

## EE 382M: VLSI-II

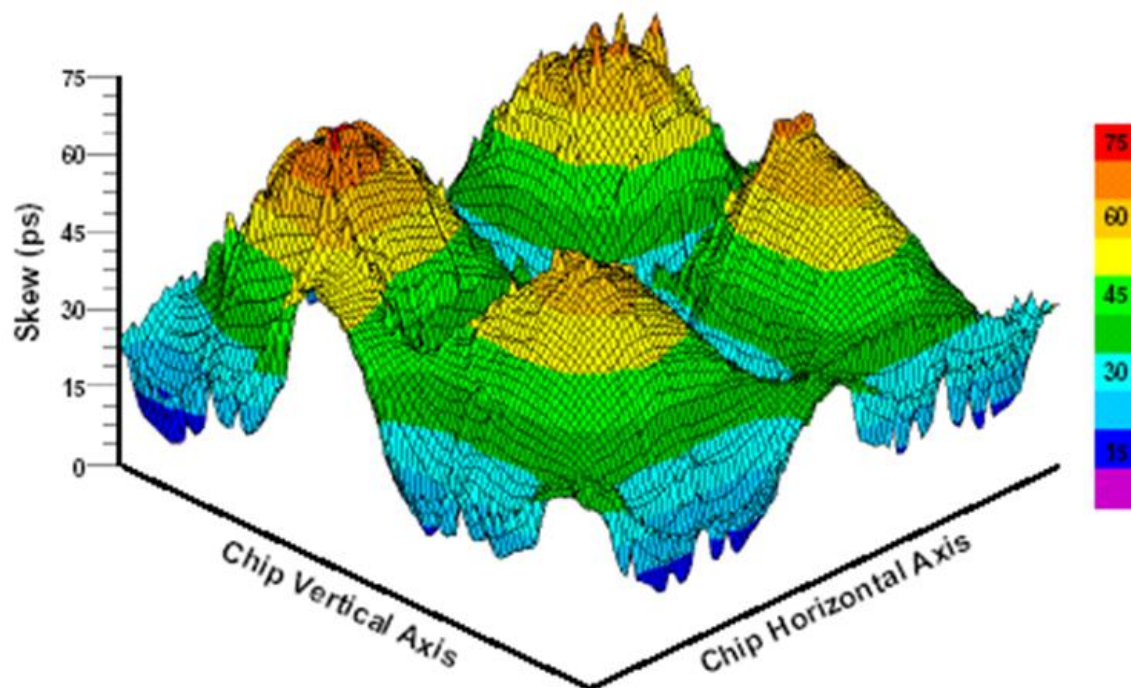
### Global Clocking

Spring 2017

Mark McDermott

Jacob Abraham

Héctor Sánchez



# Agenda

---

- **Clocking Overview**
- **Clock Uncertainty**
  - Clock Skew
  - Clock Jitter
- **Clock Generation**
- **Clock Distribution**
- **Clock Regeneration**
  - Tunable global clock buffers
  - Local clock buffers
- **Key learning's**

# Acronyms

---

- **PLL – Phase Locked Loop**
- **DLL – Delay Locked Loop**
- **PD – Phase Detector**
- **VCO – Voltage Controlled Oscillator**
- **GCDN – Global Clock Distribution Network**
- **LCB – Local Clock Buffer**
- **GCB – Global Clock Buffer**
- **MIM – Metal-Insulator-Metal**
- **PD SOI – Partially Depleted SOI**

# Clocking Overview

---

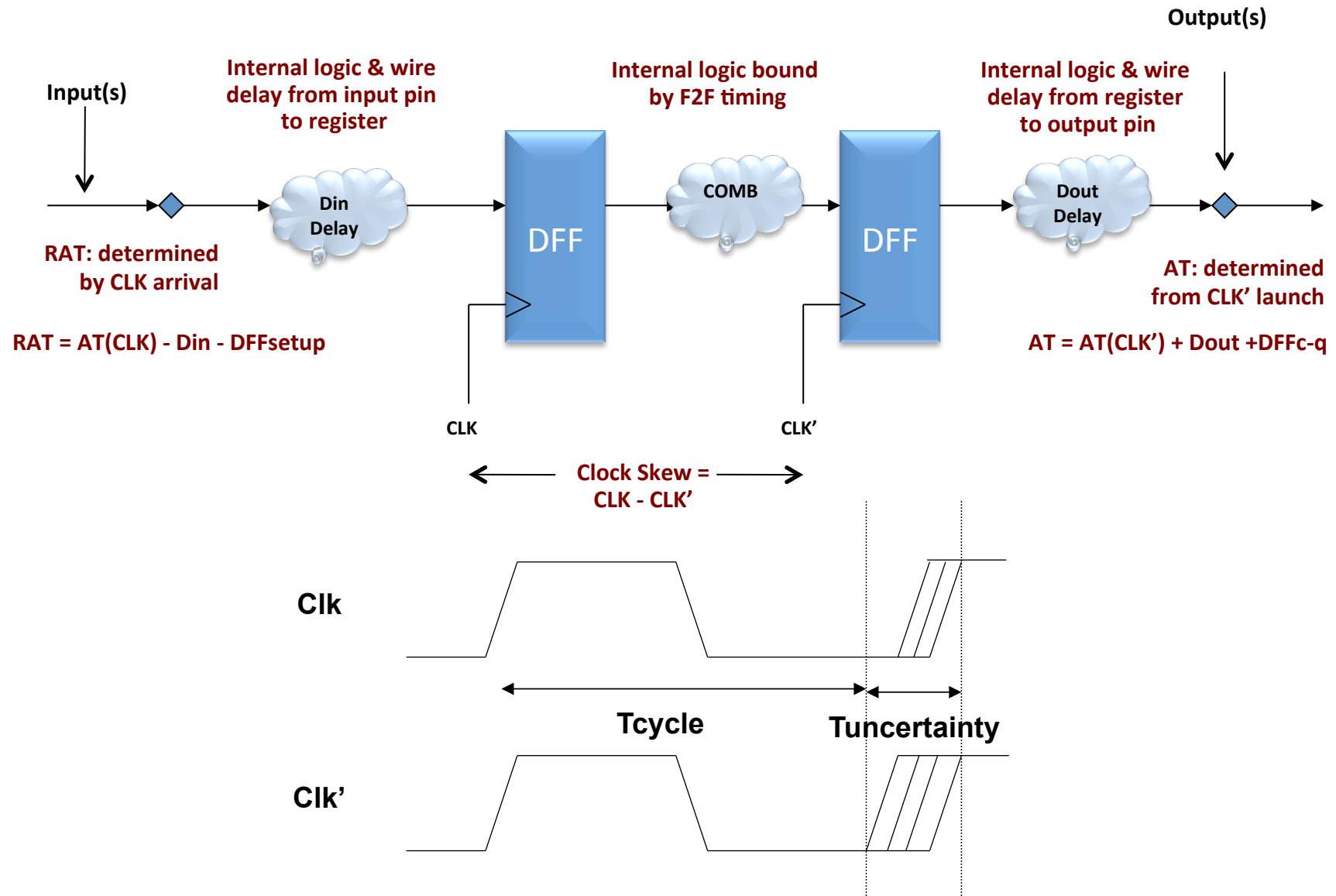
- **Most high speed digital systems use clocks to synchronize data transactions.**
- **The maximum clock frequency determines the rate at which the data can be processed.**
- **The clocking style is dependent on the circuits used to implement the logic elements and storage elements.**
- **There are three main components to clocking:**
  - **Generation: Crystal Oscillators, PLLs or DLLs.**
  - **Distribution: Trees, grids, etc.**
  - **Re-generation: LCB, GCB**

# Clocking Overview

---

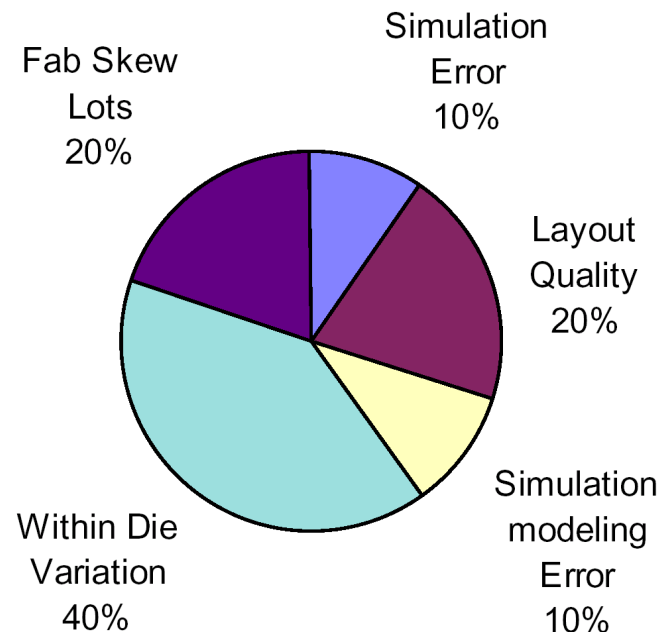
- **There are a number of issues with clocking a high performance digital system. These include:**
  - Clock uncertainty: skew and jitter.
  - Frequency dependent failures
  - Frequency independent failures.
  - Clock Distribution Power
    - Remember that clocking is merely setting up to do work and as such any clock power is essentially wasted energy!
  - Clock Distribution sensitivity to power supply noise

# Logic Transactions and Clock Dependence



# Clock Uncertainty

- Clock uncertainty is defined as the uncertainty in time in which a clock edge will appear. It is determined by clock skew, clock jitter and clock overhead.
- There are a number of sources which cause clock uncertainty. A typical breakdown of the sources are:



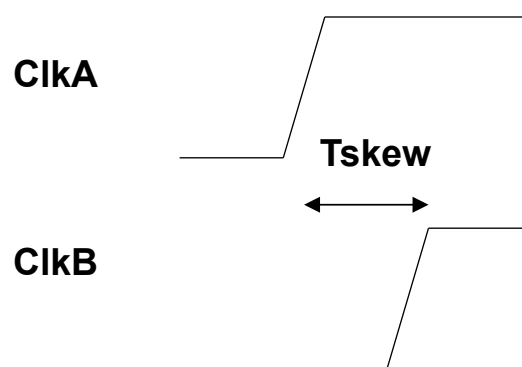
# Clock Overhead

---

- **Clock overhead refers to the time a sequential storage element needs to positively store (or resolve) the incoming data. This time is directly related to the metastability properties of the sequential storage element ( i.e. setup time or hold time )**
  
- **Clock uncertainty can have detrimental effects on the viability of a design:**
  - **Min-delay (hold) failures are frequency independent**
    - **Chip must be discarded**
  - **Max-delay (setup) failures are frequency dependent**
    - **Chip can be sold at a lower frequency**

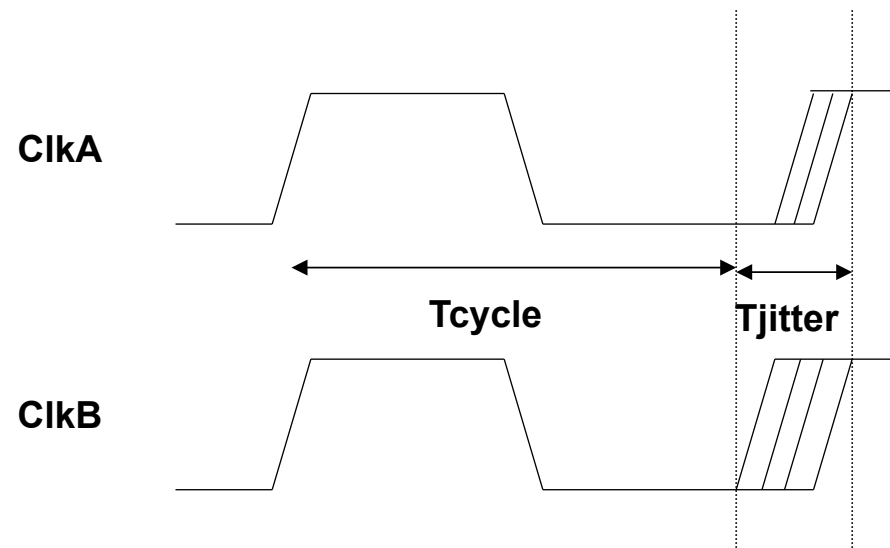
# Skew : Definition

- **Skew and Jitter are the enemies of a clocking system.**
- **Skew is the “static” time difference between any 2 electrical nodes.**
  - Typically with reference to clock signals that should in theory switch simultaneously.
  - There are techniques to reduce or eliminate skew.
  - Skew can be “managed” to your advantage (i.e., intentional (useful) skew can be used to provide “cycle-stealing” capability )



# Jitter: Definition

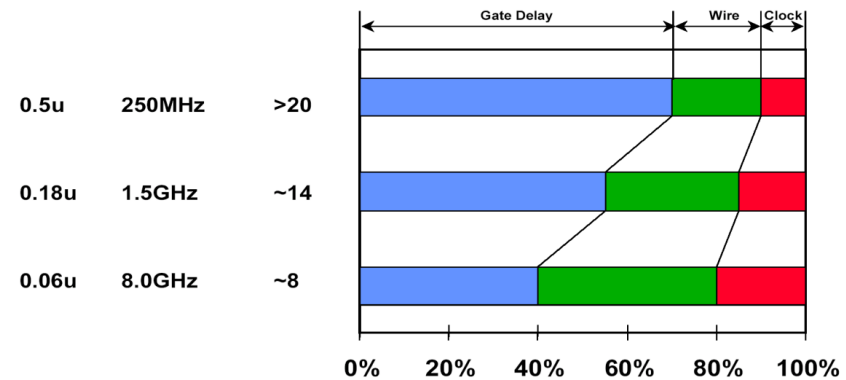
- Clock jitter is the clock edge inaccuracy introduced by the clock signal generation circuitry.
- Jitter is a “dynamic” time difference of a signal with respect to an ideal signal.
  - Jitter can not be typically “designed-out”.
  - Jitter can result from various time-dependent noise events.



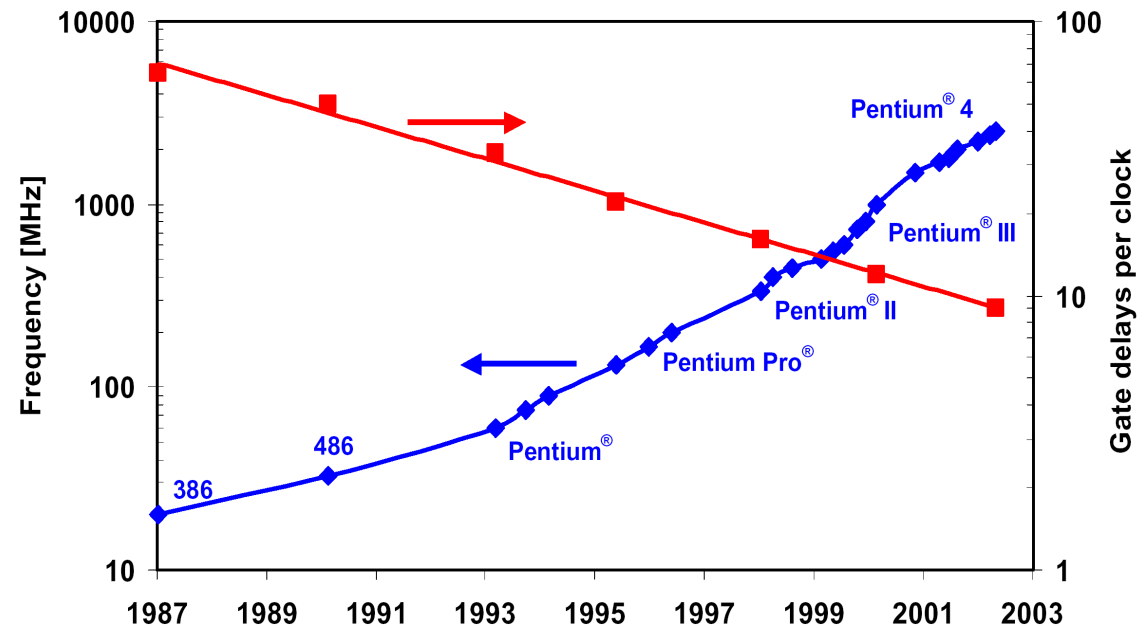
# Clocking Overview

- Clocking overhead (skew and jitter) is growing as we move to nanometer processes. Careful design of the clock generation and distribution circuits is now required for all high performance processor designs.

*Process Frequency Inv/Cycle*

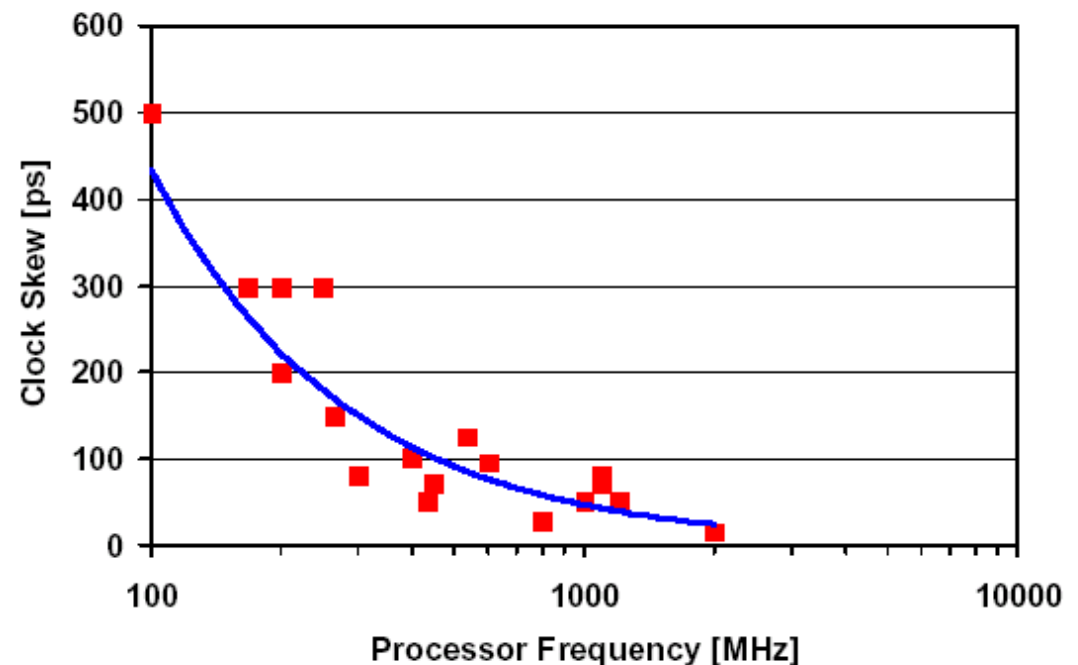


Source: D. Luick, "Beyond Superscalar RISC", ISSCC'98



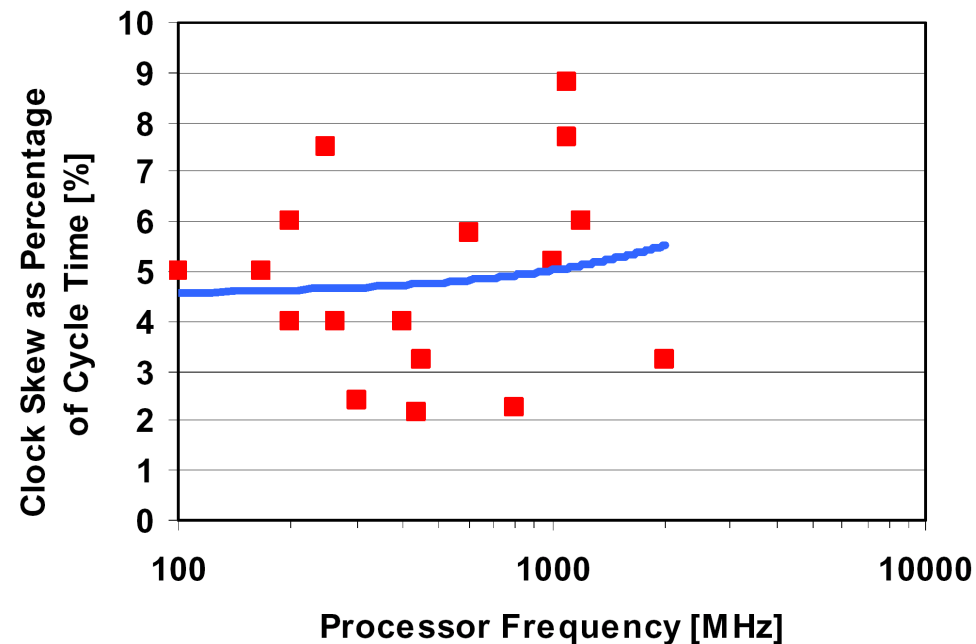
# Clock Skew

- Clock skew can be intentional or unintentional. For example, intentional clock skew may be injected in order to fix a race condition in a block of logic. This is typically achieved by the use of a variable delay clock regeneration buffer.
- Clock skew can be positive or negative depending on how the reference clock is chosen.



# Clock Skew

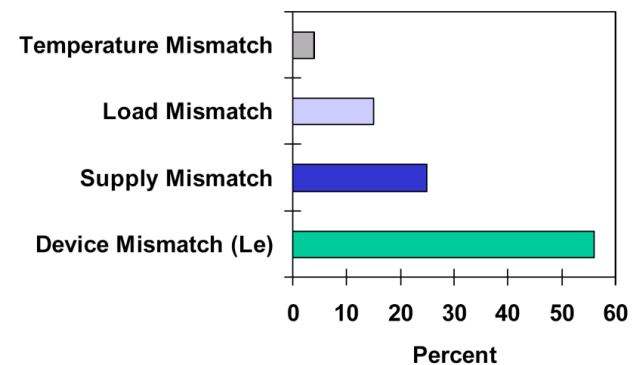
- **Clock skew accounts on average for about 5% of the cycle time and is trending higher as frequency increases**



- **In general we strive to minimize clock skew, however we must design our circuits to be clock skew tolerant.**
  - **Good reading on this topic:**
    - **Harris, Skew Tolerant Circuit Design, Morgan Kaufmann Publishers**

# Sources of Clock Skew

- **In-die Process, Voltage, Temperature (PVT) variation**
  - Different clock buffers with different channel lengths
  - Local drop in voltage leads to increased buffer delay
  - Hot spots lead to increased gate and wire delay
  - Device mismatch across die
- **Wire coupling**
  - Coupling will be different on different clock routes
- **RC mismatch**
  - Clock routes not all of equal length
  - Latches not all equal distance from LCB (local clock buffer)
- **Inductance of high speed, low resistance lines changes edge-rates. Unequal buffering can cause additional skew due to rise time-dependent delay in buffers.**



# Industry clock skew data

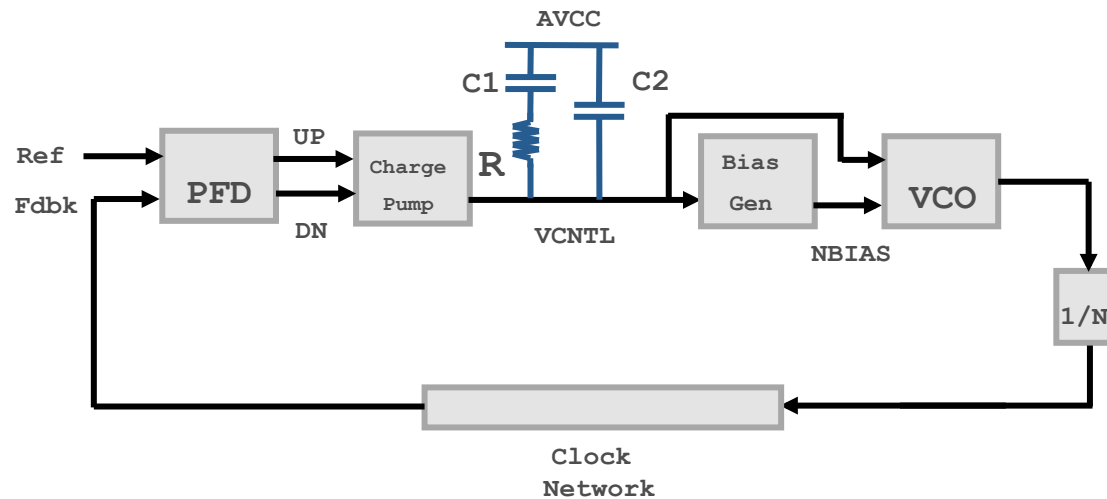
Processor	Frequency (MHz)	Clock Skew (ps)	Process
Itanium™	800	110ps w/o deskew 28ps w/ deskew	180nm
PowerPC	1000	15ps with Cu wires	220nm
UltraSPARC III	800	80ps Al wires, no deskew	180nm
Alpha	600	72ps Al wires, no deskew	130nm
IBM Cell	2000	10ps Cu, Low-K	90nm
Xeon	3600	19ps Cu, Low-K, High-K Metal Gate, w/o deskew	45nm

Source: ISSCC Papers

- **Clock jitter is the clock edge inaccuracy introduced by the clock signal generation circuitry. Clock jitter may be viewed as a statistical variation of the clock period or duty cycle.**
  
- **Sources of clock jitter:**
  - **Temporal power supply variations**
    - Changing activity can alter supply voltage in different cycles affecting either the global or regional (local) clock buffers.
  
  - **PLL Jitter**
    - Supply variation at PLL can affect oscillator frequency
    - PLL components do not have zero response time
    - Reference clock jitter being multiplied by the PLL
    - Global clock distribution may add jitter to PLL due to supply noise causing the feedback clock signal to seem to jitter.
  
  - **Wire coupling**
    - Changing data can alter coupling in different cycles

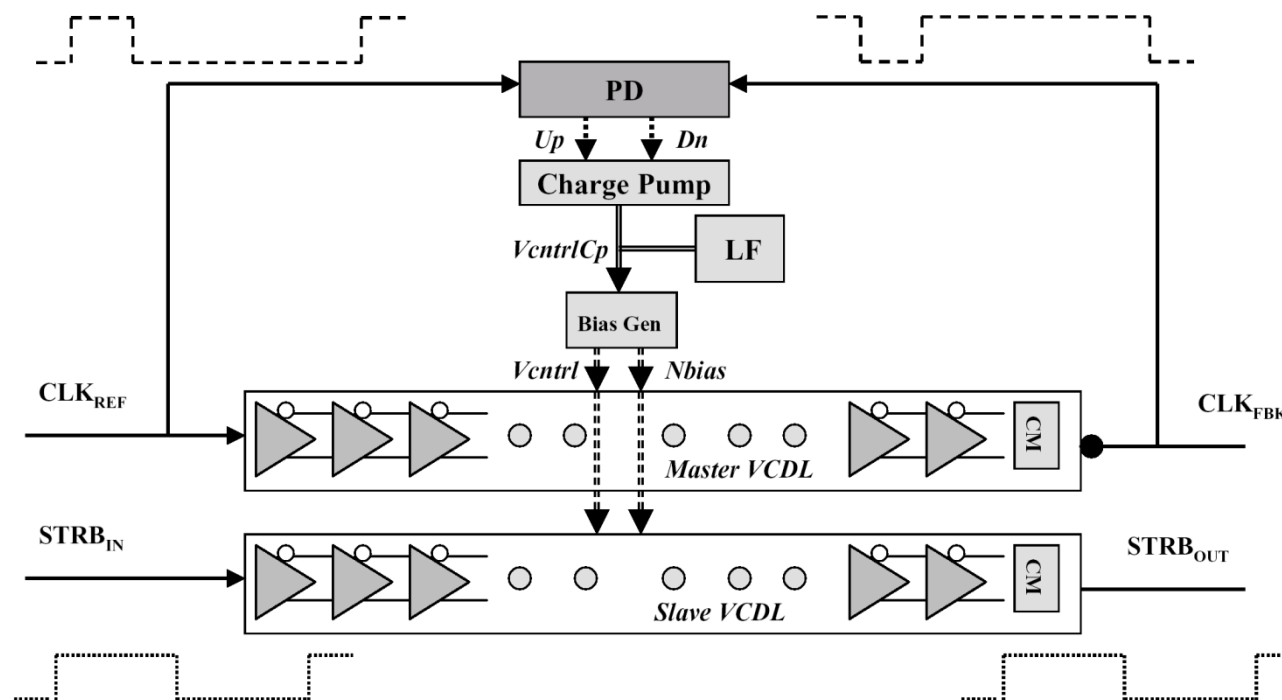
# Clock Generation: PLL

- There are two techniques used to synchronize the clocks in a high performance system: Phase Locked Loop (PLL) or a Delay Locked Loop (DLL)
- The PLL is used to “phase” synchronize (and probably multiply) the system clock WRT to a reference clock (internal or external).
  - PLL features:
    - Frequency Multiplication to run processor at faster speed than memory interface.
    - Skew reduction. The reference clock is “aligned” to the feedback clock.
    - Possible “stability” issues with PLL due to 2nd or 3rd order loop behavior.

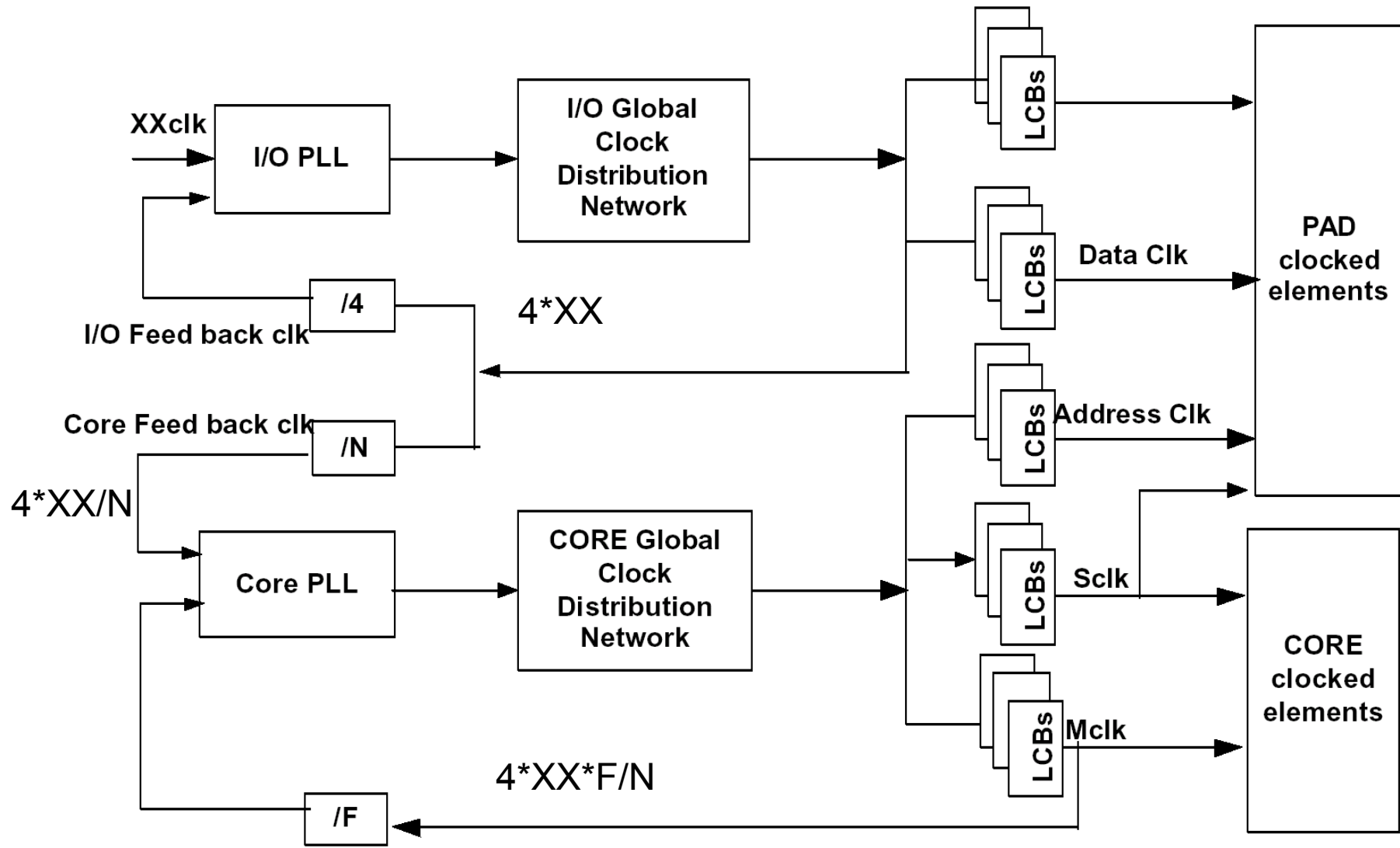


# Clock Generation: DLL

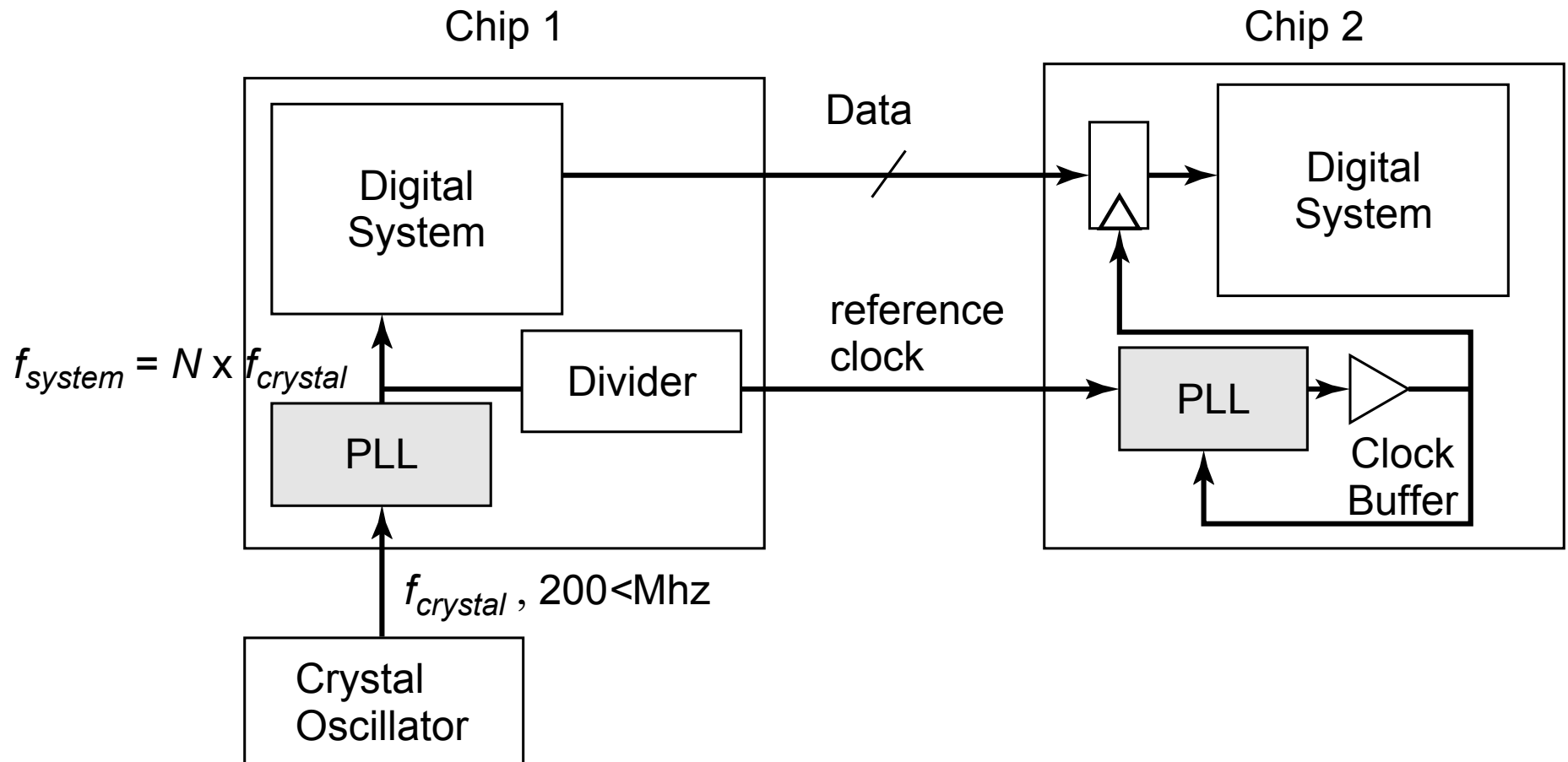
- The DLL is used to “delay” synchronize the system clock to a reference clock.
- Some high performance systems use a combination of both to generate the various clocks in a multiple clock domain design.
  - SOC designs can have many multiple frequency clock domains.



# High Performance Processor Clock Network



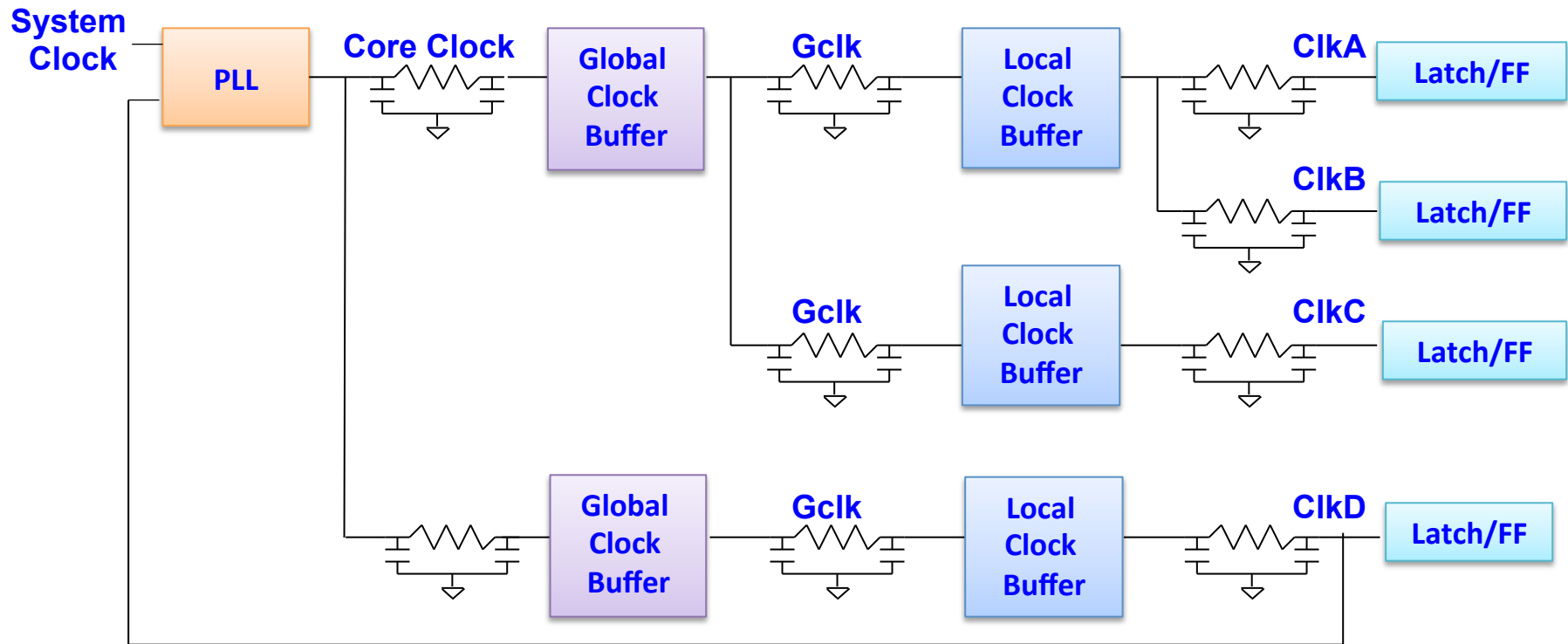
# Chip-2-Chip PLL Synchronization



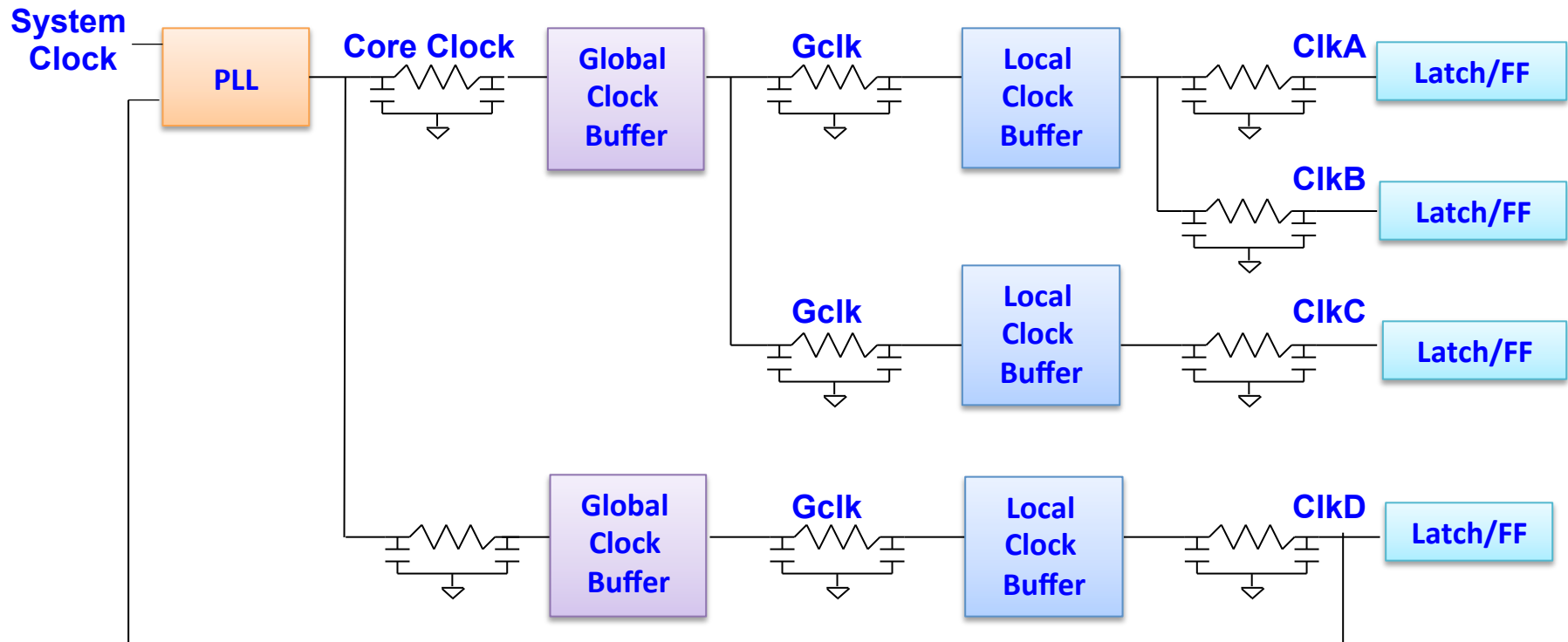
© J. Rabaey, 2002

- **Clock distribution is one of the most critical areas in the design of high performance VLSI chips.**
  - Poor clock distribution can result in excessive clock skews between clusters on the chip, reducing the maximum operating frequency.
- **In general we need to reduce the effect of clock skew on the chip. This requires:**
  - Reducing the wire delay and RC effects by making the effective delay small and balancing the delays of all the paths. (This changes a total delay problem to a matching delay problem.)
  - Matching the clock buffer delay
  - Reducing the process variations sensitivities by careful placement and design of the of clock buffers.
- **A side effect of long clock distribution delays is increased jitter due to supply voltage variations. This adversely affects the PLL used to generate the chip clock frequency.**

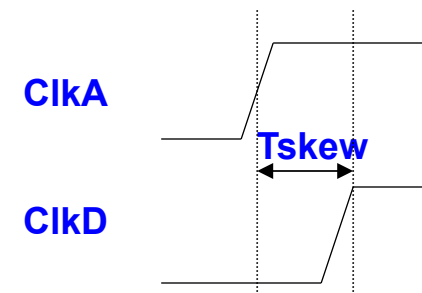
# Schematic of a typical clock distribution tree



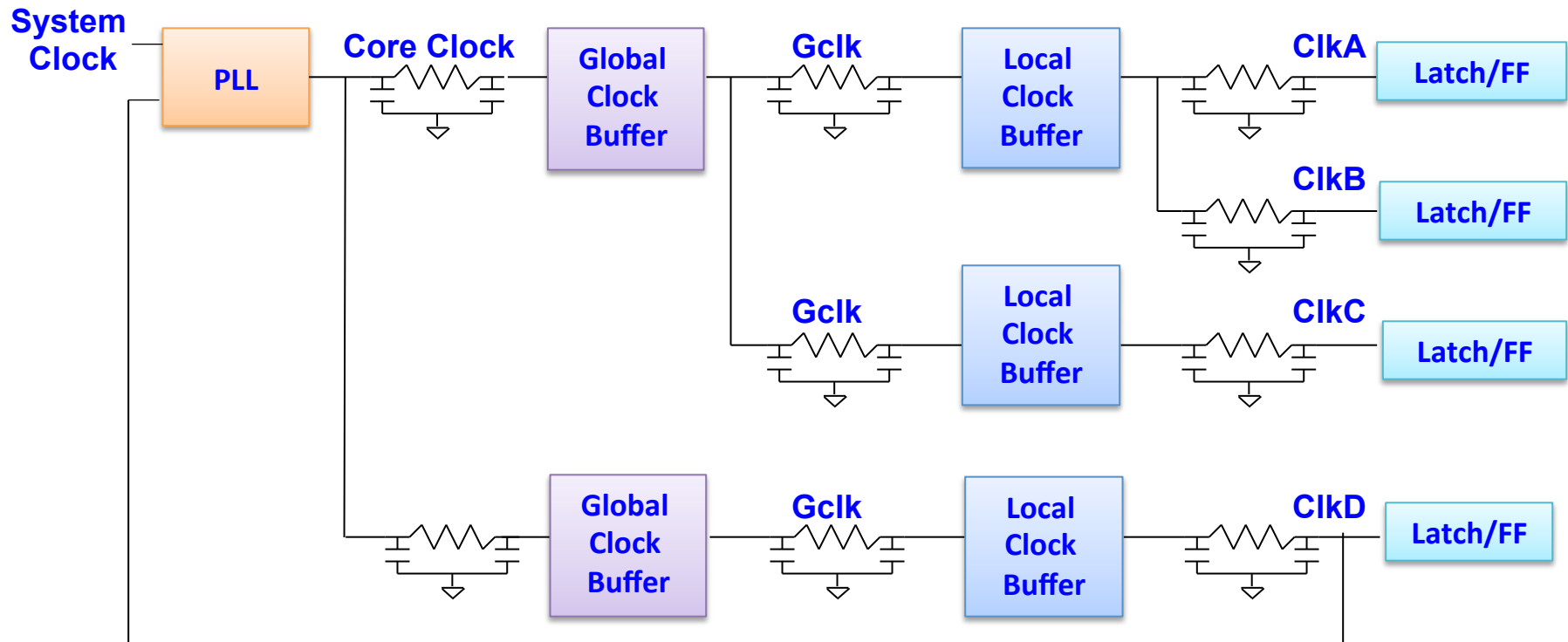
# Clock Distribution with Skew



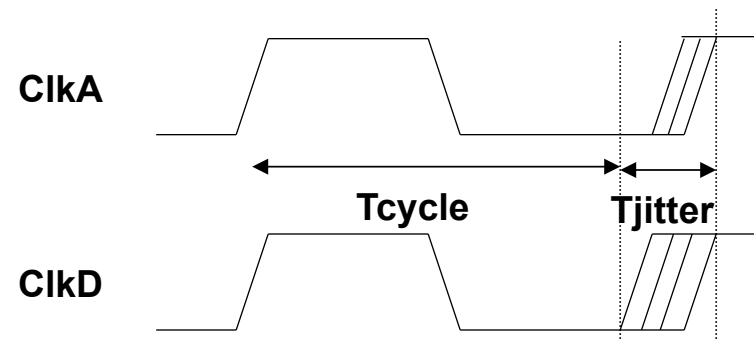
**A single transition of the core clock does not arrive at all latches or flip-flops at the same time.**



# Clock Distribution with Jitter



**Clock frequency at any point in the clock tree is not constant. The worst case jitter determines usable clock cycle time.**



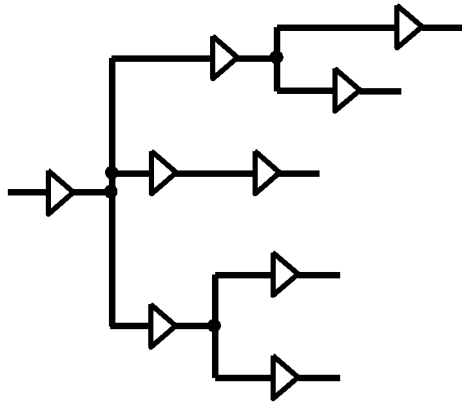
# Clock Distribution

- **There are four basic types of clock distribution networks used in high performance processor designs:**
  - **Tree:** IBM and Freescale PowerPC, HP PA-RISC
  - **Grid:** SPARC, Alpha
  - **Serpentine:** Pentium-III
  - **Spine:** Alpha, Pentium-4
- **Each technique has advantages and disadvantages:**

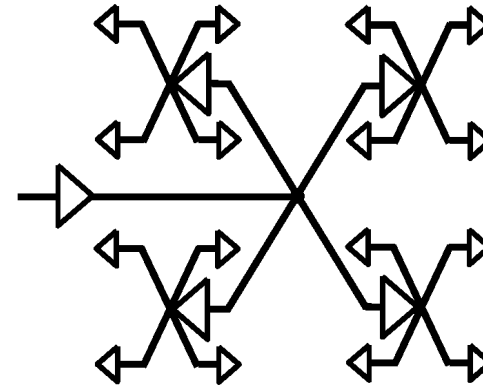
	Wire Cap	Delay	Skew
Grid	High – 15x	Low – sub100 ps	Low-Med
Trees	Low – 1x	High – 100's ps	Low
Serpentine	Very High – 30x	High – 100's ps	Low
Spine	High – 10x	Low-sub100ps	Med

# Variations of tree distribution networks

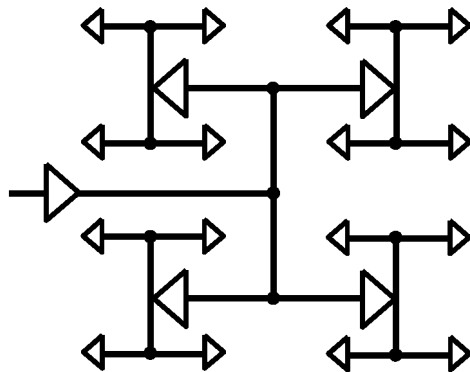
*Target: Metallization and Gate topology uniformity*



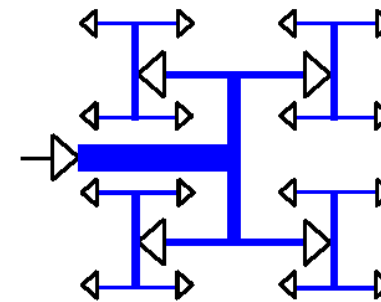
**Tree**



**X-Tree**



**H-Tree**



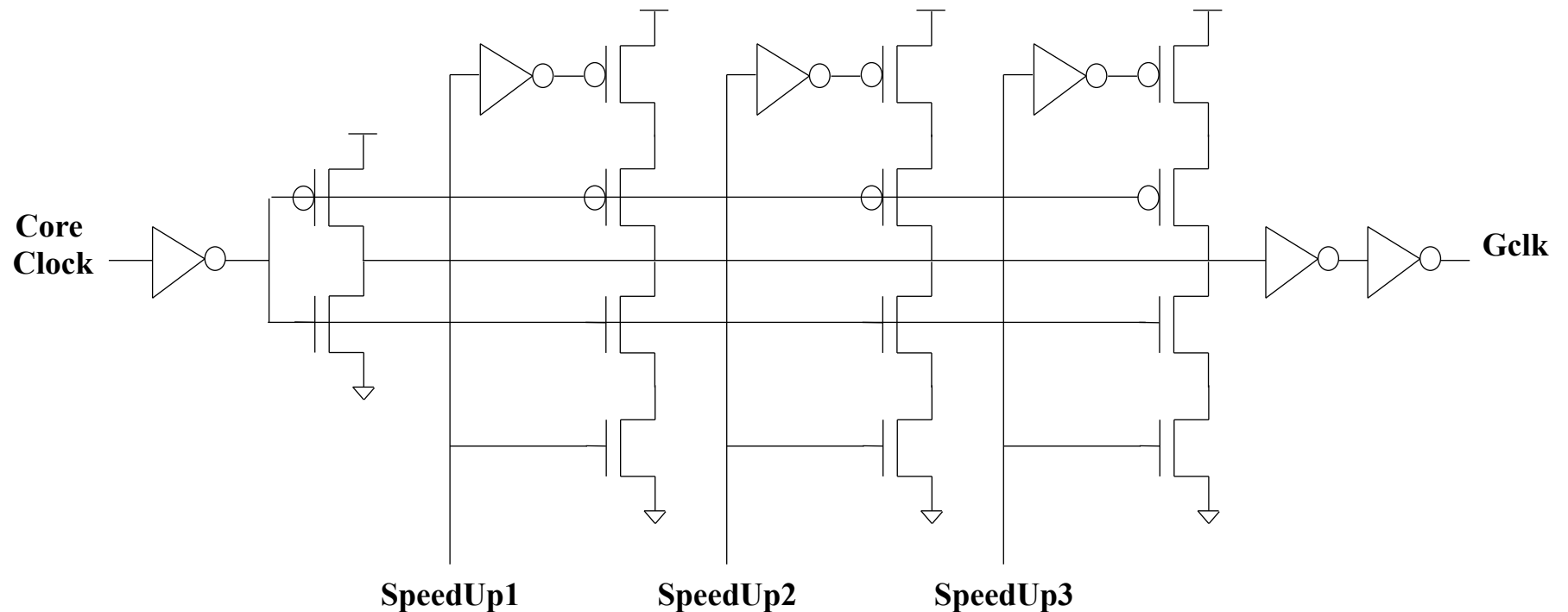
**Tapered H-Tree**

# Clock Regeneration

---

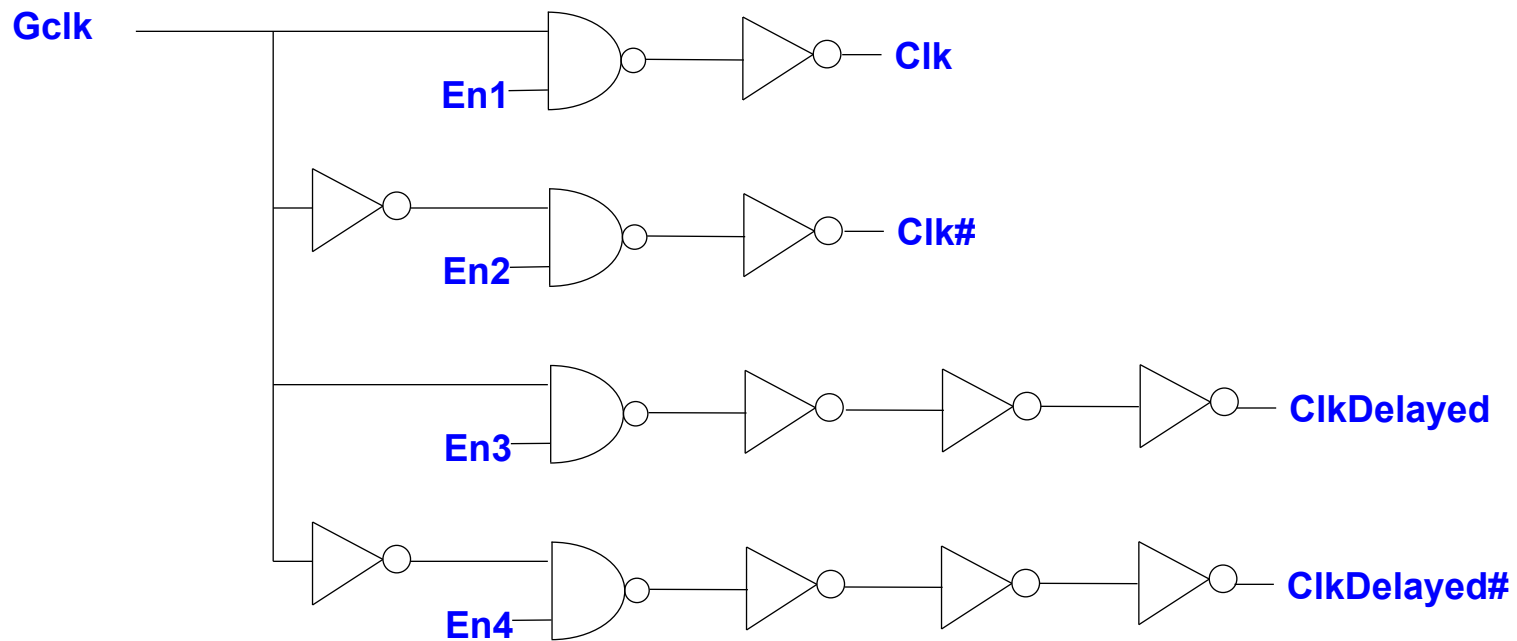
- **In the Tree, Grid and Serpentine clock networks it is necessary to buffer (regenerate) the clock signals to ensure satisfactory edge rates and reduce skew.**
- **The global clock buffers (GCB) are used to regenerate the clock signal(s) to a region or cluster in the chip. They are typically designed with skew adjustment control.**
- **The local clock buffers (LCB) are used to regenerate the clock signal(s) to functional blocks in each cluster. The LCB usually contains logic which allow the clock signals to be gated on or off to reduce power.**

# Global Clock Buffer



**Global clock buffers can use variable delay to compensate for RC mismatches**

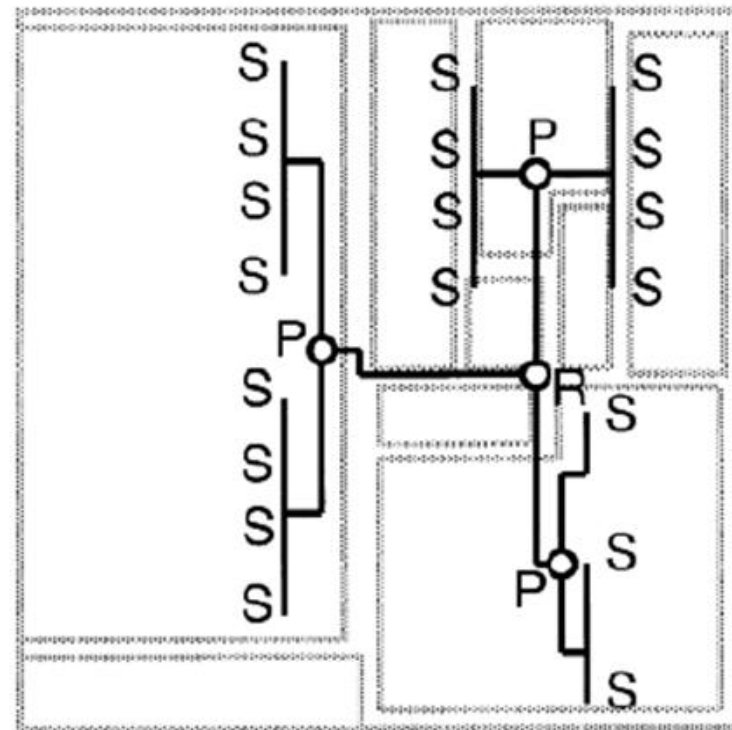
# Local Clock Buffers



**Local clock buffers use enable signals to reduce average power.**

# HP PA-RISC Clock Distribution

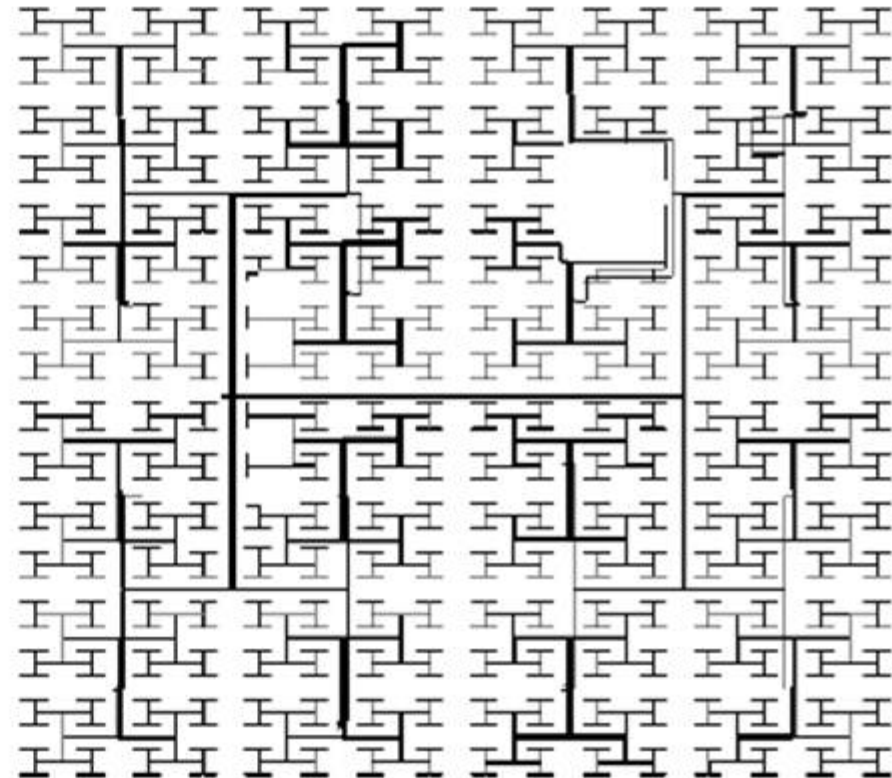
- ◆ Plan the clock usage points to minimize the distribution area
- ◆ Large caches do not require the clock to be distributed over them
- ◆ Example from the HP RISC design:
  - ◆ 1 Receiver Buffer (R)
  - ◆ 3 Primary Buffers (P)
  - ◆ 19 Secondary Buffers (S)
  - ◆ Balanced Tree Distribution
  - ◆ Die Size 21.3 x 22.0 mm



Source: P. Barnes, A 500MHz 64b RISC CPU with 1.5MB On-Chip Cache, ISSCC'99

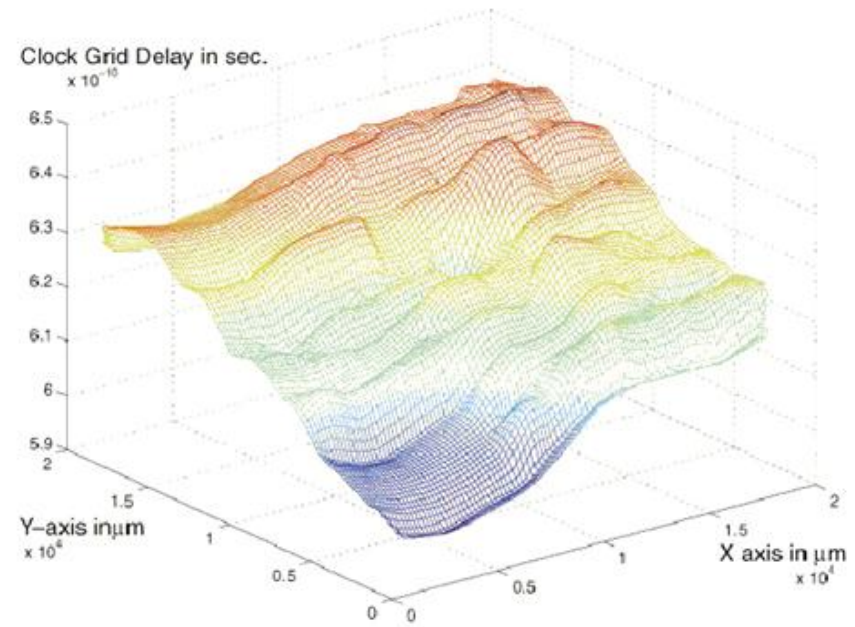
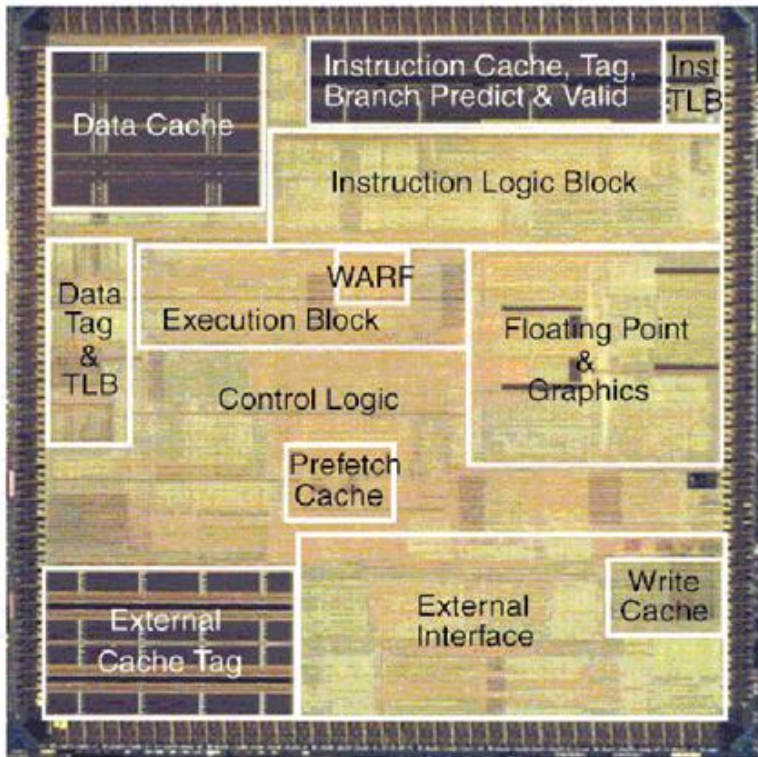
# IBM PowerPC Clock Distribution

- ◆ 0.22 $\mu$ m technology
- ◆ 17mm x 17mm die size
- ◆ 19M transistors
- ◆ 6 level metal with copper interconnect technology
- ◆ Clock tree on top 2 metal levels
- ◆ 1 GHz clock frequency
- ◆ Almost symmetric H-tree
- ◆ Simulated clock skew under 15ps



Source: A 1GHz single-issue 64b PowerPC processor, ISSCC'2000

# Sun Microsystems UltraSparc III



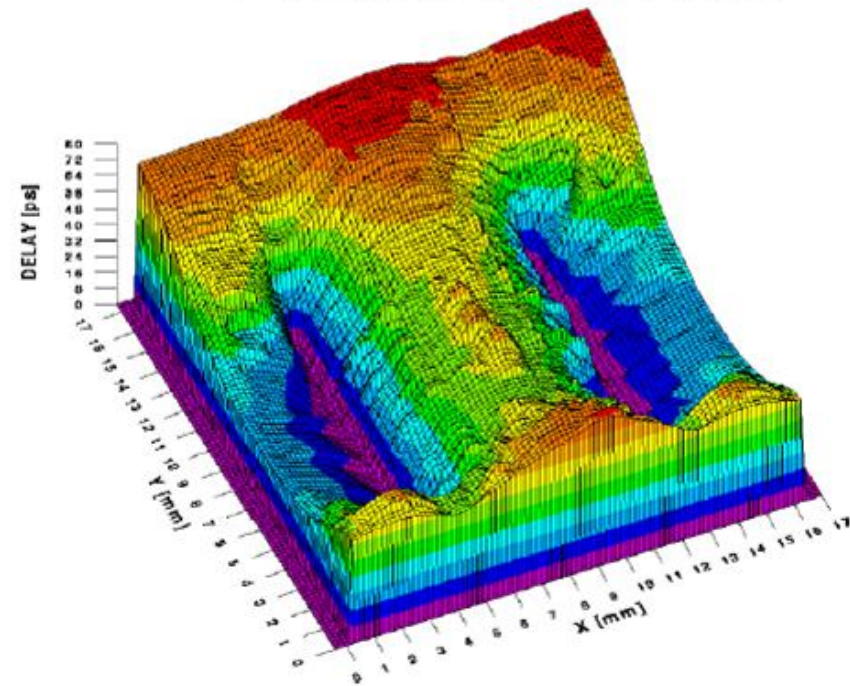
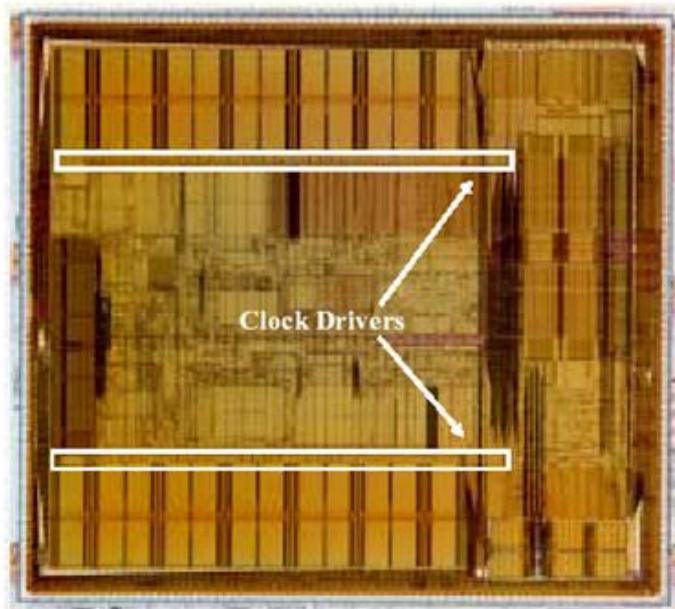
- ◆ 15.0 x 15.5 mm die size, 23M transistors
- ◆ Overall clock skew <80ps

# DEC Alpha 21164 Clock Distribution

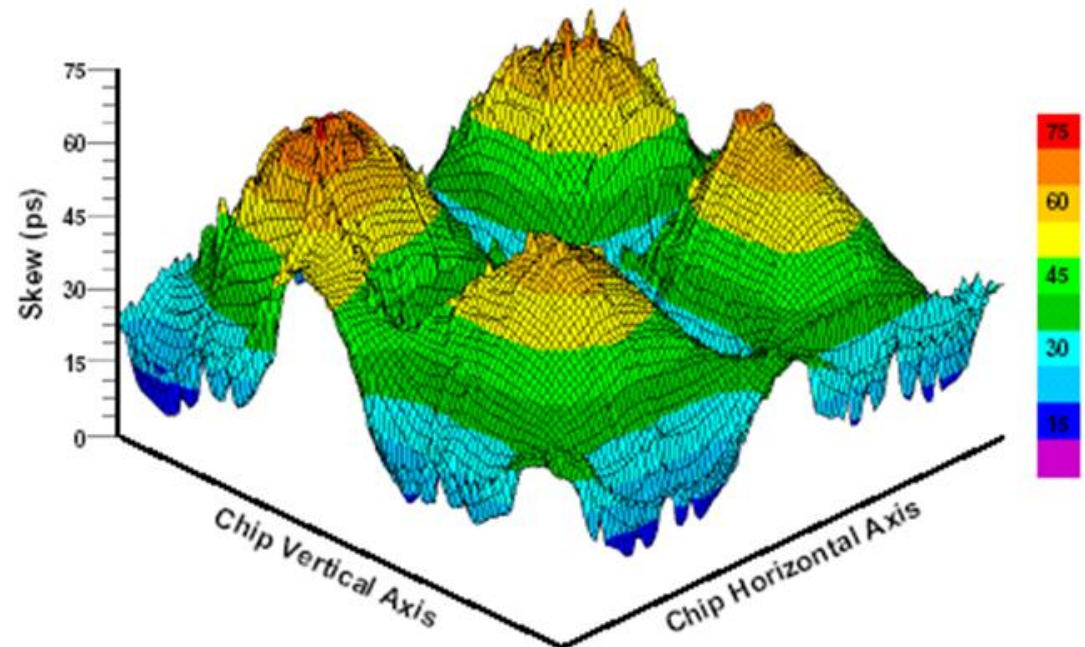
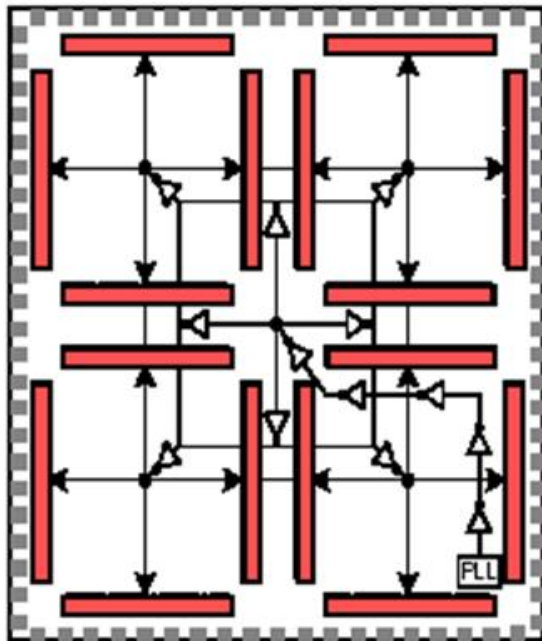
- ◆ Clock Frequency: 300MHz
- ◆ 9.3 Million transistors
- ◆ Total Clock Load: 3.75nF
- ◆ Power in Clock Distribution network: 20W (out of 50 total)

- ◆ Uses two level clock distribution

- ◆ Single 6-stage driver at center of chip
- ◆ Secondary buffers drive left and right side clock grid in M3 and M4
- ◆ Total driver size: 58cm !!



# DEC Alpha 21264 Clock Distribution

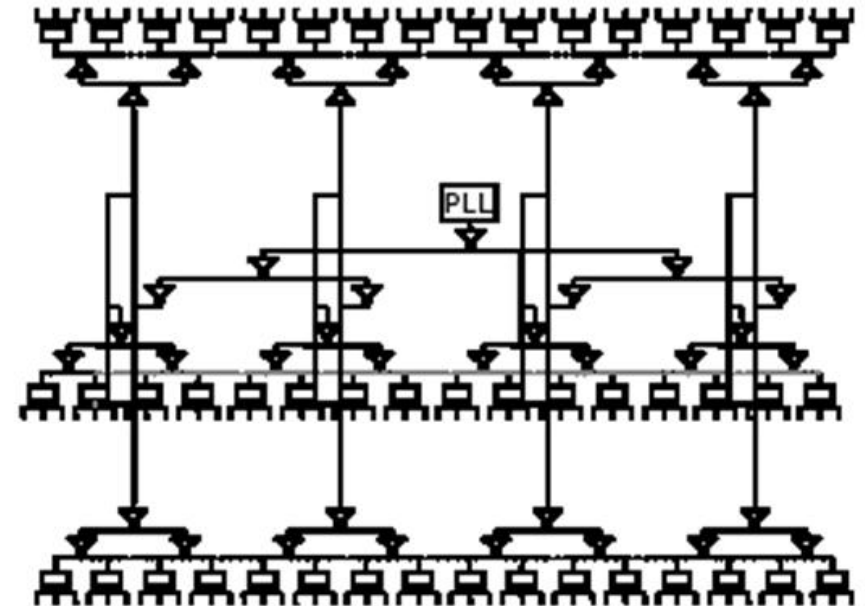
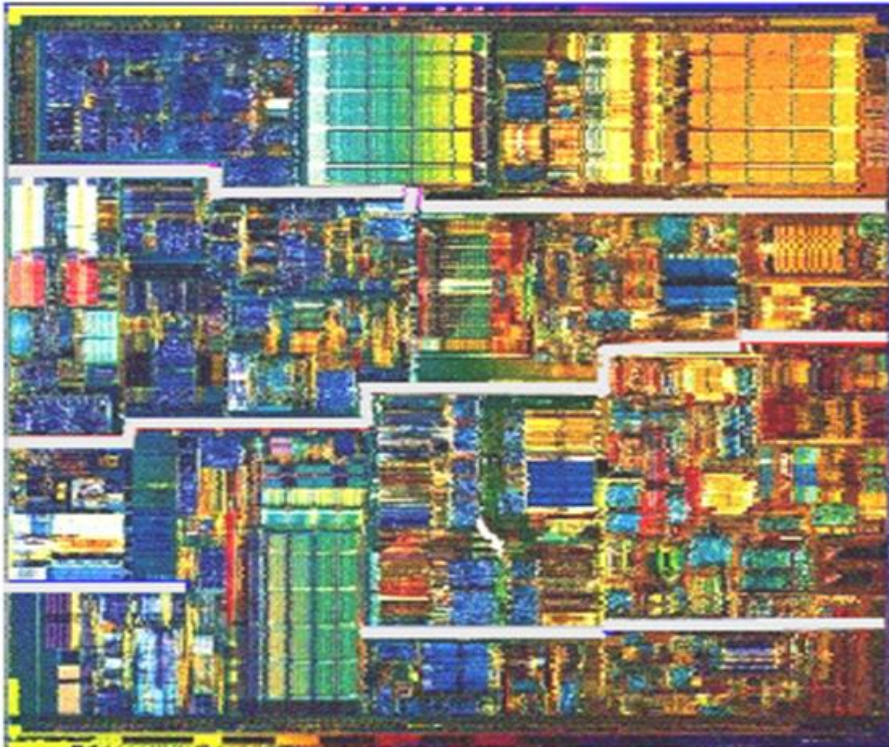


- ◆ 600MHz, 9.3M transistors, 0.35 $\mu$ m CMOS, 6 metals, 72W power dissipation at 2.0V
- ◆ Total clock load: 2.8nF

Gronowski, JSSC 1998

# Pentium® 4 Processor Clock Network

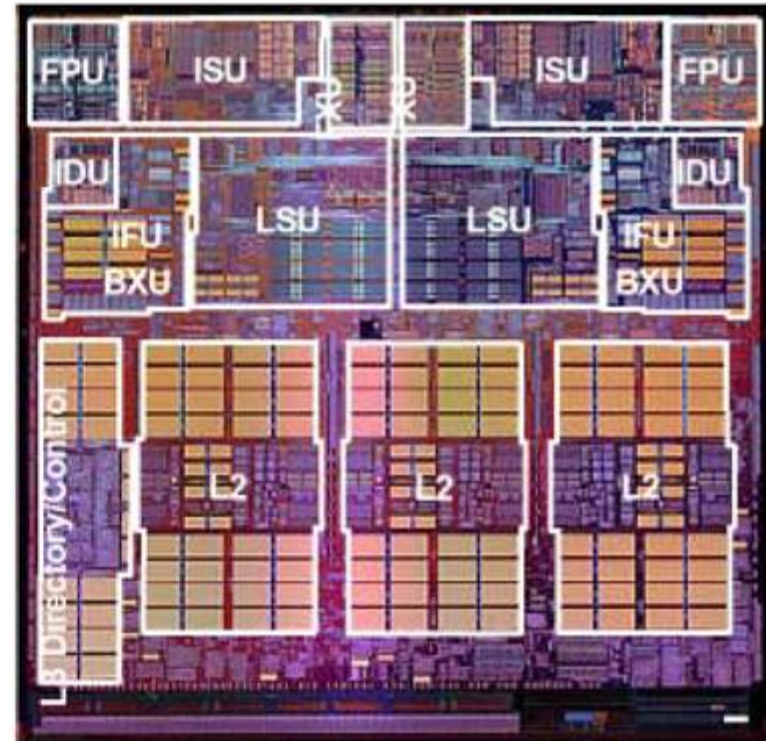
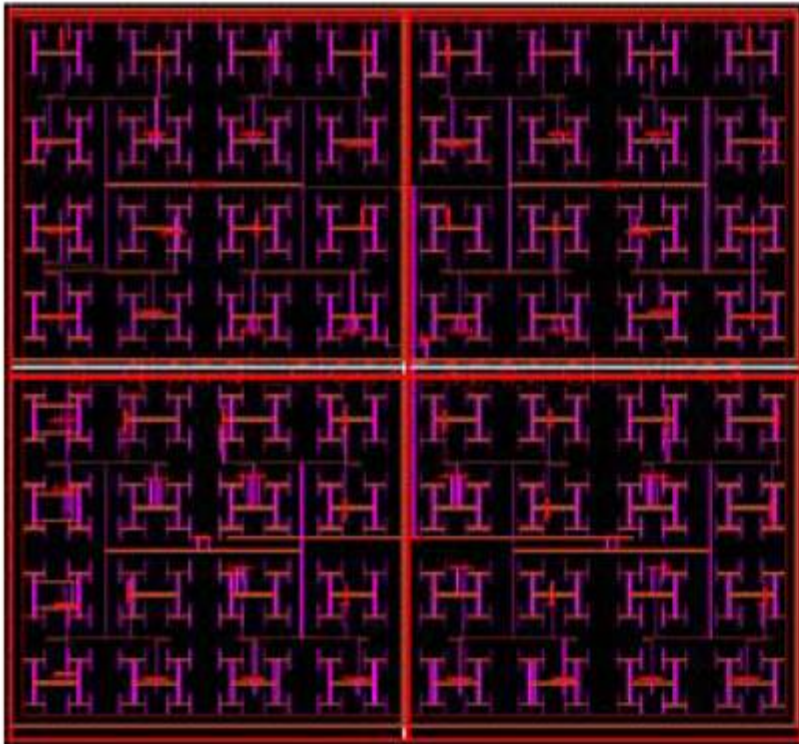
- 2GHz triple-spine clock distribution



Kurd, JSSC-2001

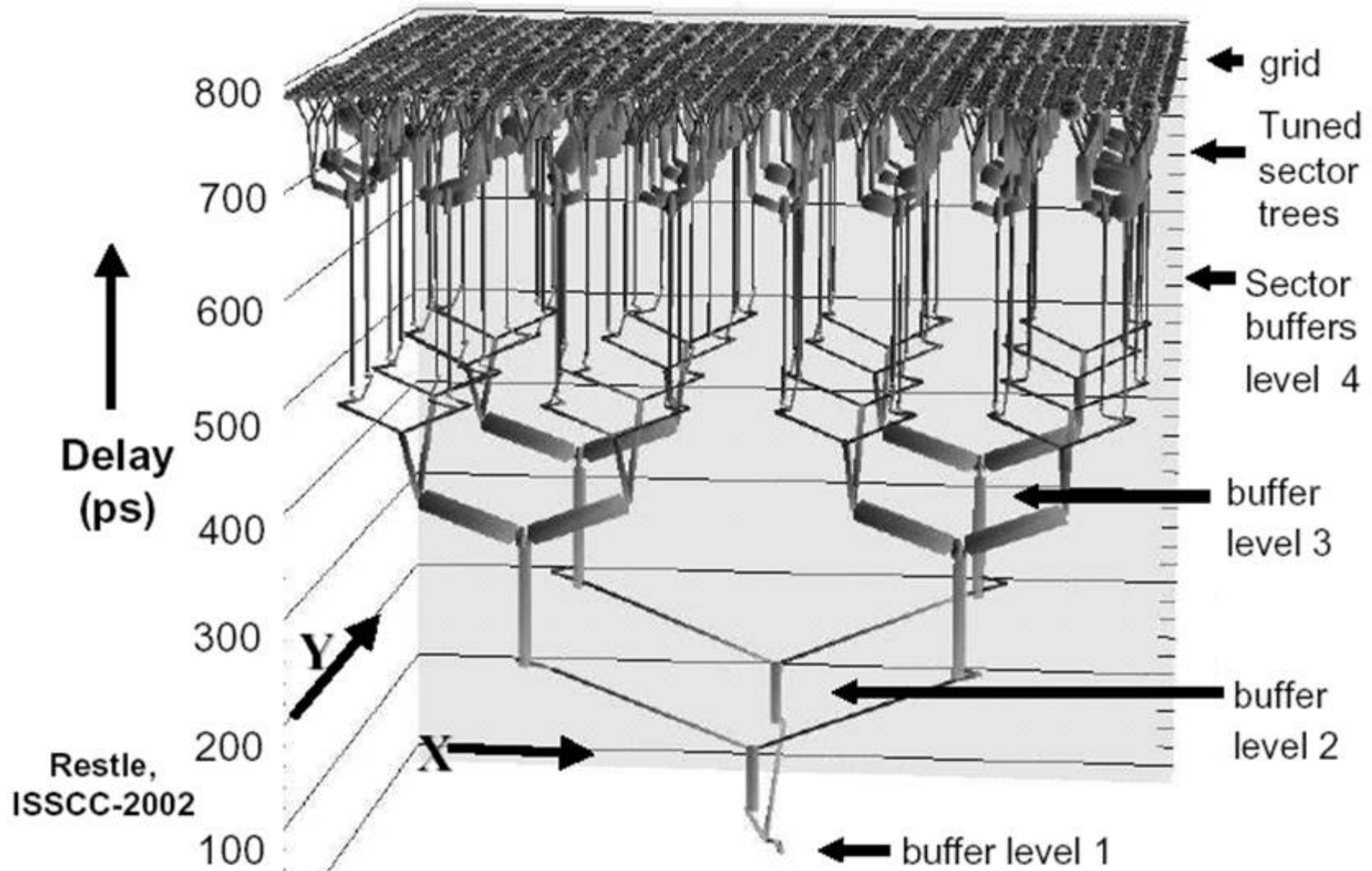
# IBM POWER-4 in .18u SOI

- H-Tree and Grid Distribution
- Clock skew: <25ps
- ~70% power in clock and
- Dual core, shared L2
- 174M transistors
- 115W at 1.1GHz, 1.5V

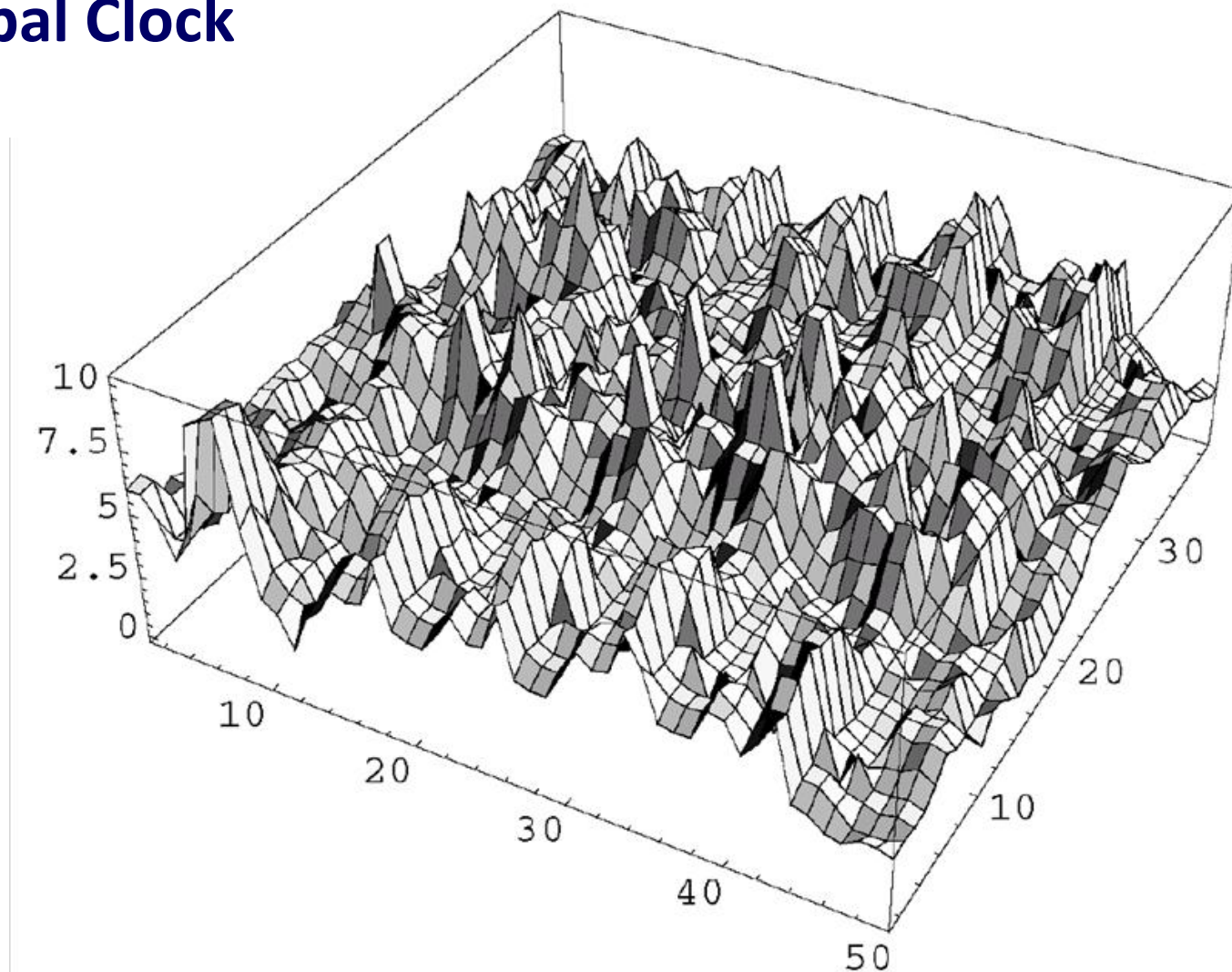


Source: Physical Design of a Fourth-Generation POWER GHz Microprocessor, ISSCC'2001

# IBM POWER-4 3D Skew Visualization



# Clock Skew across CELL Processor Global Clock Distribution

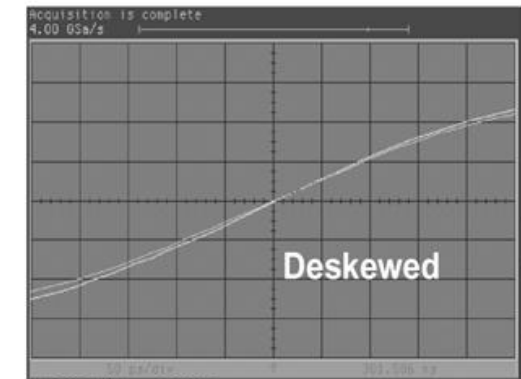
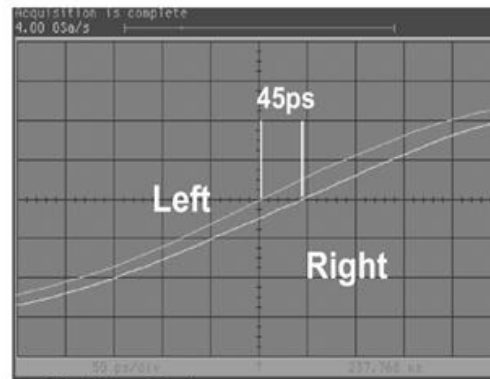


**Figure 10.2.2: Clock-skew map.**



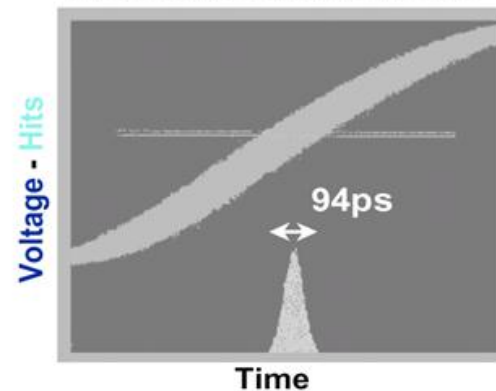
# Clock Deskewing

- **Active clock deskewing is accomplished by dynamically delaying the global clock signals.**

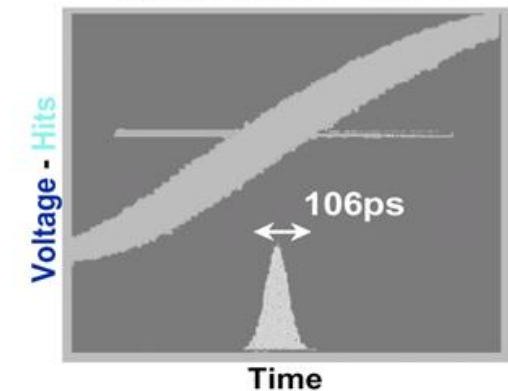


- **This can result in clock jitter. Careful analysis is required to validate the benefits.**

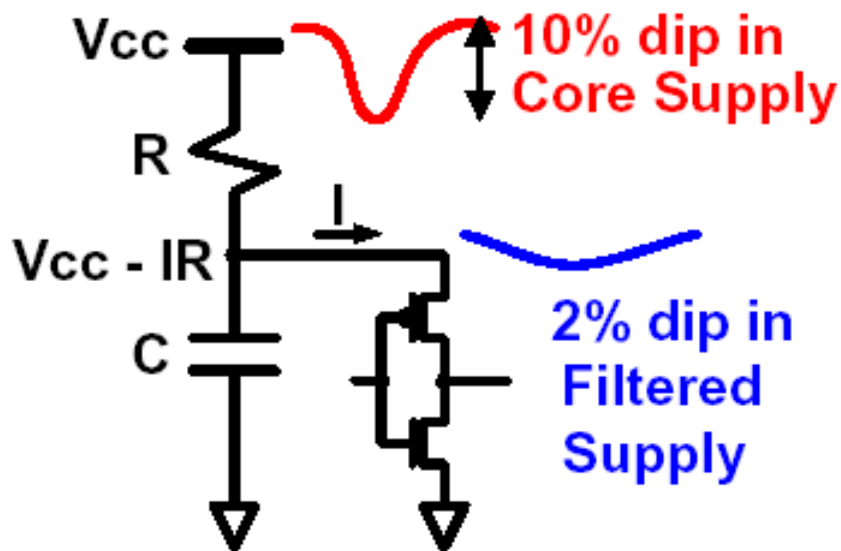
**Clock Jitter  
with out Deskew Circuit**



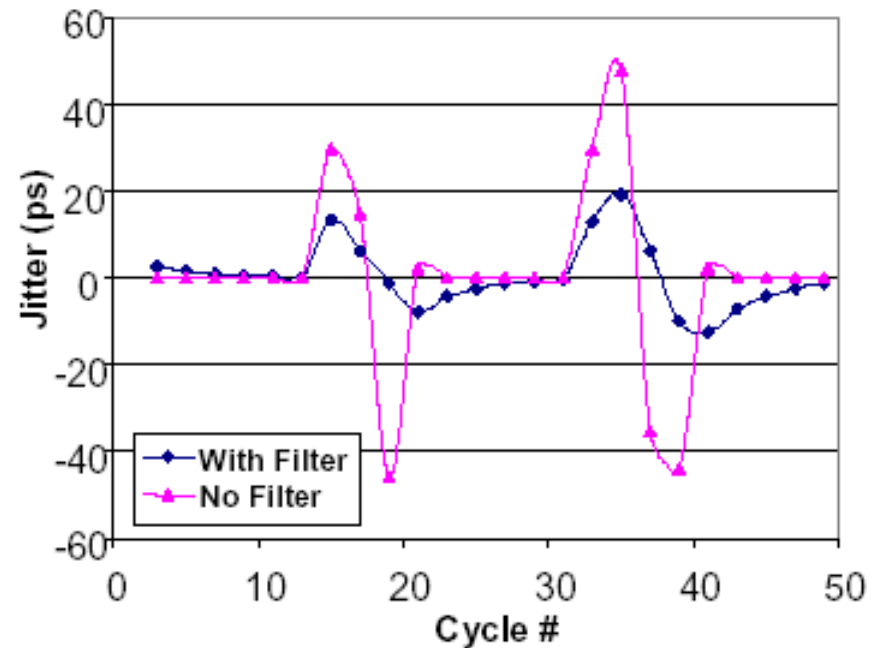
**Clock Jitter  
with Deskew Circuit**



# Pentium® 4 Processor Jitter Reduction



Kurd, JSSC-2001



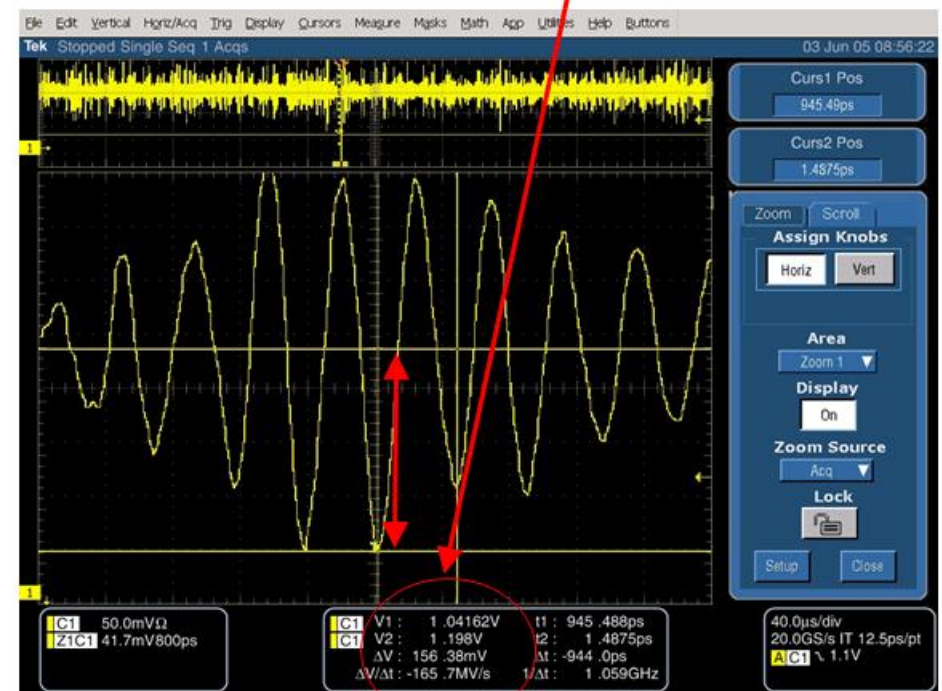
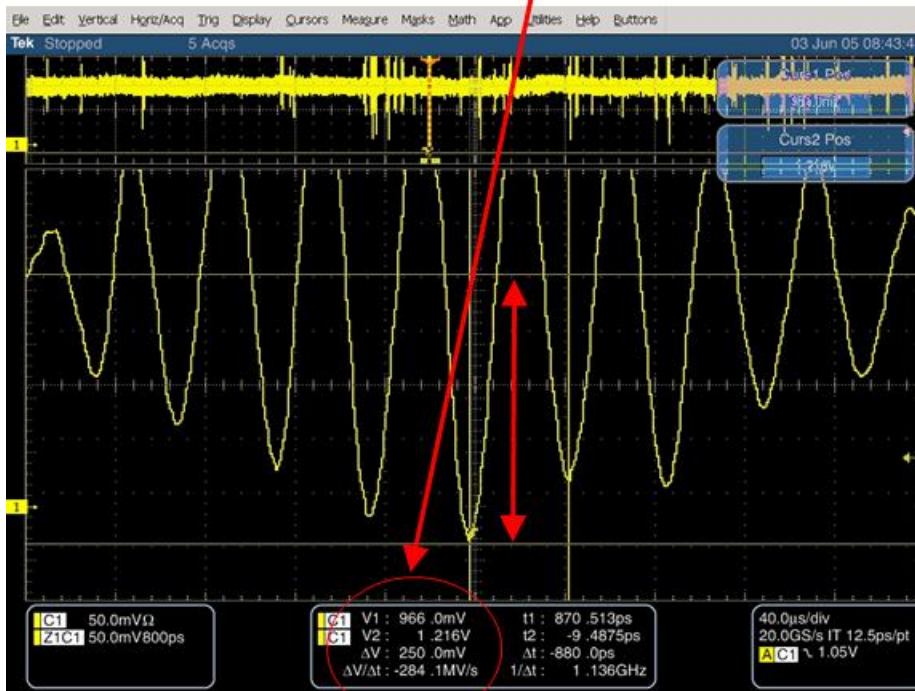
# Power Supply Noise Influences Jitter

## Voltage Comparison: MIM vs. no MIM

### 1.2V 90nm PD-SOI microprocessor

No MIM => 250mV noise => -20.6%

8ff/um<sup>2</sup> MIM => 156mV noise => -13.0%

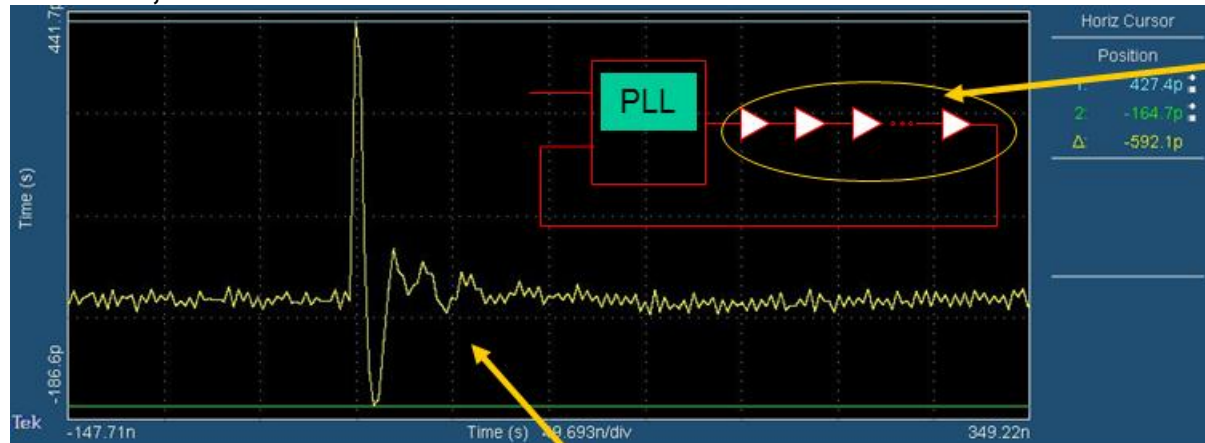


Measured Core Vdd no-MIM vs. with-MIM while running same code

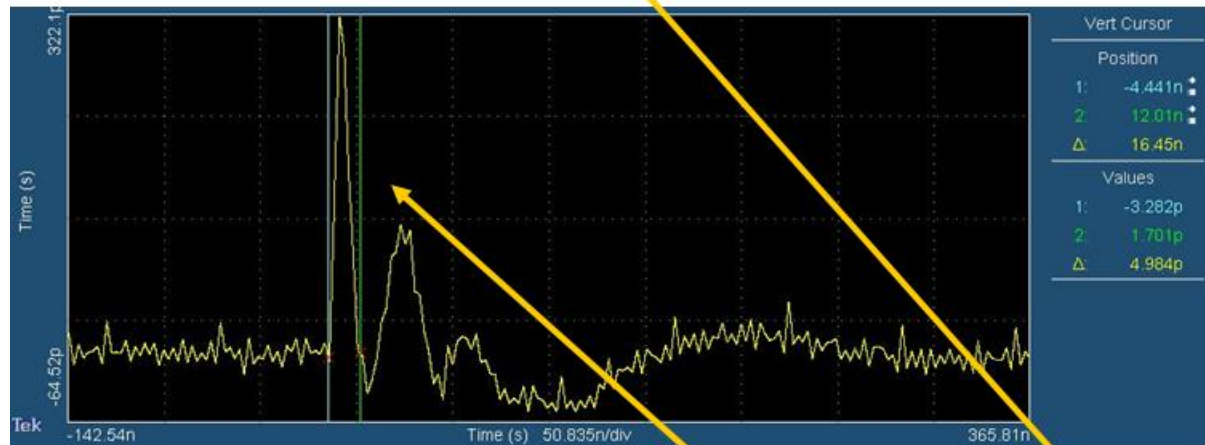
H. Sanchez, ISSCC2006

# Supply Noise Clocking Issue: Instantaneous Phase Shifts

H. Sanchez, ISSCC2006



Measured data from highly integrated SoC



- Clock distribution buffers are exposed to Vdd noise
- Large di/dt events of microprocessor Core can cause substantial Vdd noise that in turn affects the instantaneous phase relationships of clocks
- Addition of MIM decap shows a measured 40% reduction in peak-peak phase change due to large transient currents related to microprocessor Core activity.

TIE of global clock: Non-MIM shifts **592ps** as compared to MIM where it is reduced to **388ps**.

- **Reducing feature size and increasing frequency is the trend for future high performance processors.**
  - Reducing feature size means bigger impact of cross die variations.
  - Interconnect delays (RC) do NOT scale well with feature size.
  - RLC effects of the GCDN may require extensive analysis.
  - Reducing cycle time means clock skew budget is a larger percentage of the cycle time.
- **Die sizes are getting bigger**
  - Longer clock distribution networks resulting in increase skews and jitter.
  - Larger clock loads
- **Power consumption increase coupled with decreasing  $V_{dd}$  and increasing noise, pose challenges to multi-GHz clocking schemes known to be industry standards.**

# Clocking realities

---

- **If setup times are violated, reduce clock speed**
- **Useful clock skew can be your friend**
- **Jitter is never your friend**
- **Pulse latches do not scale well from generation to generation**
- **Metastability is very real (and deadly)**
- **Lastly, if hold times are violated, chip fails at any speed and PVT**
  - You have a “brick” for a chip
  - You may be out of business if you are a startup

# Key Learnings

---

- **Choose your skew battles wisely**
  - Look at the overall skew and fix it globally if possible.
  - It is okay to be loose in one area if you can gain back in effort, productivity and better design.
  - Getting the design out a quarter earlier at a slightly lower speed wins. There is a 1% “effective” performance loss for every week the schedule slips.
- **Minimize if not eliminate interconnect matching requirements**
  - Fewer paths to match means fewer things can go wrong
- **Make clock distribution more tolerant to design mismatch**
  - Don’t need to tune each path correct to the last picosecond
- **Unit and block clock loading data changes till the last minute**
- **Final clock loadings can be up to 2.5x the original expectations**
- **Most practical solutions involve combination of techniques ( i.e. global H-tree, local grid ... )**

- **A successful clock distribution network should:**
  - Maintain low (useful) skew
  - Be tolerant to design mismatches and deviations from ideality
  - Reduce and or eliminate very careful matching of layouts
  - Reduce or eliminate final tuning effort right before tapeout
  - Reduce / minimize design effort
  - Include ability to be flexible: plan for things to go wrong, and put in place backup solutions.

# References

---

- **G. Geannopoulos, X. Dai - “An adaptive digital deskewing circuit for clock distribution networks”, ISSCC Digest of Technical Papers, 1998, Pg 400 -401**
- **S. Tam, et al, - “Clock generation and distribution for the first IA-64 microprocessor”, IEEE Journal of Solid-State Circuits, Nov. 2000, Pg 1545-1552**
- **D. Harris, S. Naffziger – “Statistical clock skew modeling with data delay variations”, IEEE Transactions on VLSI Systems, Dec. 2001, Pg 888-898**
- **E. Friedman – “Clock distribution networks in synchronous digital integrated circuits”, Proceedings of the IEEE, May 2001, Pg 665-692**
- **N. Kurd, et al, - “A multi-GHz clocking scheme for the Pentium®4 microprocessor”, IEEE Journal of Solid-State Circuits, Nov. 2001, Pg 1647-1653**
- **P. Restle, et al, - “The clock distribution of the Power4 microprocessor ”, ISSCC Digest of Technical Papers, 2002, Pg 144 -145**
- **D. Bailey, B. Benschneider - “Clocking Design and Analysis for a 600-MHz Alpha Microprocessor”, IEEE Journal of Solid-State Circuits, Nov. 1998, Pg 1627-1632**

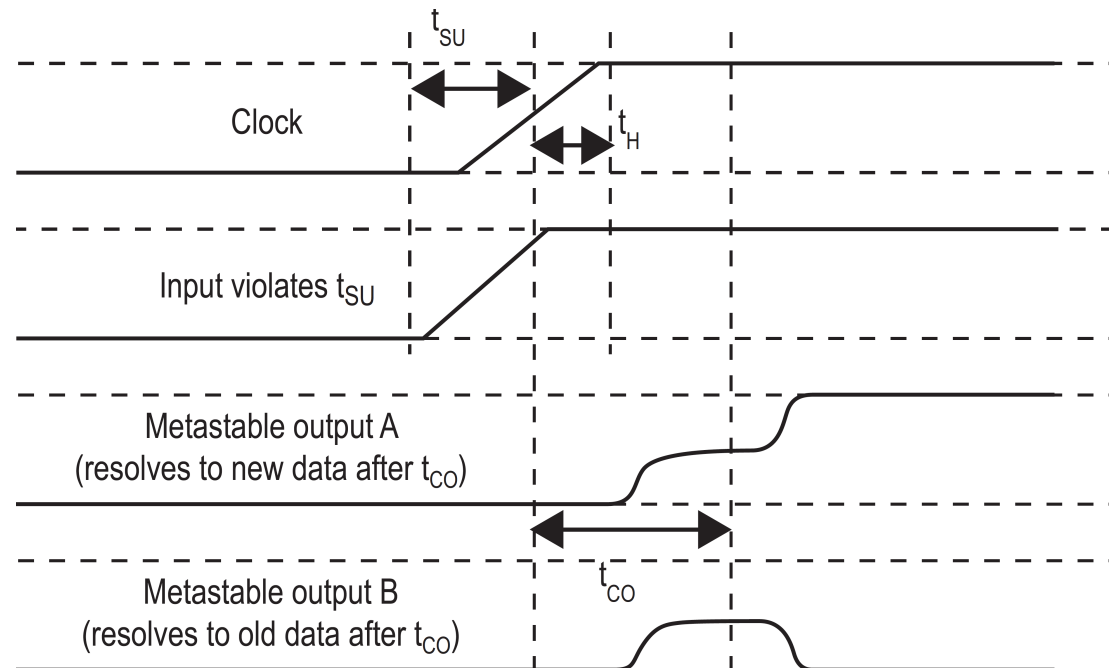
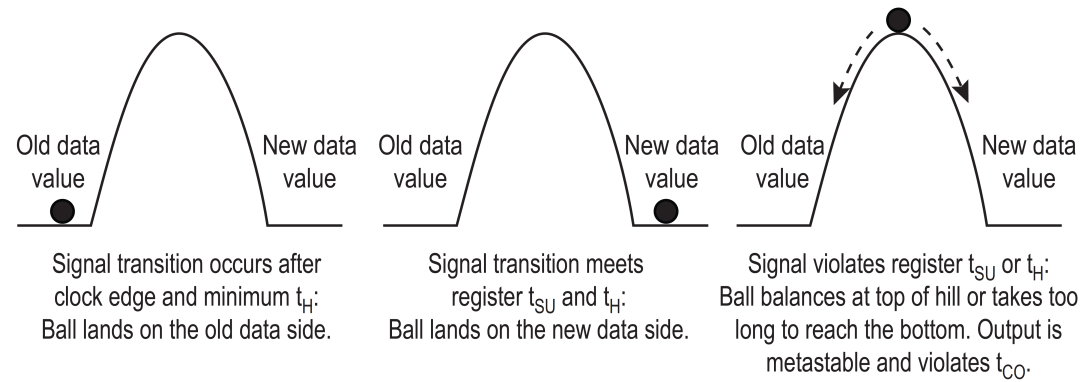
# References

---

- P. Gronowski, et al, "High-performance microprocessor design," IEEE Journal of Solid-State Circuits, May 1998, Pg 676 - 686
- J. Maneatis - "Low-jitter process-independent DLL and PLL based on self-biased techniques", IEEE Journal of Solid-State Circuits, Nov. 1996, Pg 1723 –1732
- I. Young, et al, - "A 0.35um CMOS 3-880MHz PLL N/2 Clock Multiplier and Distribution Network with Low Jitter for Microprocessors", ISSCC Digest of Technical Papers, 1997
- H. Kojima - "Half-swing clocking scheme for 75% power saving in clocking circuitry", IEEE Journal of Solid-State Circuits, April 1995, Pg 432 -435
- T. Xanthopoulos, et al, - "The Design and Analysis of the Clock Distribution Network for a 1.2GHz Alpha Microprocessor", ISSCC Digest of Technical Papers, 2001
- J. Wood, et al, - "Multi-gigahertz low-power low-skew rotary clock scheme", ISSCC Digest of Technical Papers, 2001, Pg 400 -401

# Metastability

# Metastability



Courtesy Altera

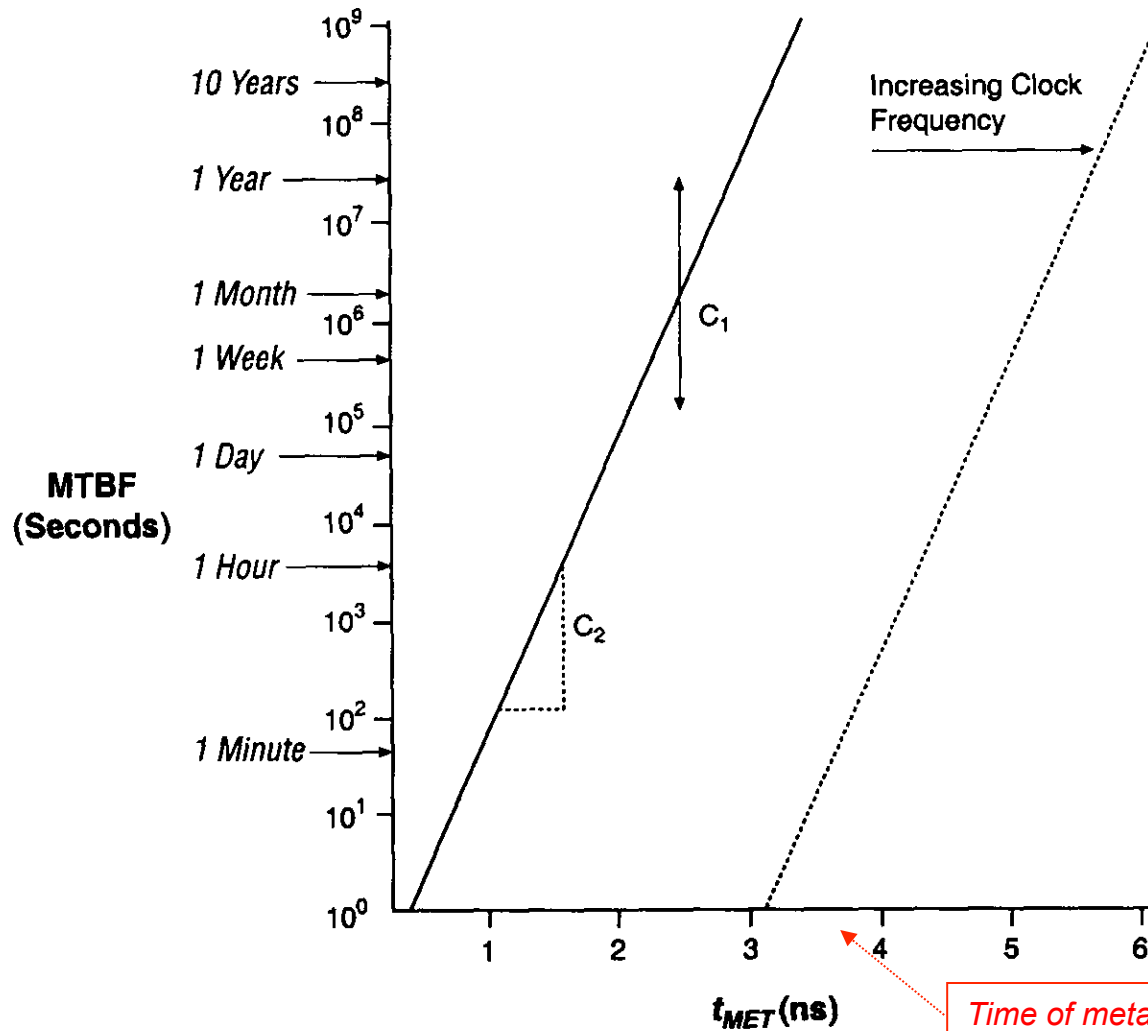
# Metastability - MTBF

- The  $C_1$  and  $C_2$  constants depend on the device process and operating conditions. Determined empirically.
- $f_{CLK}$  is the clock frequency of the clock domain receiving the asynchronous signal
- $f_{DATA}$  is the toggling frequency of the asynchronous input data signal. Faster clock frequencies and faster-toggling data reduce (or worsen) the MTBF.
- The  $t_{MET}$  parameter is the available metastability settling time, or the timing slack available beyond the register's  $t_{CO}$ , for a potentially metastable signal to resolve to a known value.

$$MTBF = \frac{e^{t_{MET}/C_2}}{C_1 \cdot f_{CLK} \cdot f_{DATA}}$$

Courtesy Altera

# MTBF vs. $T_{MET}$



$$C_2 = \frac{V \ln(MTBF)}{V t_{MET}}$$

$$C_1 = \frac{e^{(C_2 \times t_{MET})}}{MTBF \times f_{CLOCK} \times f_{DATA}}$$

Courtesy Altera

## MTBF: Alternate definition

To avoid synchronizer failure wait long enough before using a synchronizer's output. Where "long enough", is the mean time between synchronizer failures and is several orders of magnitude longer than the designer's expected length of employment!

*John Wakerly*



© UFS, Inc.

# Phase Lock Loop Backup Material

# PLL Reading List

---

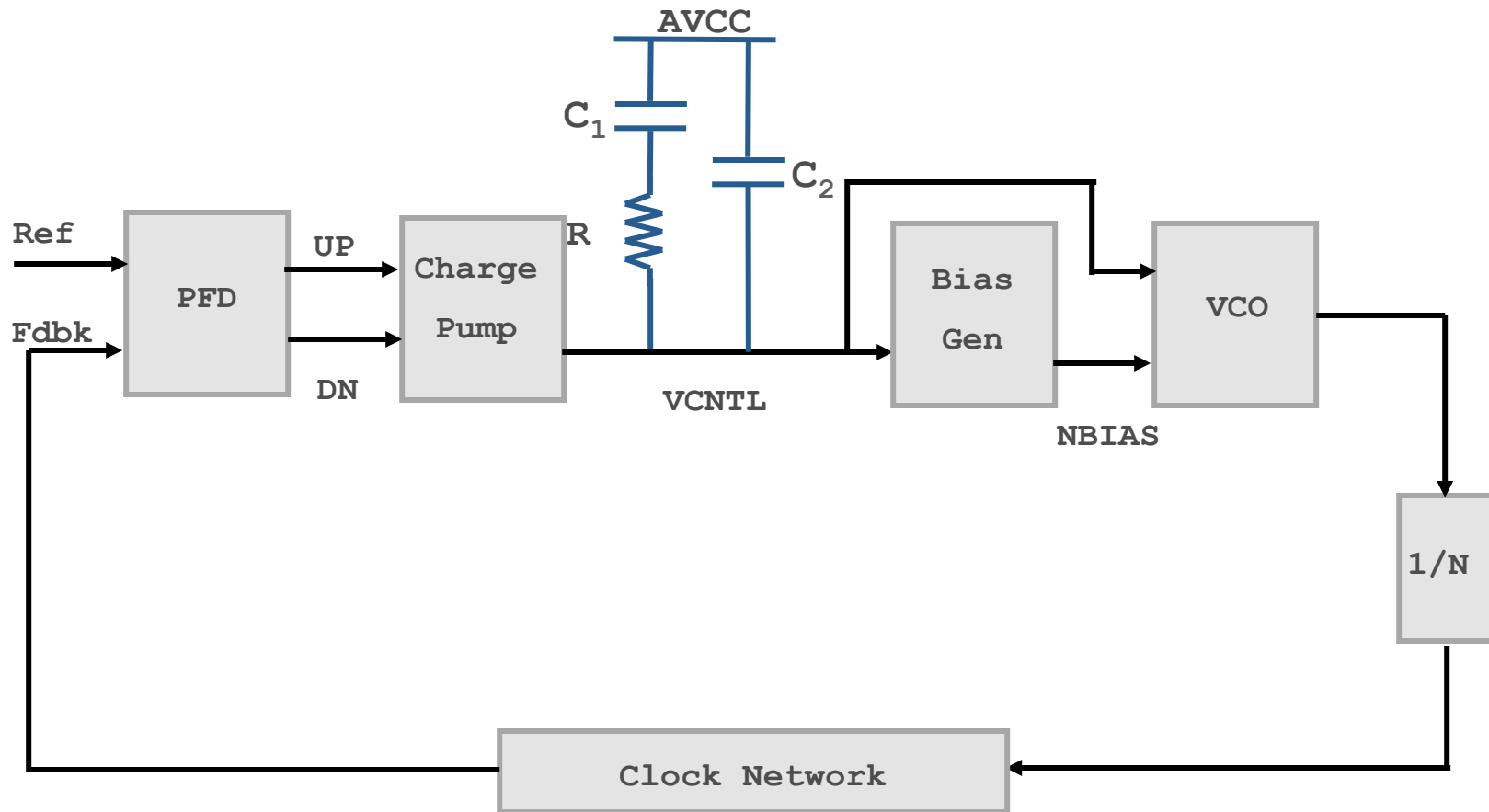
## ■ Books:

- “Phase Locked Techniques,” F.M. Gardner, 1979.
- “Phase Lock Loops, Design,” Simulation and Applications”, R. Best, 1999.
- “Phase Lock Loop Circuit Design,” Dan Wolaver, 1991.

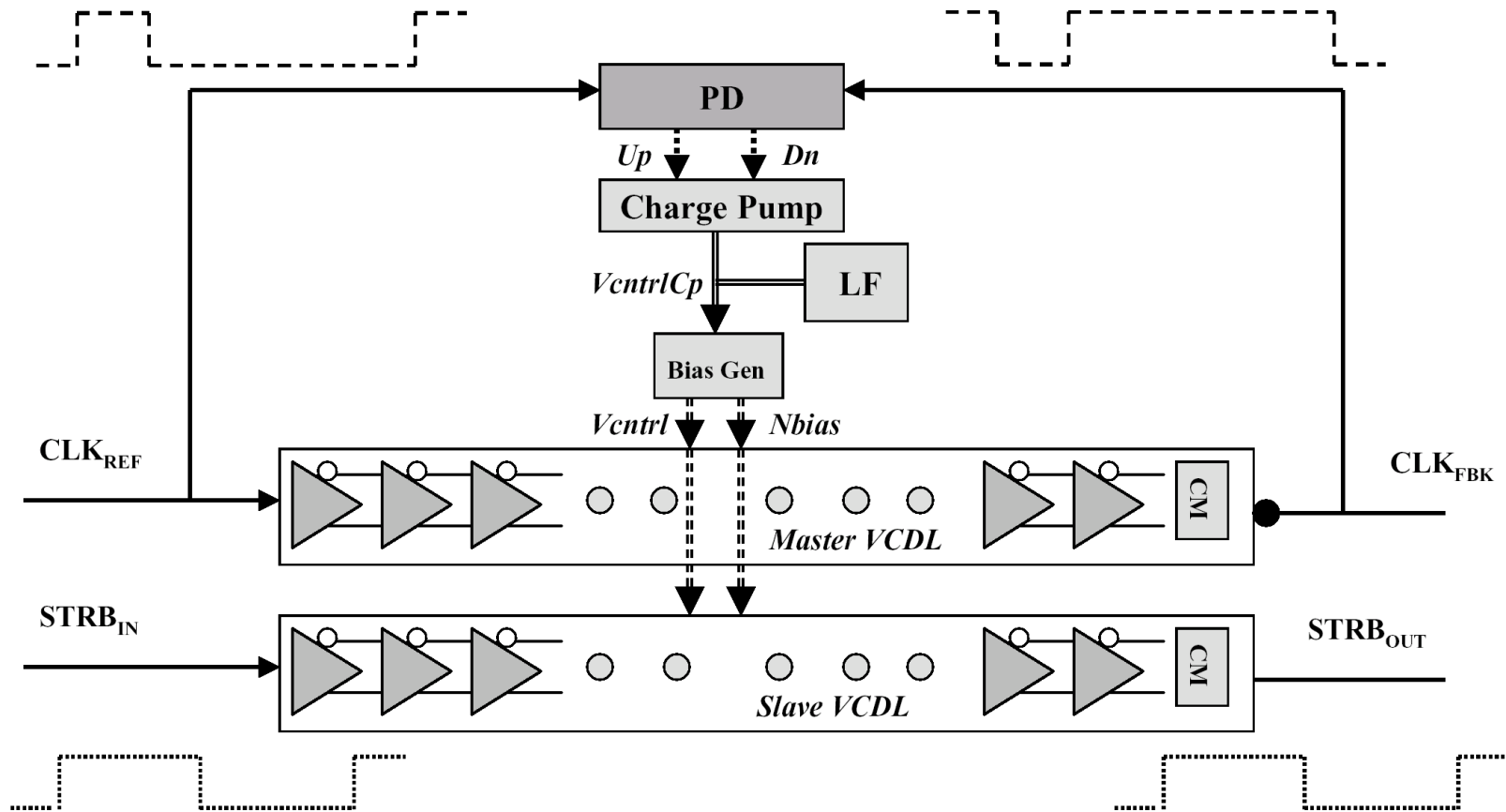
## ■ Articles:

- “Charge-Pump Phase-Lock Loops,” F.M. Gardner, IEEE Trans. On Comm., Vol. Com-28. No. 11, November 1980.
- “A PLL Clock Generator with 5 to 110 MHz of Lock Range for Microprocessors,” I.A. Young, IEEE Journal of Solid-State Circuits, Vol. 27, No. 11, November 1992.
- “Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques,” John G. Maneatis, IEEE Journal of Solid-State Circuits, Vol. 31, No. 11, November 1996.

# Typical 2nd order PLL Block Diagram



# DLL – Delay Locked Loop

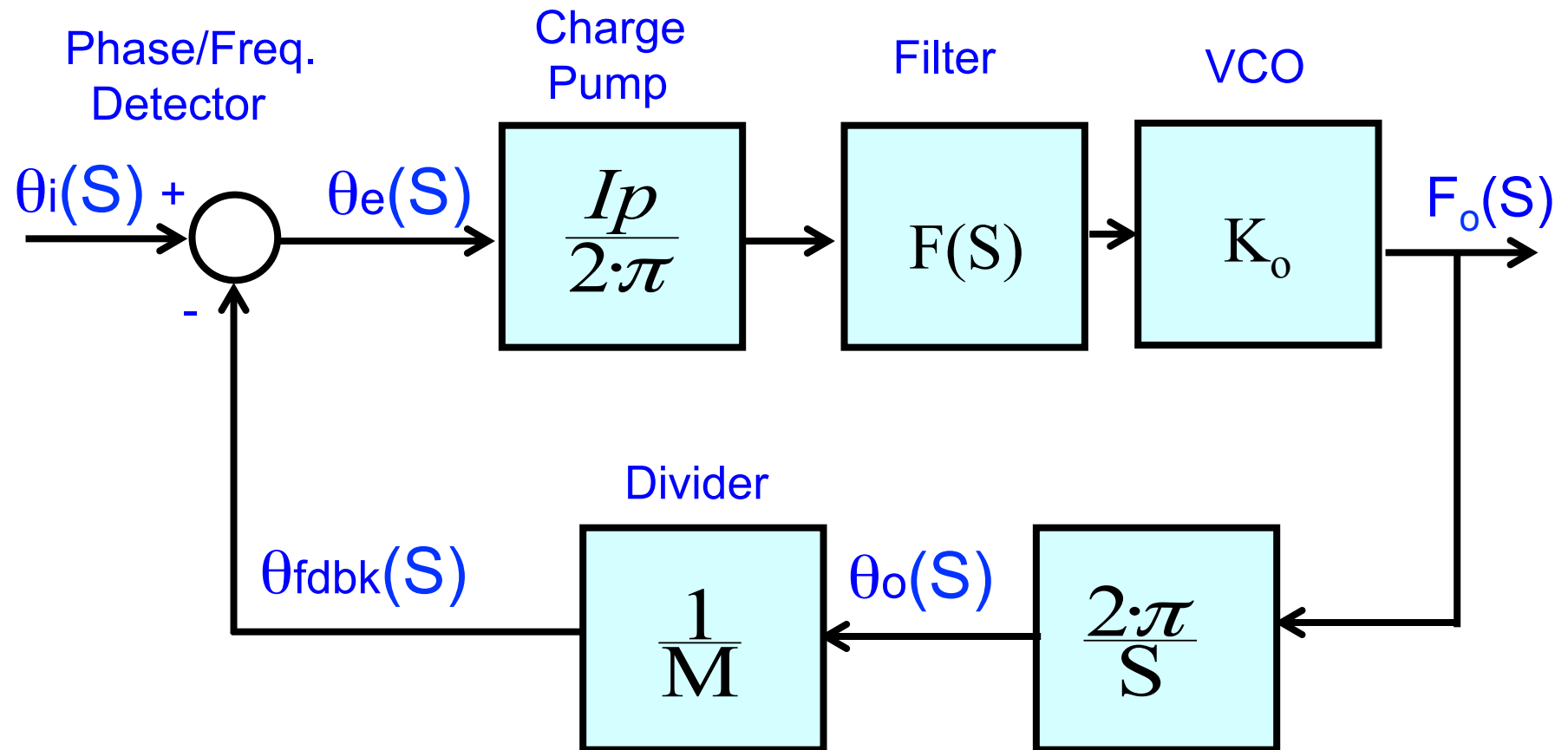


# PLL Loop Bandwidth vs. Jitter Trade-off

---

- **A low loop bandwidth is desirable to suppress reference clock jitter**
- **A high loop bandwidth is desirable to track power supply noise**
- **In general, PLL output jitter is mainly due to power supply noise rather than reference clock jitter (crystal oscillator based)**
- **Minimize peaking in closed-loop response**
- **Can use filtering to help reduce jitter due to external power supply noise**
  - **Dedicated power supply rail**
  - **LC Filter**
  - **On-die supply regulation**

# PLL Analysis: Locked State



# PLL Loop Equations

---

Analyzing the Transfer Function:  $\theta_o(S) / \theta_i(S)$

Loop Natural Frequency (Bandwidth):

$$F_n = \left( \frac{1}{2 \cdot \pi} \right) \cdot \sqrt{\frac{K_o \cdot I_p}{M \cdot C_1}}$$

Loop Damping (First Order Filter):

$$\zeta = \pi \cdot R \cdot C_1 \cdot F_n$$

# PLL Loop Analysis

---

- Higher VCO Gain ( $K_o$ )  $\rightarrow$  More Difficult to Stabilize Loop.
- Loop Bandwidth is chosen based on application and stability limitations.
- Desirable to minimize peaking effect in closed-loop frequency response.
- Under-damping  $\rightarrow$  more frequency overshoot, faster loop response.
- Over-damping  $\rightarrow$  less frequency overshoot, better stability, slower loop response.

# PLL Parameters

---

## ■ PLL Lock Time

- This is the time it takes for the PLL to achieve steady-state lock condition from power-up.
- Function of Loop Parameters → VCO gain and loop bandwidth.
- It is determined by a one-shot measurement of the PLL output clock when the PLL is powered-up.
- This time is important if the reference clock is dynamically changed during operation, e.g., changing to slow speed clock to reduce power.

## ■ PLL Tracking Range

- Determines the Min & Max VCO frequency of operation for stable PLL loop.
- Function of VCO gain characteristics.
- Determined by sweeping the PLL input frequency and measuring the PLL output frequency.

# PLL Parameters

---

## ■ PLL Frequency Overshoot

- It is a measure of PLL stability.
- It is determined by the Filter and the VCO Gain Parameters
- It is a one-shot measurement of the PLL output clock when the PLL is powered-up.
  - Need to limit the overshoot to prevent erroneous circuit operation.

## ■ PLL Output Jitter

- Jitter performance is a strong function of PLL loop parameters along with many other sources of noise.
- Reference clock jitter and power supply noise along with the loop bandwidth also impact PLL output jitter
- Must consider both high and low frequency content of jitter:
  - Low frequency jitter → jitter is measured over many clock cycles.
  - High frequency jitter → edge-to-edge variations

# VCO - Voltage Controlled Oscillator

---

- **There are a number of requirements placed on voltage controlled oscillators used in clock generation circuits:**
  - Phase stability
  - Broad tuning (tracking) range
  - Linearity of frequency vs. control voltage
  - Large gain factor  $K_o$
- **These requirements are usually in direct conflict with each other.**
  - To obtain good wideband features phase stability will be reduced.

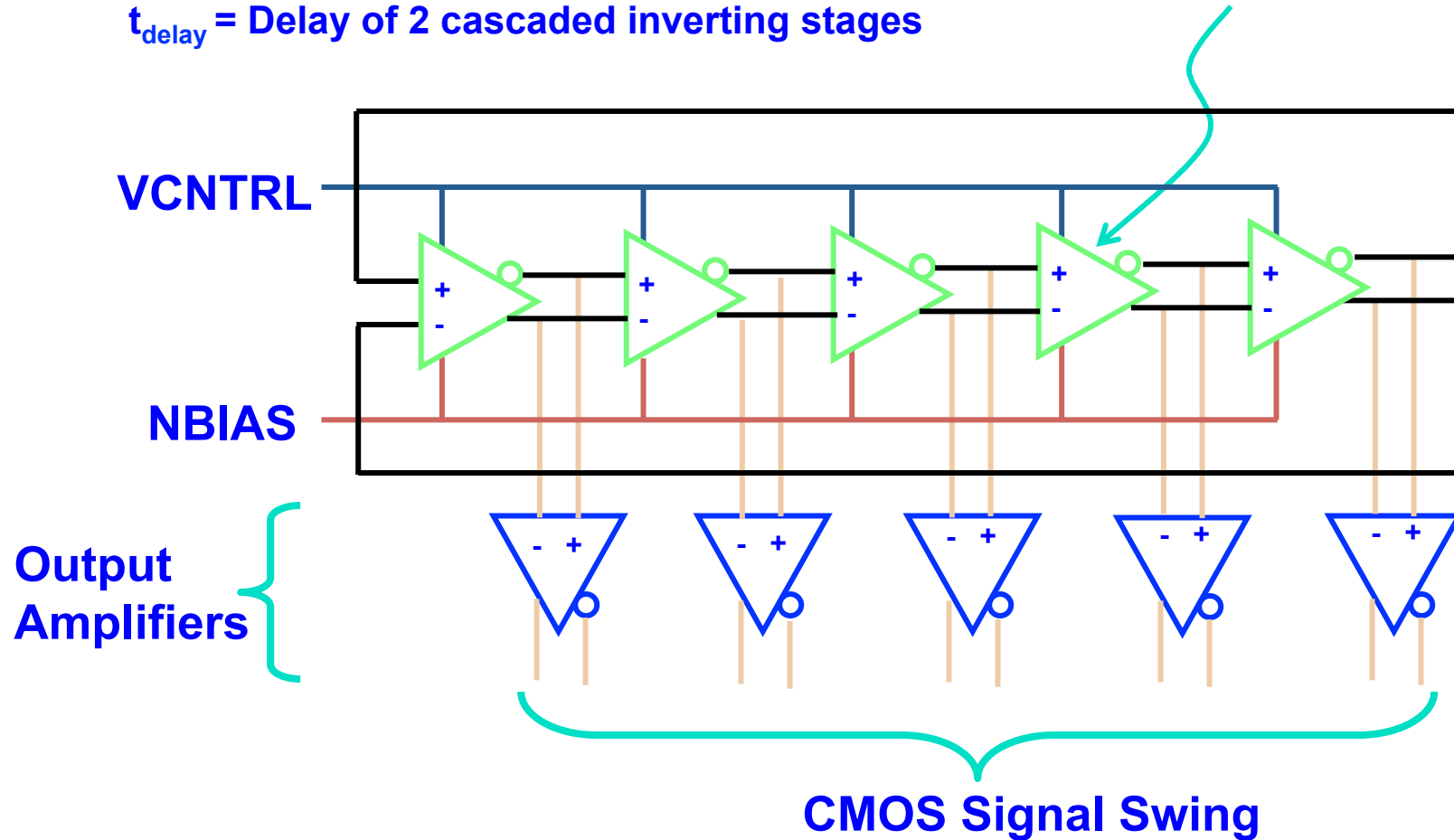
# Differential VCO

$$\text{Output Period} = N * t_{\text{delay}}$$

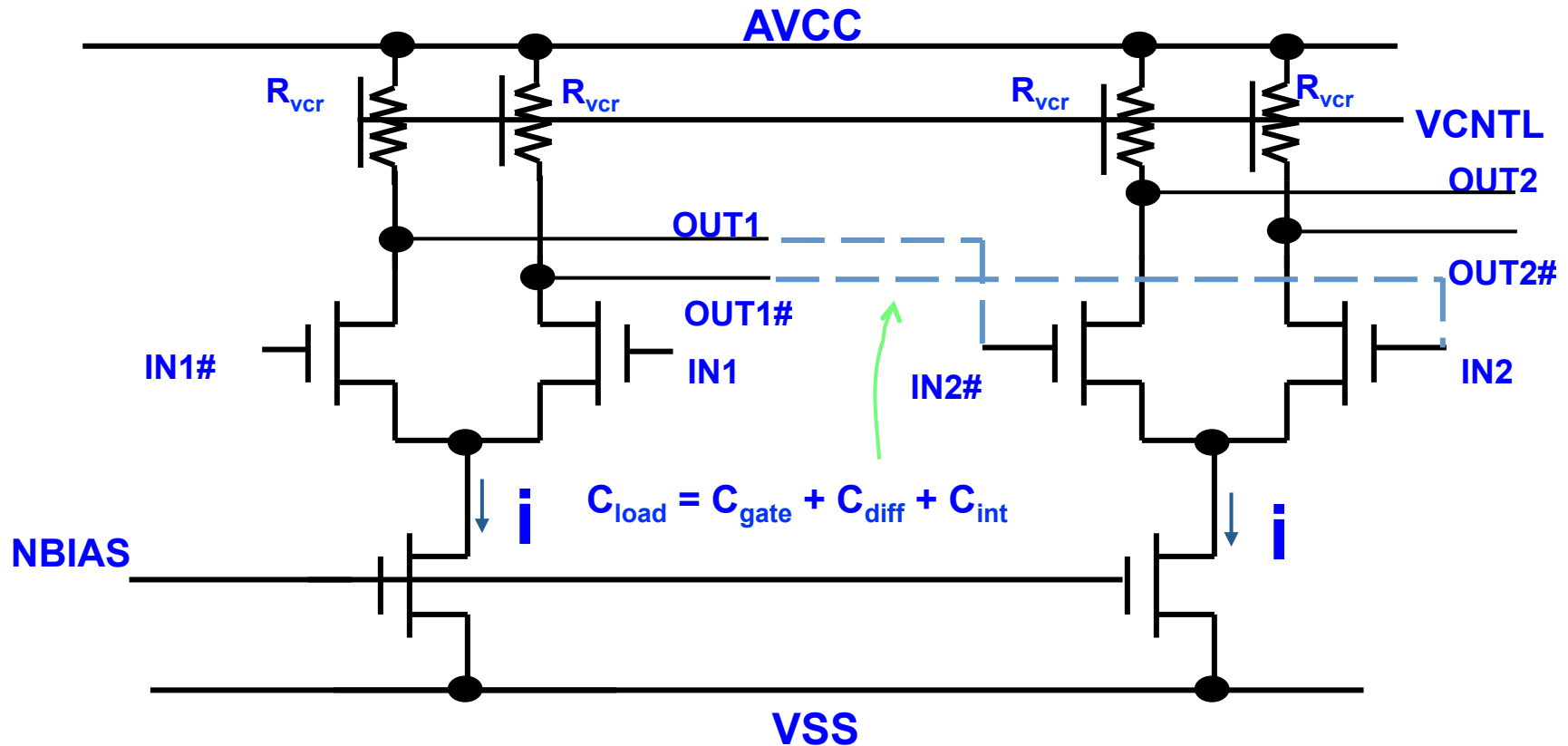
$N$  = Number of Delay Stages

$t_{\text{delay}}$  = Delay of 2 cascaded inverting stages

Differential  
Delay Stage



# Differential Delay Stage



## Advantages:

Differential signals more immune to noise

Insensitive to switching trip-point inaccuracy

# VCO Frequency of Oscillation

---

$$i = C \cdot \frac{\Delta V}{\Delta T}$$

$$t_{\text{delay}} = C \cdot \frac{\Delta V}{i}$$

$$\textit{Frequency} = \frac{i}{C \cdot \Delta V \cdot N}$$

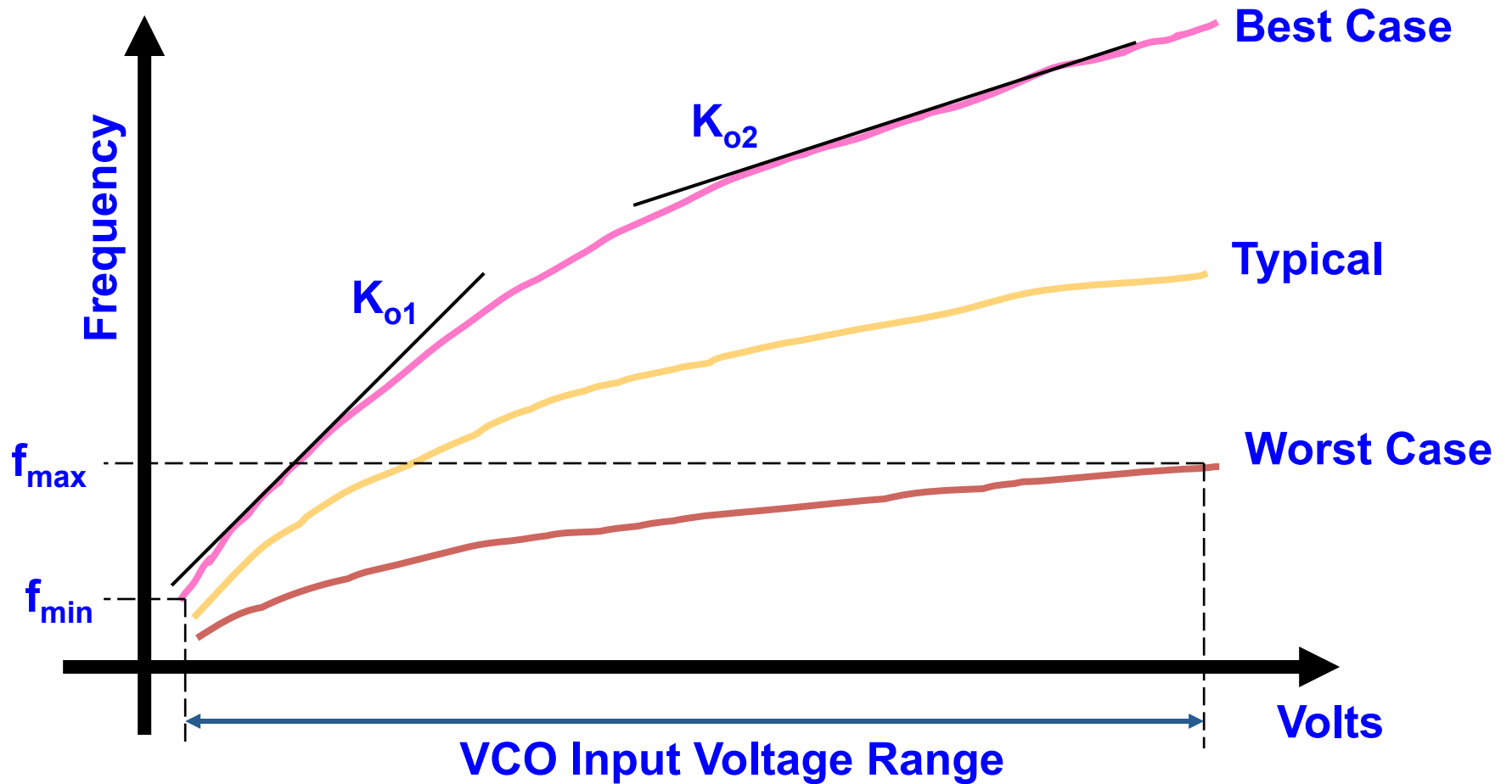
where **N = number of delay stages**

**$\Delta V$  = Peak-Peak Voltage Swing (AVCC-Vcntrl)**

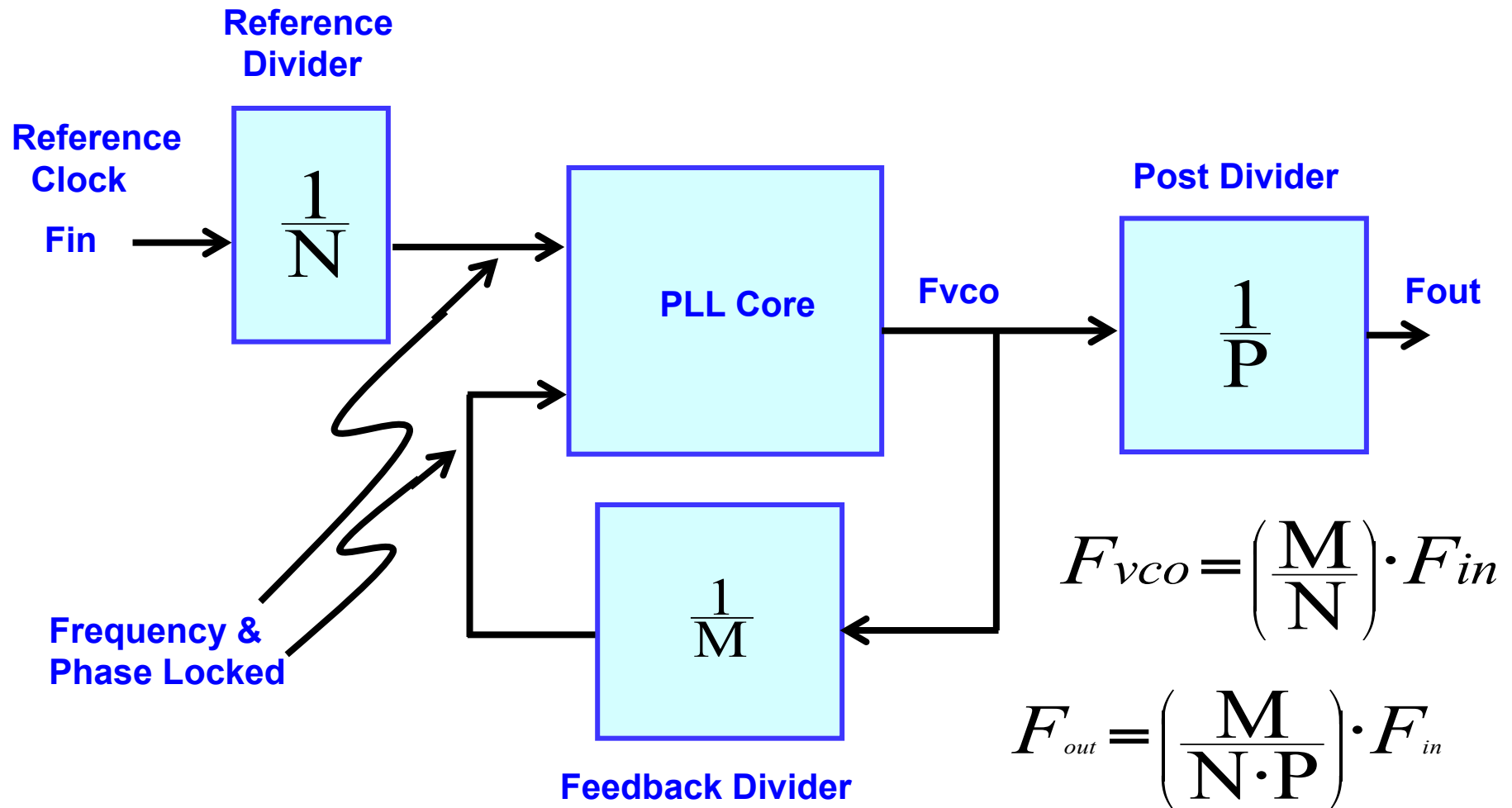
**C = total capacitance on output node**

**i = input current to Ring Osc. (Tail Current)**

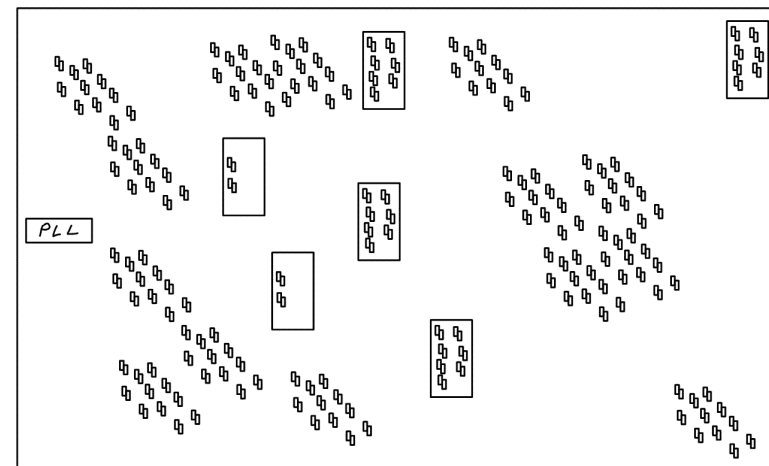
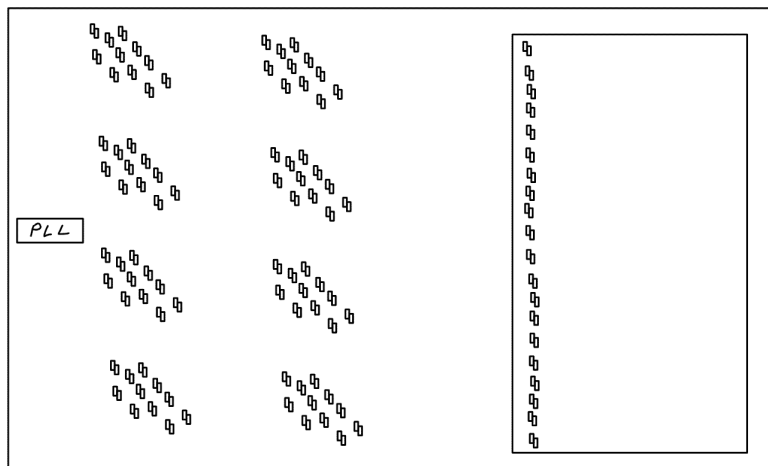
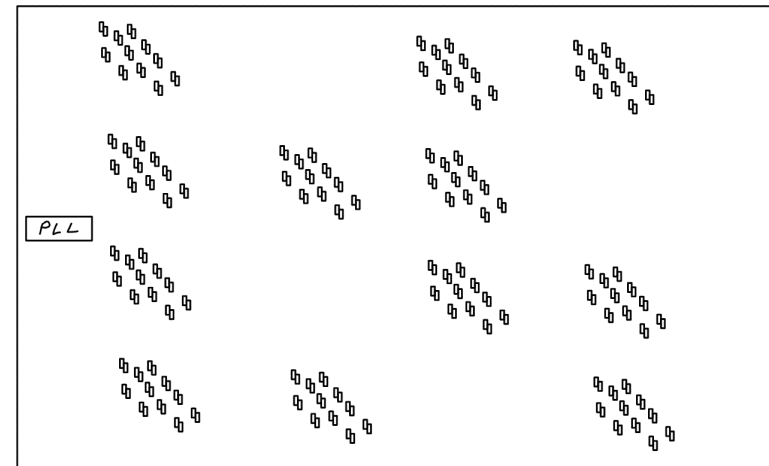
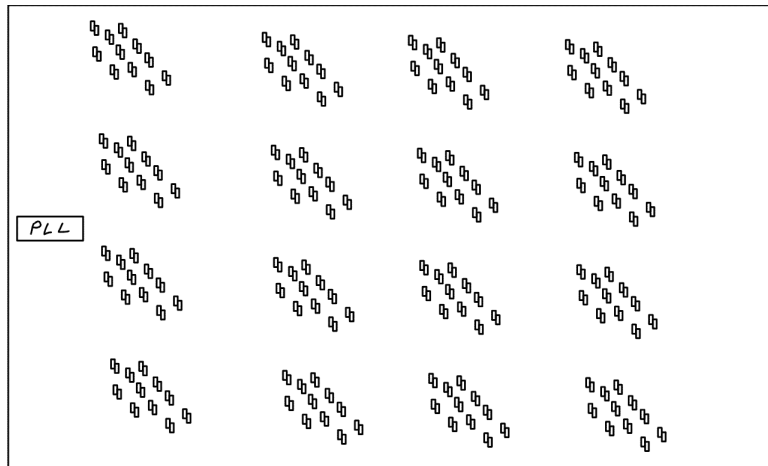
# VCO Transfer Function



# PLL Block Diagram



# Clock Distribution Design: Do it Yourself



Assume chip is 10mm x 5mm. Load cap equivalent to 4 minimum sized inverters. Choose metal Rsh and Cap/um from technology. Use clustering to decide distribution style, lowest skew, lowest power solution.