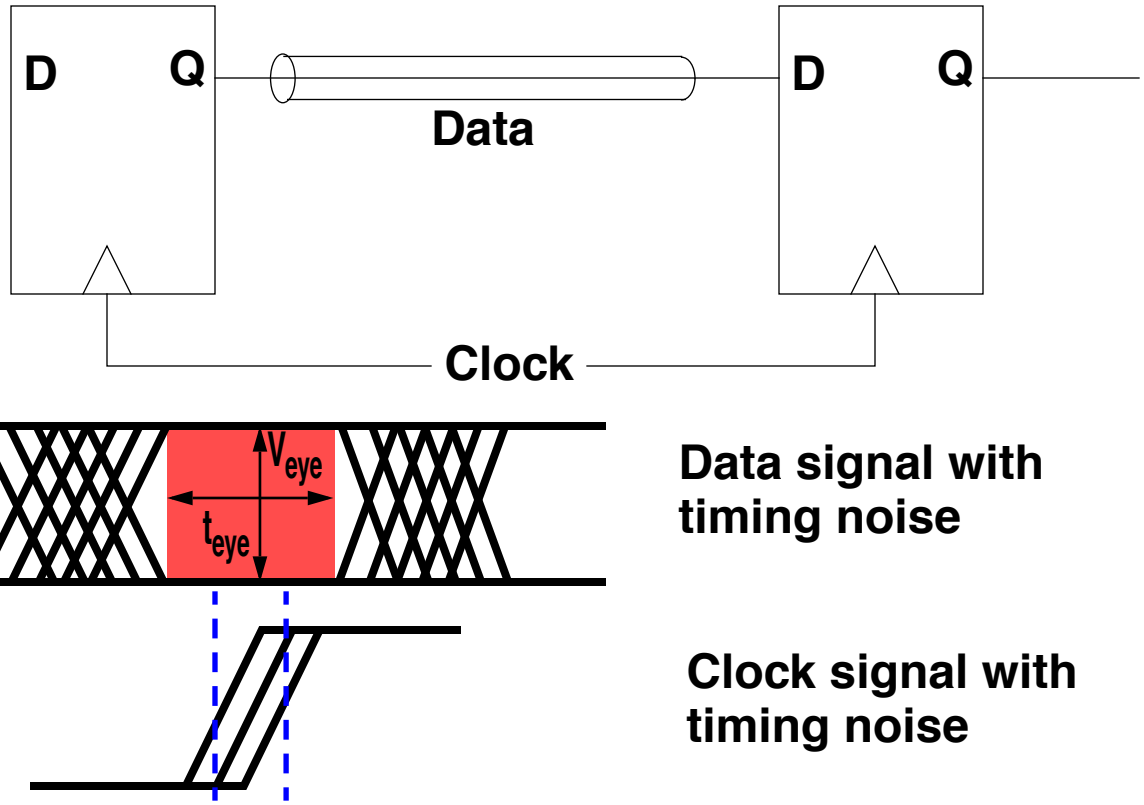
# Basic Problem: Voltage Noise




**Voltage noise reduces operating margins**

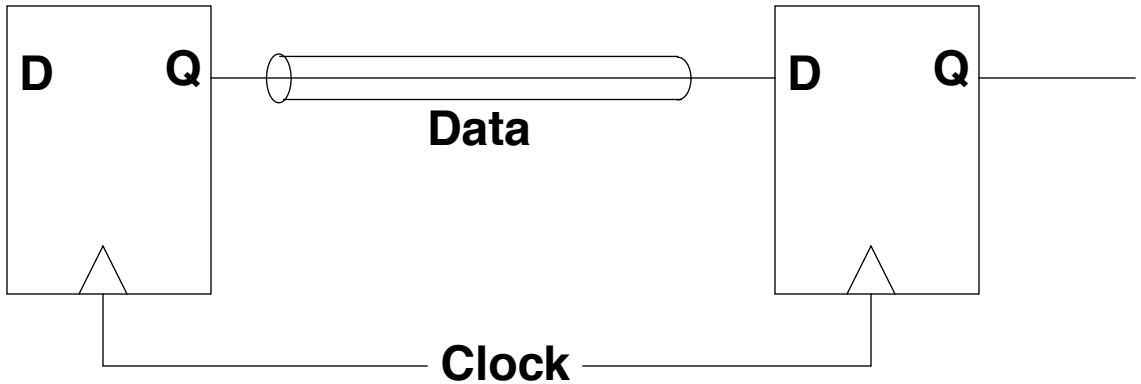


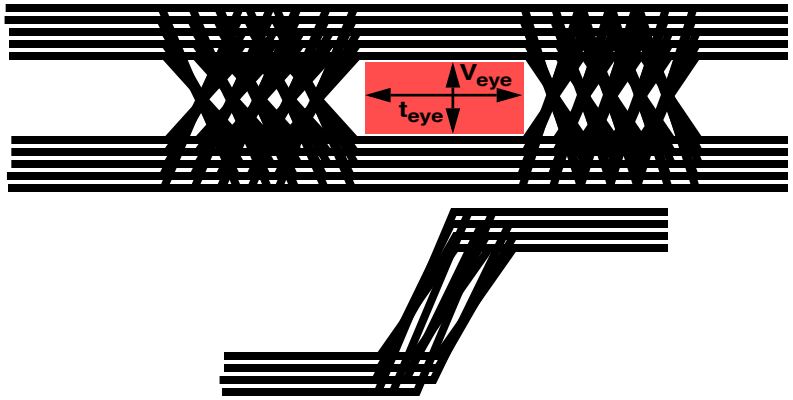
# Basic Problem: Timing Noise




**Timing noise reduces operating frequency**

# Basic Problem: Both

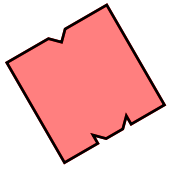


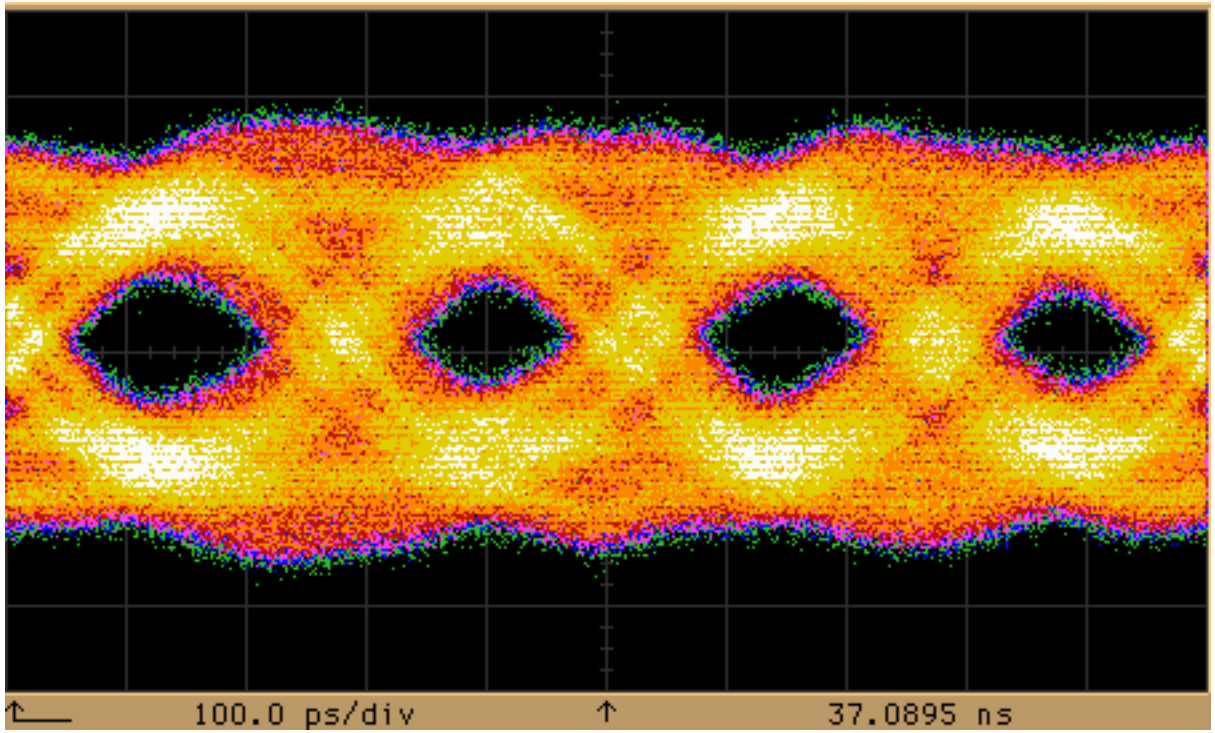

Data signal with both noise sources

Clock signal with both noise sources

**Note: Clock signal is just another signal subject to same constraints of voltage noise, skew and jitter as data signals**



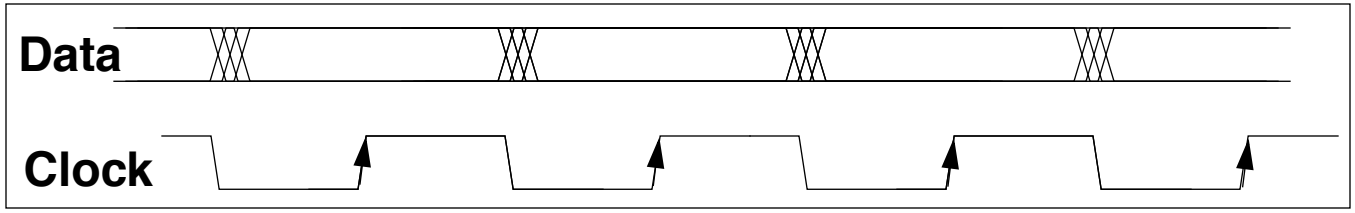
# Eye Diagram



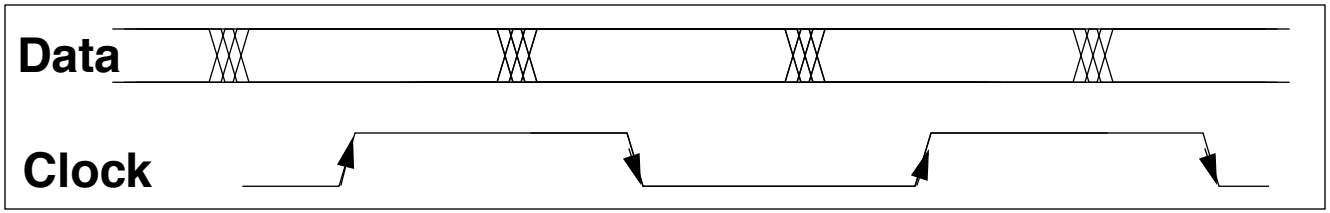

**Yes, there really is that much voltage noise;**
**Yes, there really is that much timing noise …**

**Life sucks; deal with it.**

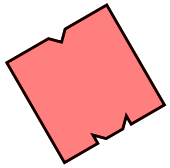
# Dual Edge Clocking



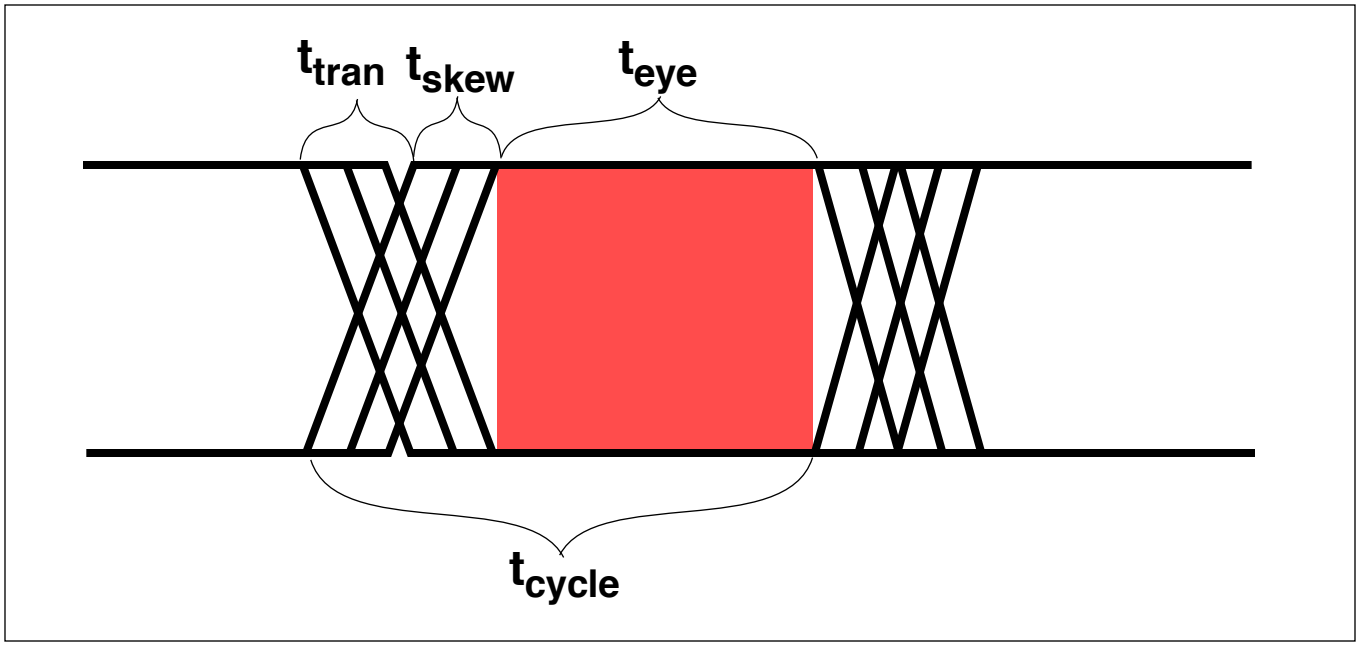

- Only one edge of clock latches data
- Duty cycle of clock signal is not relevent
- Clock signal operating at 2X switching rate of data
- Always a clock edge where you need one




- Both edges of clock used to latch in data
- Duty cycle & rise/fall times of clock must be even
- Clock signal must be phase shifted by 90 degrees relative to phase of data signal
- How do you get 90 degrees ??



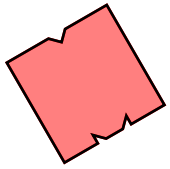
# Cycle Budget




$t_{skew}$ = max(skew) + max(jitter)

$t_{tran}$ = Edge transition time = max(rise_time, fall_time)

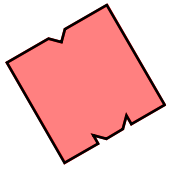
$t_{eye} \leq t_{cycle} - t_{tran} - t_{skew}$



# One "Typical" Timing Budget

| Timing Parameters | pS |
|---|---|
| Clock Skew on Transmit Side | 100 |
| Silicon on Transmit side | 1,000 |
| Wiring - drive to chip pad, drive module, card (40cm), receiver module, chip pad to receiver. | 3,000 |
| Silicon on Receive Side | 1,000 |
| Clock Skew on Receive Side | 100 |
| Coupling, Jitter, Reflections | 700 |
| PLL - Long term jitter | 250 |
| **Total** | **6,150** |

***Power4 Synchronous Wave Pipelined Interface Presentation, Hotchips '99**



# Timing: Clock ∩ Data

**Define *when* a value is present on a line …**

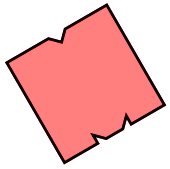
***How many 1's? How many 0's?***

**Need convention to distinguish where one '1' ends and next '1' begins … conventions typically mark boundaries w/ *TRANSITIONS***

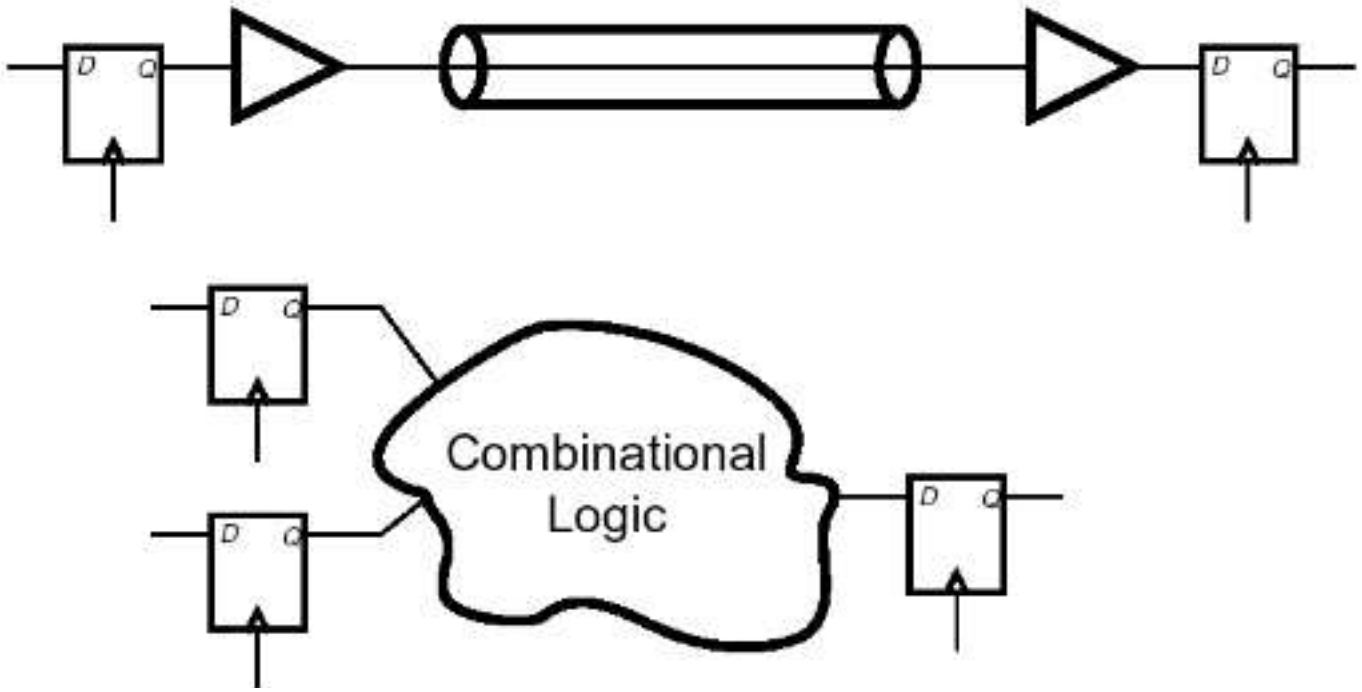
- **of the signal itself**
- **of an associated *clock* signal**

**(original definition of "synchronous")**

**Uncertainty in timing limits operating speed**



# Timing Conventions

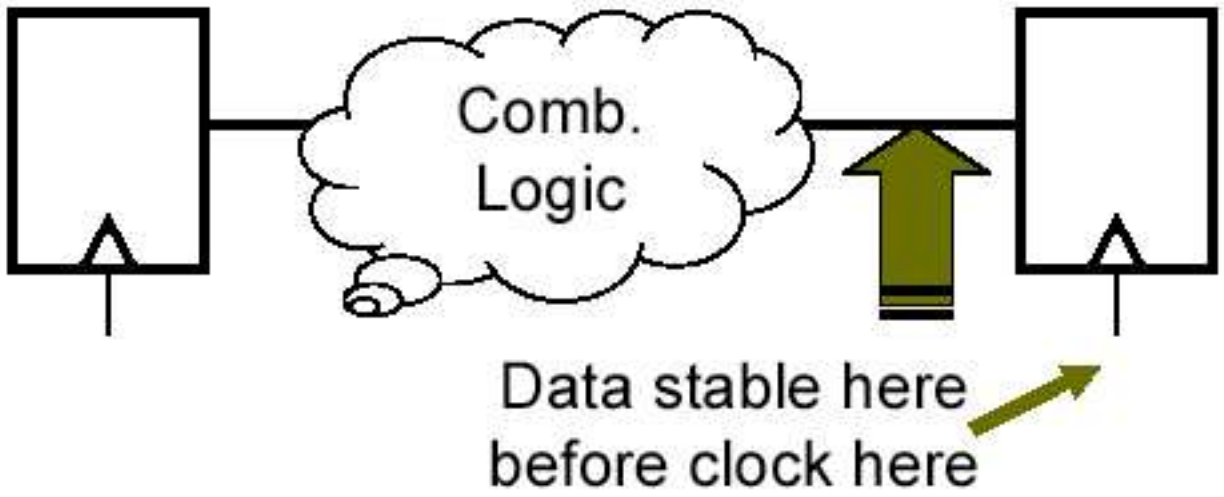



**Apply to both inter- *and* intra-chip signaling**

# Setup Time

## Required time for input to be stable

## BEFORE CLOCK EDGE

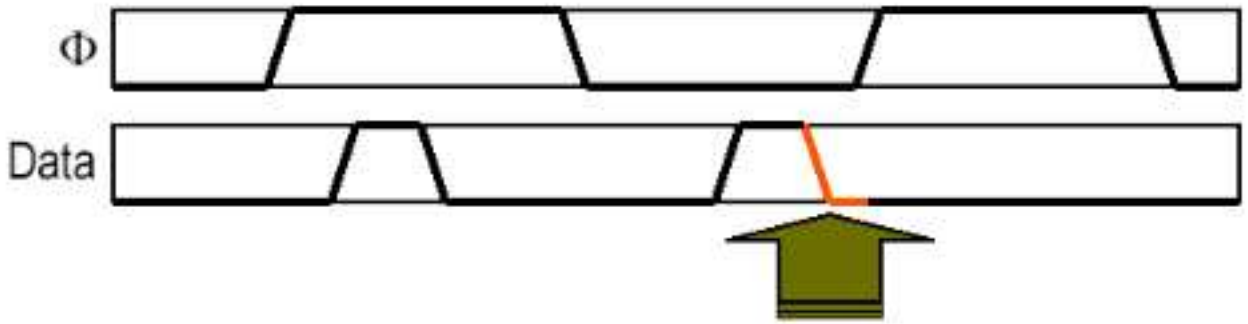


Bruce Jacob

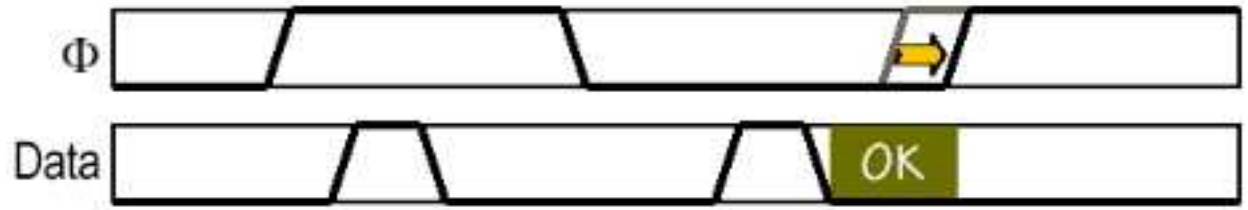
University of
Maryland
ECE Dept.

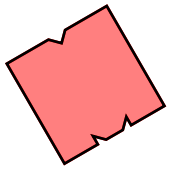


# Setup Time Fix




This violation can be fixed by stretching the clock cycle



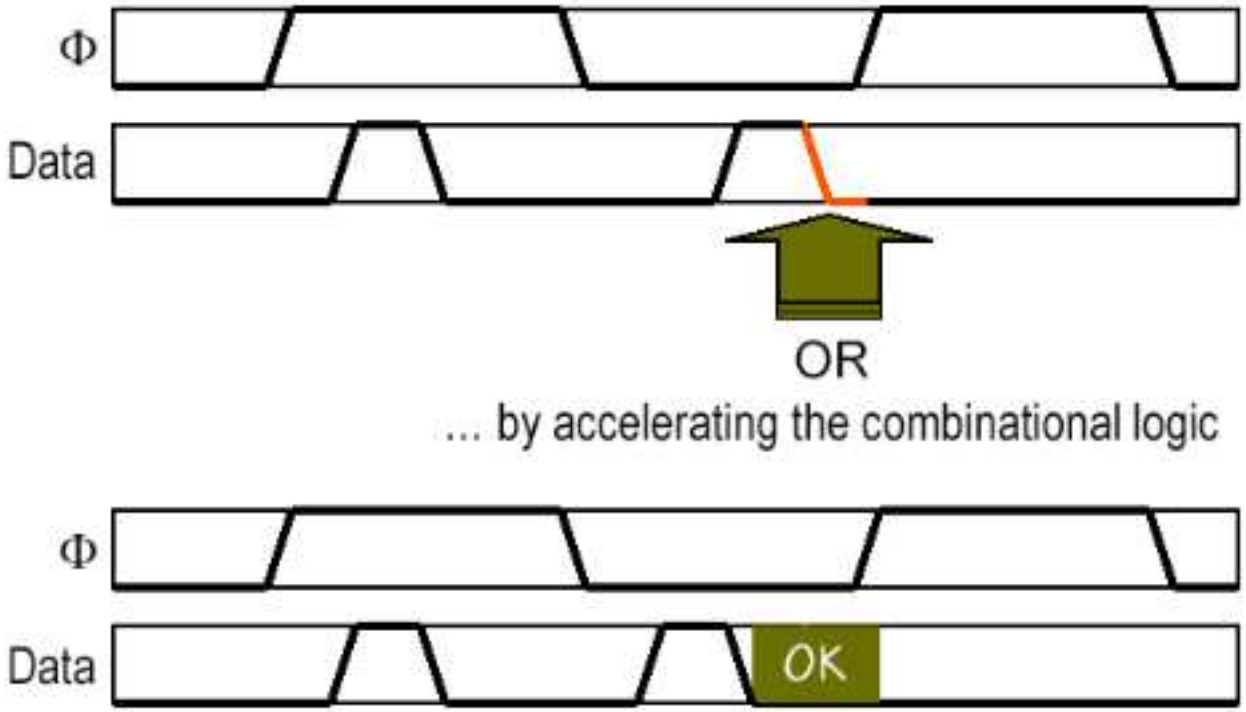



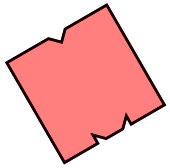
Bruce Jacob

University of
Maryland
ECE Dept.



# Setup Time Fix II

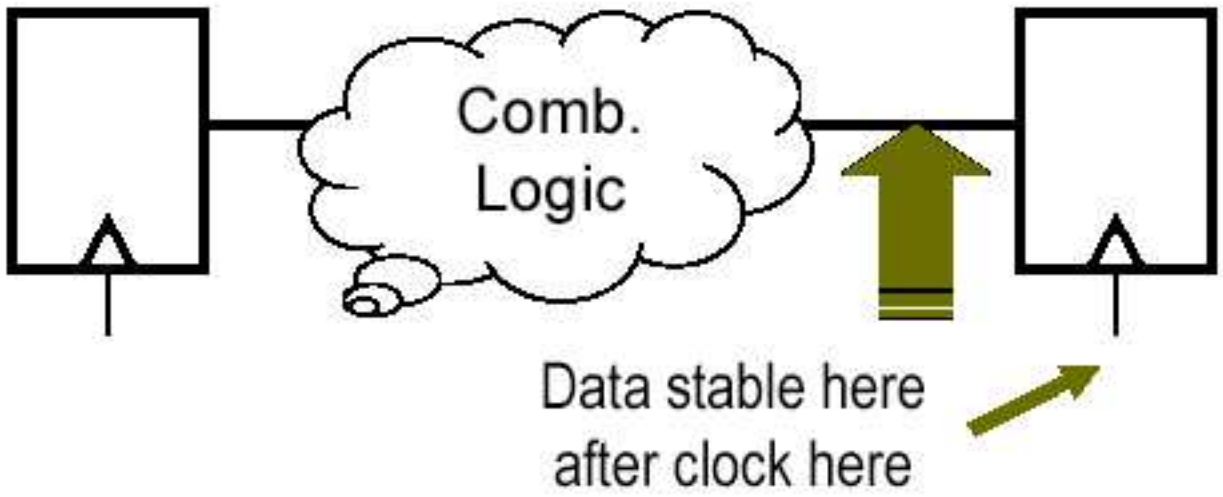


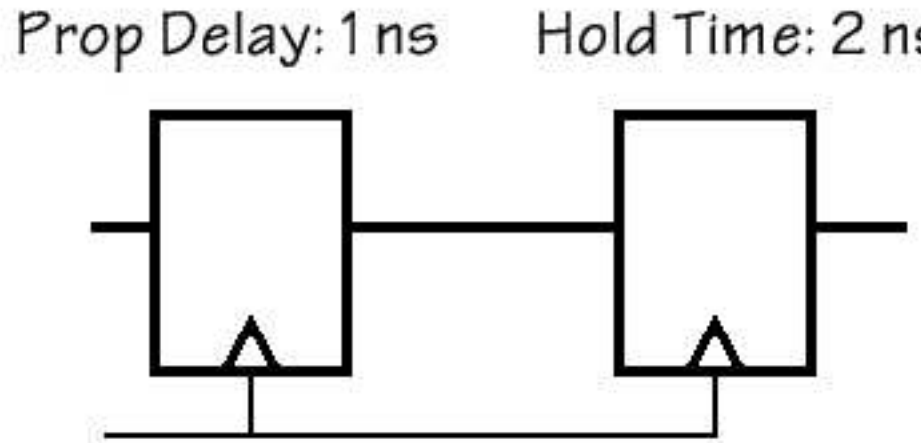
# Hold Time

## Required time for input to be stable

## AFTER CLOCK EDGE



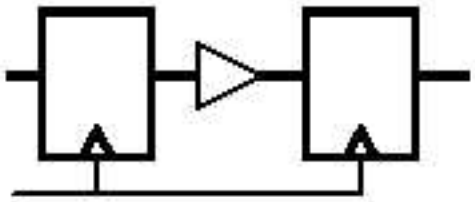
# Hold Time Violations




Hold time violations are caused by "short paths"

*Cannot be fixed by slowing down the clock!!!*

Fixed by slowing down fast paths



Bruce Jacob

University of
Maryland
ECE Dept.



# A Tale of Two (or more) Timing Conventions

**Synchronous: global clock**

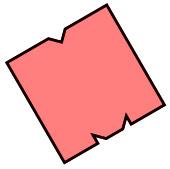
**Synchronous: source-synchronous I**

***("open-loop" meaning no control loop)***

**Synchronous: source-synchronous II**

***("closed loop" meaning feedback control)***

~~**Asynchronous: self-timed**~~

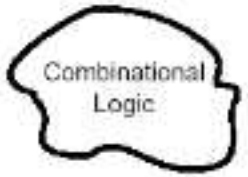


# System Building Blocks

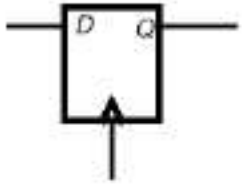
## DELAY ELEMENTS

- Nominal delay
- Timing uncertainty, skew, jitter

## COMBINATIONAL LOGIC

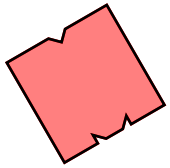



- Contamination delay
- Propagation delay



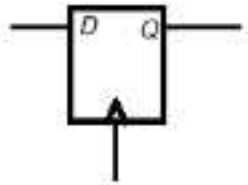

## CLOCKED STORAGE ELEMENTS

- Align signal to a clock
- Signal waits to be sampled by clock
- Output held steady until next clock

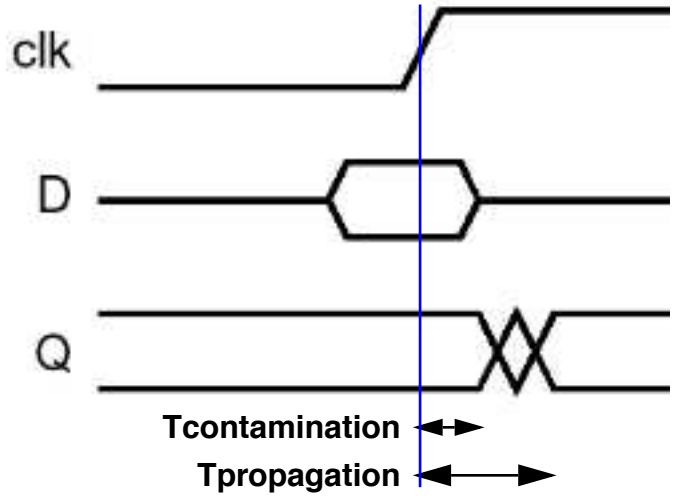


# Edge-Triggered Register

**Samples data on rising edge of CLK**

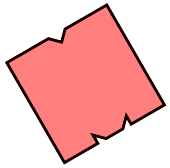



- **Data must remain valid during an *aperture* of time during sampling**



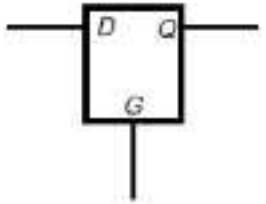

**Output held steady until next CLK edge**

- **Output is held until a *contamination delay* following CLK edge**
- **Output has a correct value after a *propagation delay* following CLK edge**

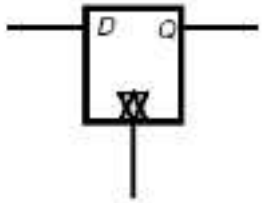


# Other Storage Elements

## LEVEL-SENSITIVE LATCH

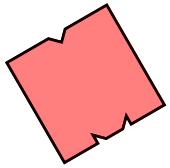



- Passes data through when enable (clock) is high
- Holds data stable when enable (clock) is low




## DUAL-EDGE-TRIGGERED FLIP-FLOP/REGISTER

- Samples data at both edges of the clock
- Internally two interleaved flip-flops (1 posedge FF + 1 negedge FF)
- Allows CLK to run at same speed as data

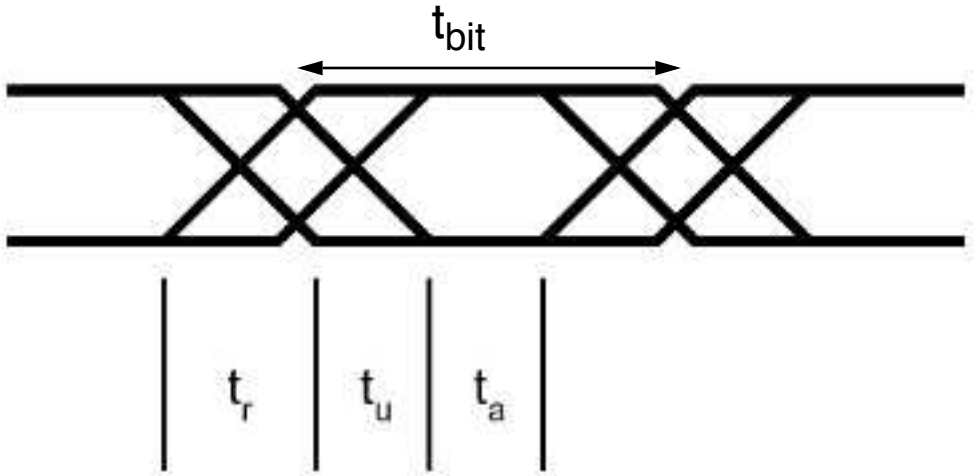


# Eye Diagram, Redux

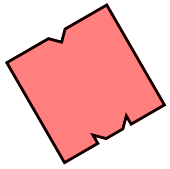
**MAXIMUM OPERATING RATE**

**(i.e., signaling speed)**

**LIMITED BY THREE FACTORS:**

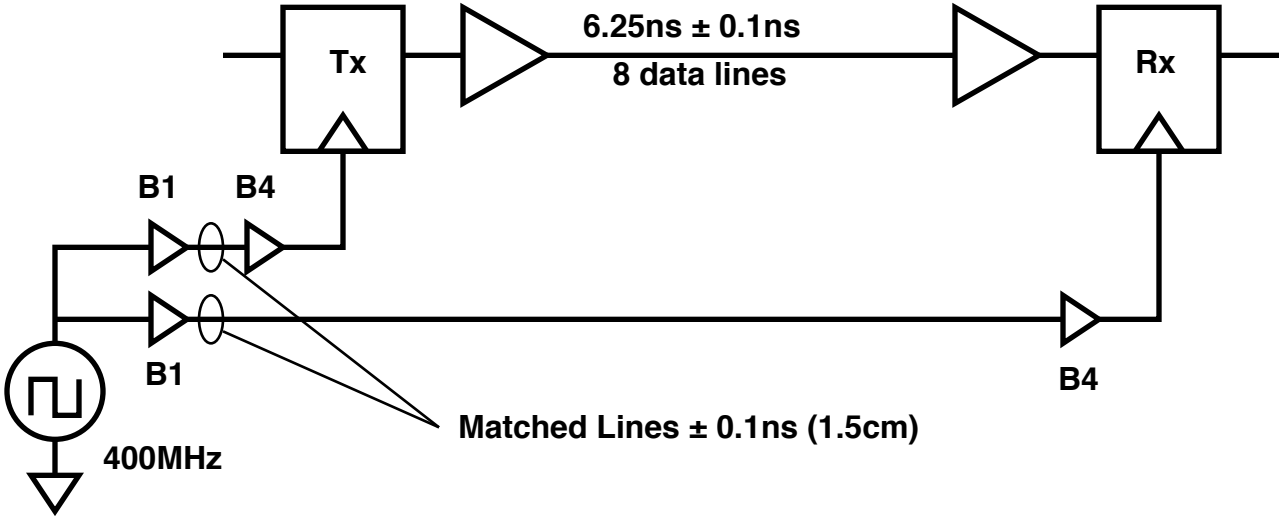



- **Tr — Transition time (rise/fall time)**
- **Tu — Timing uncertainty, skew, jitter**
- **Ta — Aperture time**
- **Tbit ≥ Tr + Tu + Ta *(but not that simple …)***

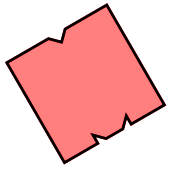


# Synchronous Timing I

## GLOBAL CLOCK (Conventional) EXAMPLE

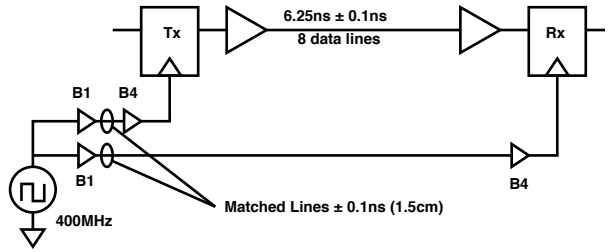



| Parameter | Symbol | Nominal | Skew | Jitter |
|---|---|---|---|---|
| **Bit Cell (data period)** | Tbit | 2.5 ns | | |
| **Transmitter Rise Time** | Tr | 1.0 ns | | |
| **Cable Delay** | Twire | 6.25 ns | 100 ps | |
| **Receiver Aperture** | Ta | 300 ps | 100 ps | 50 ps |
| **Transmitter Delay** | | 500 ps | 150 ps | 50 ps |
| **Buffer Stage Delay** | B# | 250 ps | 100 ps | 50 ps |



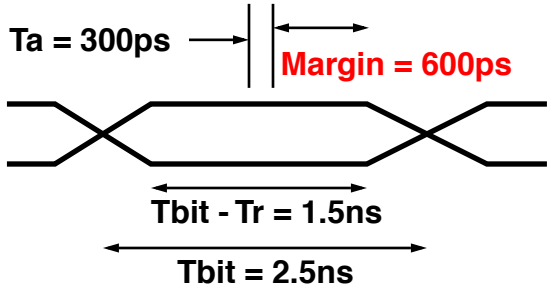
# Synchronous Timing I

## Sources of uncertainty:

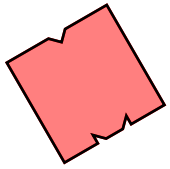



- **Skew** (across multiple lines) of line delay
- **Jitter** of Tx, Rx, and line delay
- **Skew** and **jitter** of global clock (usually large due to high fan-out)

For best performance, center sampling edge on data eye

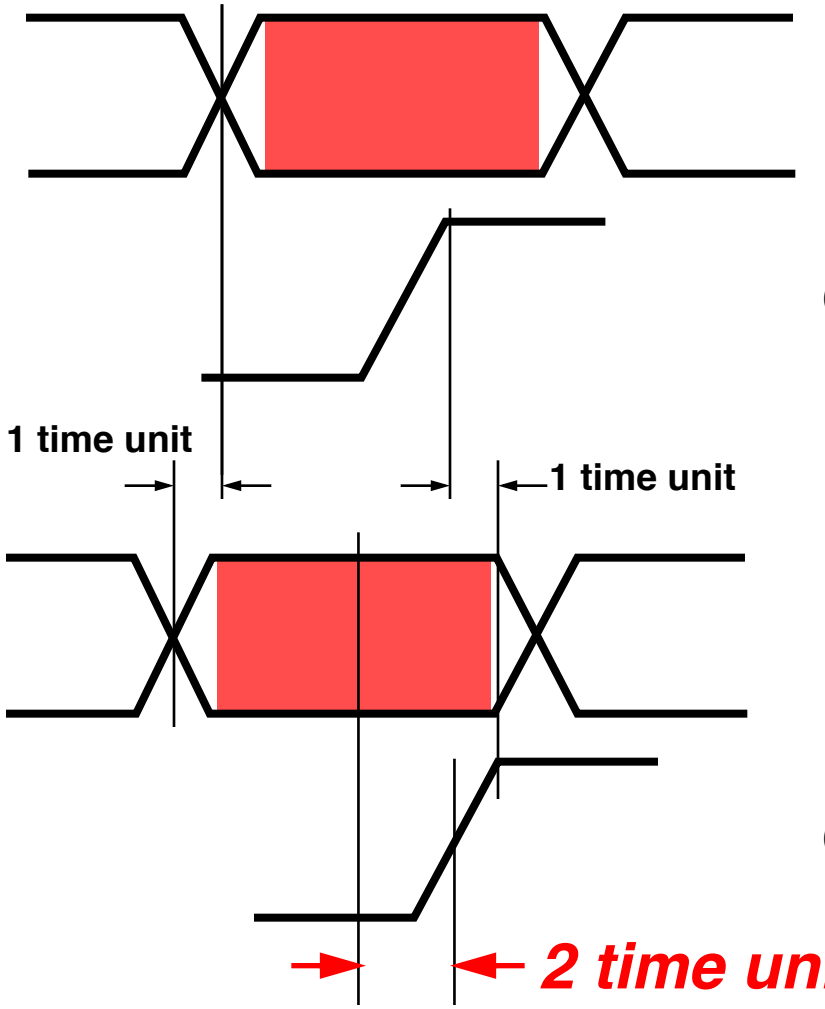



**Gross timing margin**
**= 1/2 [ Tbit - Tr - Ta] = ± 600ps = *1/2 Tu?***



# An Aside …

***Why do we simply add up the uncertainties?***
***And why does each effectively count twice?***




**DATA w/ no skew/jitter**

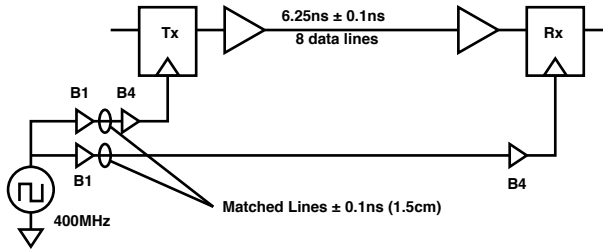
**CLOCK w/ no skew/jitter**

**DATA *early* by 1 time unit**

**CLOCK *late* by 1 time unit**

# Synchronous Timing I

## TIMING ANALYSIS




**Clock Skew: 100ps lines + 100ps B1 + 400ps B4**

**Clock Jitter: 50ps B1 + 200ps B4**

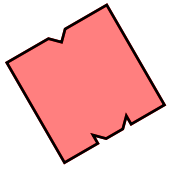
**[CLKs times 2: one for xmit, one for recv]**

**Transmitter: 150ps skew, 50ps jitter**

**Receiver: 100ps skew, 50ps jitter**

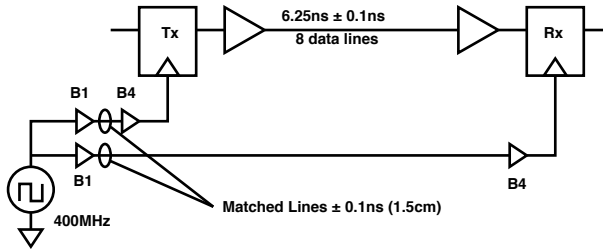
**Data Cable: 100ps skew**

**TOTAL: 1550ps skew, 600ps jitter (BAD)**

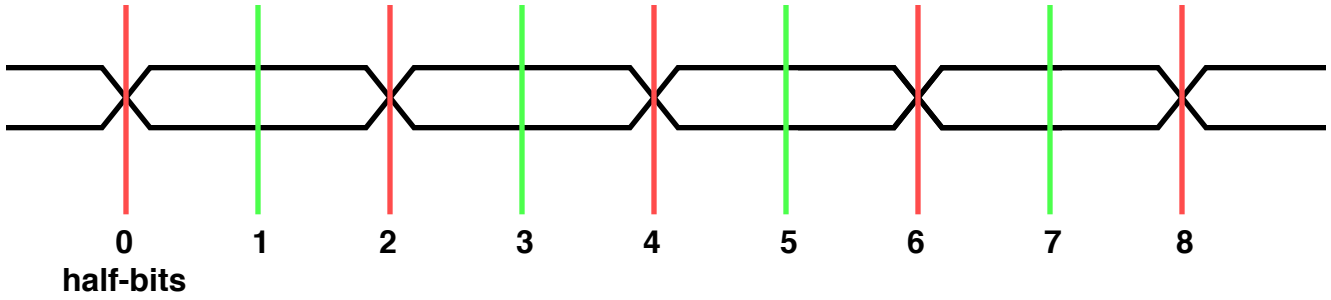


# Synchronous Timing I

## LIMITS TO LINE DELAY & DATA FREQUENCY



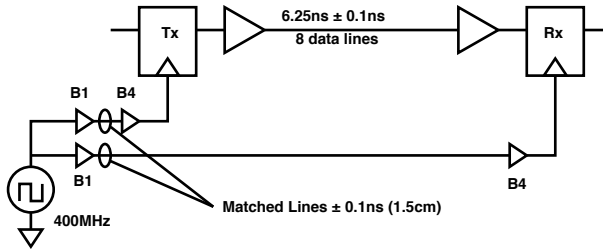

**Conventional Wisdom:**
**Line Delay Must Be**
**ODD NUMBER of HALF-BITS … WHY?**




**For fixed line length and tight margins, this limits the bus speeds that can be used**

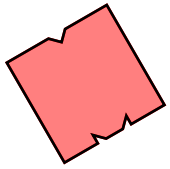
# Synchronous Timing I

## SUMMARY



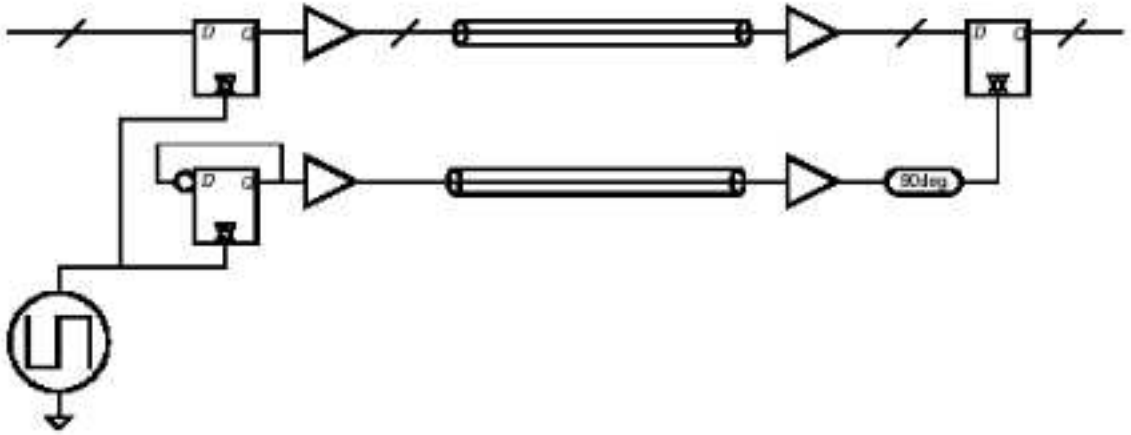

## GLOBALLY SYNCHRONOUS DESIGN:

- For long wires and high speeds, only a handful of frequencies work
- Impractical to control uncertainties
- Cannot switch frequencies



# Synchronous Timing II

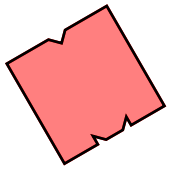
## PIPELINED TIMING: BASIC IDEA




**Delay the clock by the same amount as data … PLUS half a bit-cell**

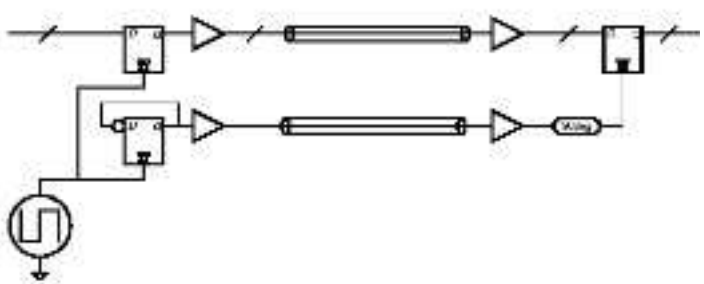
**System will work from DC to maximum theoretical frequency 1/(Tr + Tu + Ta)**

**Defines new clock domain at receiving end**



# Synchronous Timing II

## SOURCES OF UNCERTAINTY

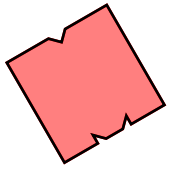


## SKEW:

- Between CLK & Data line
- Fixed differences in FF, Tx, Rx delays
- Different CLK delays to different FFs
- Aperture offset in Rx FF
- Extra offset in the delayed CLK line

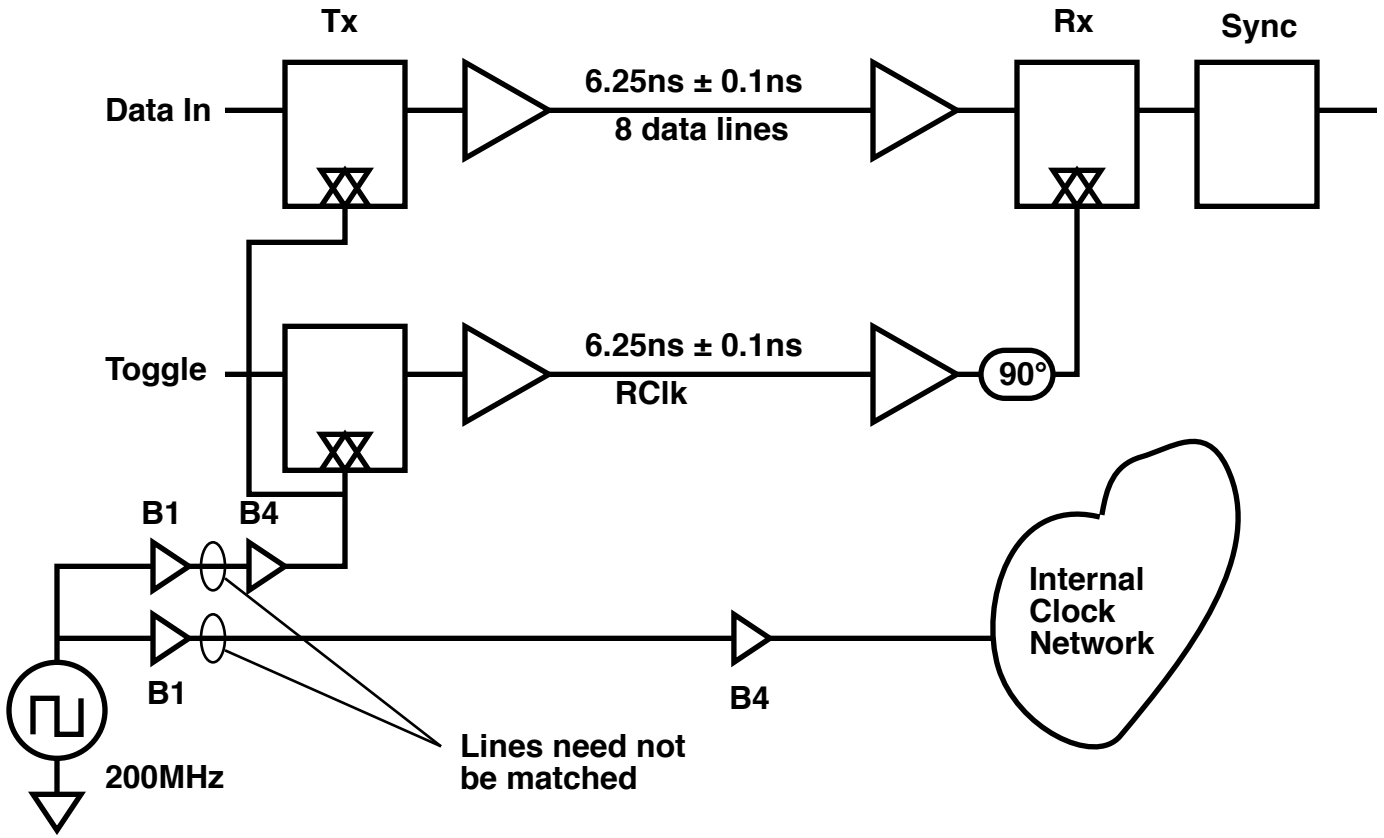
## JITTER:

- In Tx clock
- In FF, Tx, Rx delays



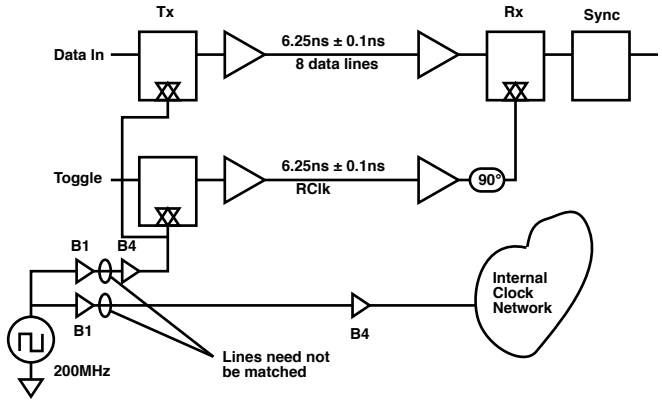
# Synchronous Timing II

## OPEN-LOOP PIPELINED EXAMPLE



# Synchronous Timing II

## TIMING ANALYSIS




**Xmit data: 150ps skew, 50ps jitter**

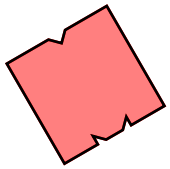
**Xmit toggle: 150ps skew, 50ps jitter**

**Receiver: 100ps skew, 50ps jitter**

**Data cable: 100ps skew**

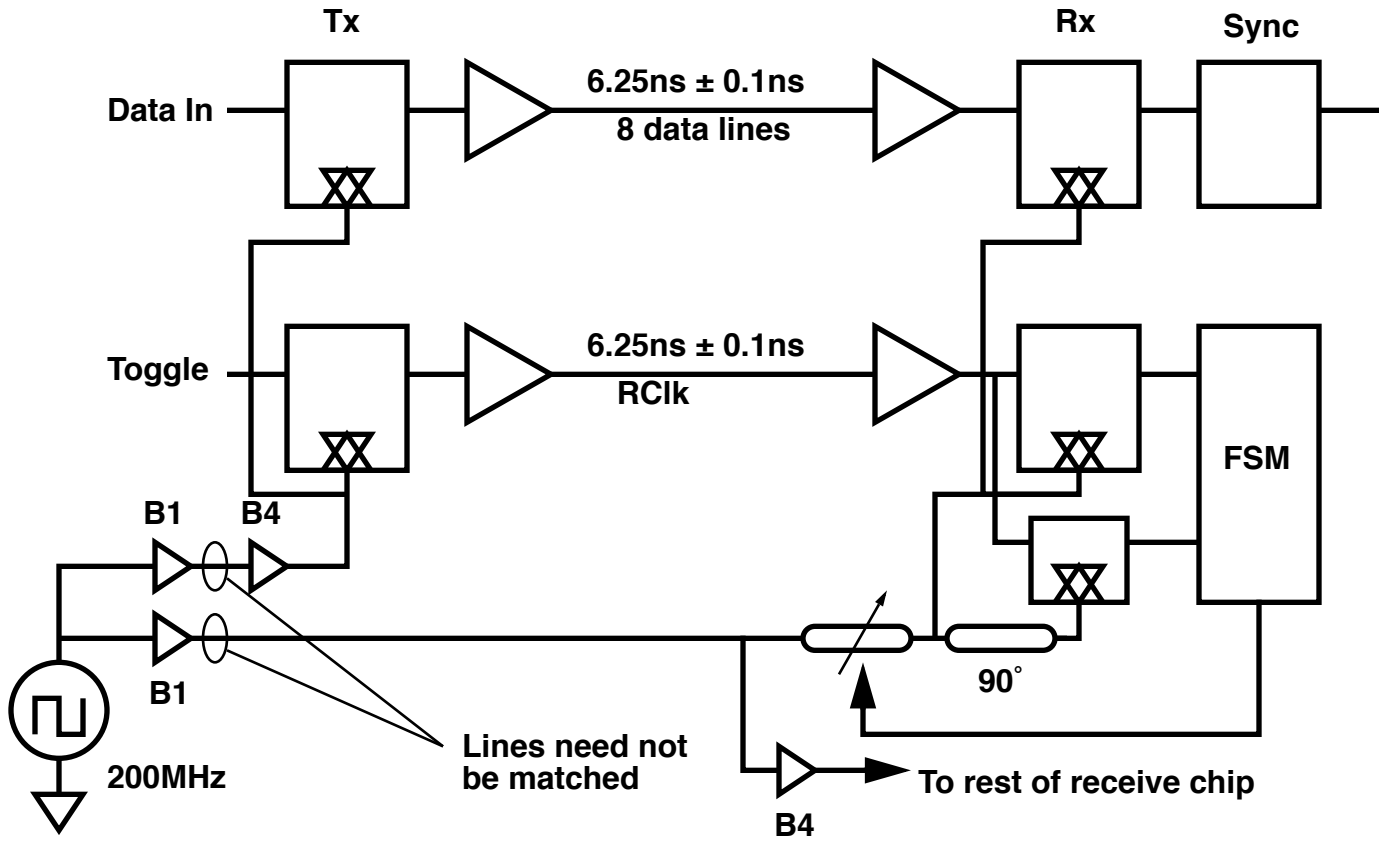
**Toggle clock cable: 100ps skew**

**TOTAL: 600ps skew, 150ps jitter (BETTER)**



# Synchronous Timing III

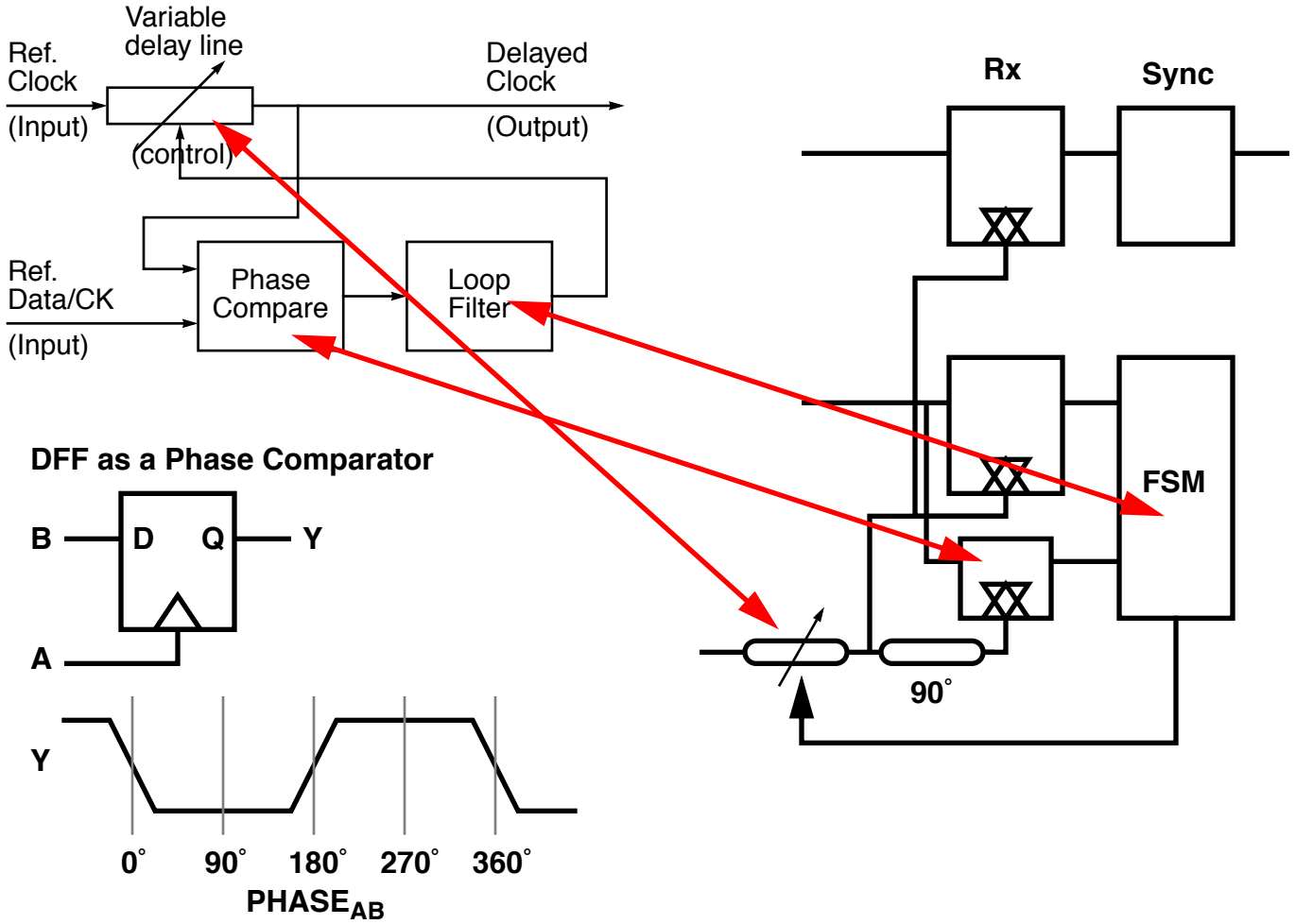
## CLOSED-LOOP PIPELINED EXAMPLE




**Variable delay line can cancel (almost) ALL SKEW**

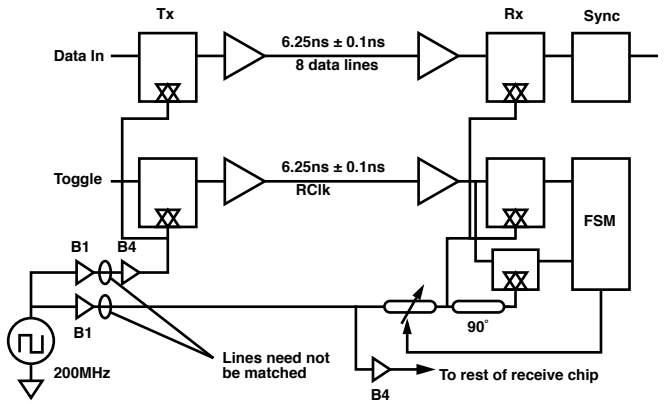
# Synchronous Timing III

## COMPONENTS of CONTROL LOOP (DLL)



# Synchronous Timing III

## TIMING ANALYSIS




**Xmit data: 50ps jitter**

**Recv data: 50ps jitter**

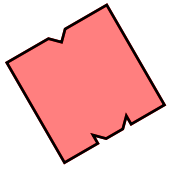
**Xmit toggle: 30ps**
**skew data/toggle (0?), 50ps jitter**

**Recv toggle: 20ps skew (0?), 50ps jitter**

**Data cable: 100ps skew**

**Toggle clock cable: 100ps skew**

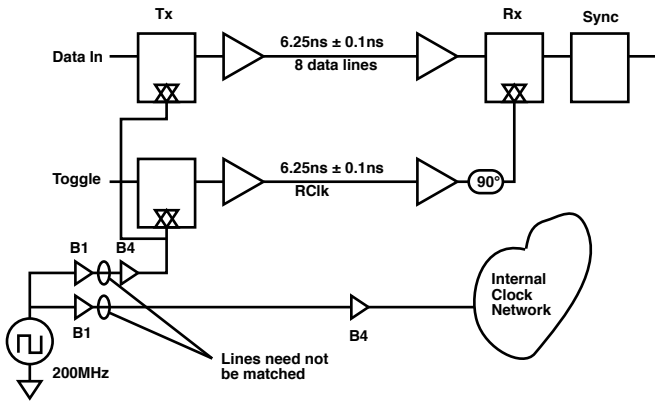
**TOTAL: 250ps skew, 200ps jitter (GOOD!)**



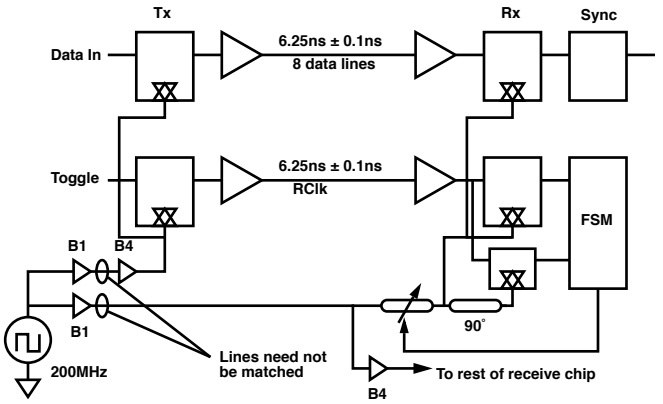
# Synchronous Timing II & III

## LIMITS TO LINE DELAY & DATA FREQUENCY

**None.**




**Only limiter to bus frequency is the rate at which you can successfully transmit & receive data (e.g. Taperture + Tuncertainty + Ttransmit)**



# Example from the other day (re: question — why does DDR need a turnaround time but Rambus does not?)