Lecture 9:
Clocking, Clock Skew, Clock Jitter, Clock Distribution and some FM

Mark McDermott
Electrical and Computer Engineering
The University of Texas at Austin

Why Clocking?

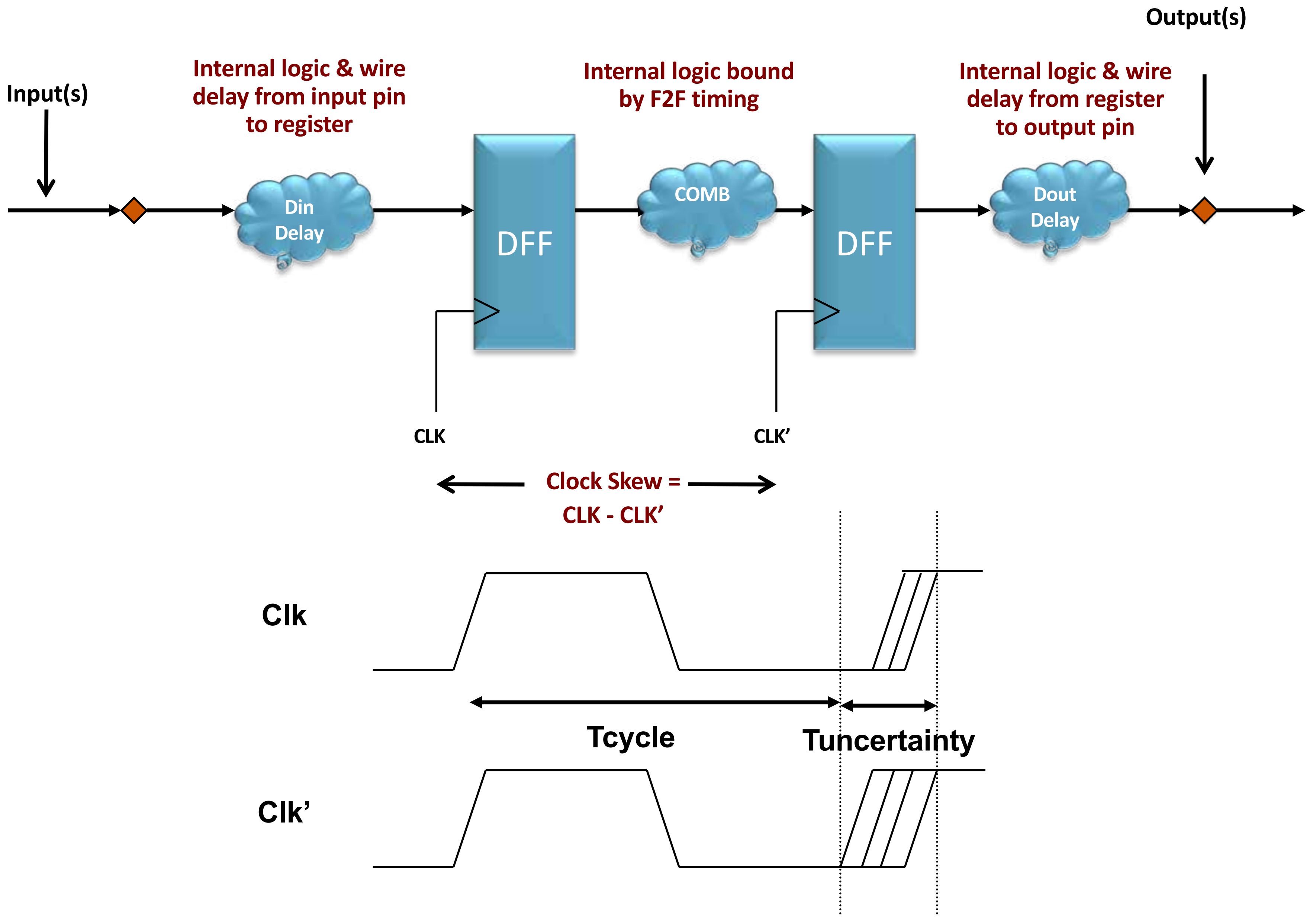
- Synchronous systems use a clock to keep operations in sequence
  - Distinguish *this cycle* from *previous cycle* or *next cycle*
  - Determine speed at which machine operates

- Clock must be distributed to all the sequencing elements
  - Flip-flops and latches

- Also distribute clock to other elements
  - Domino circuits and memories

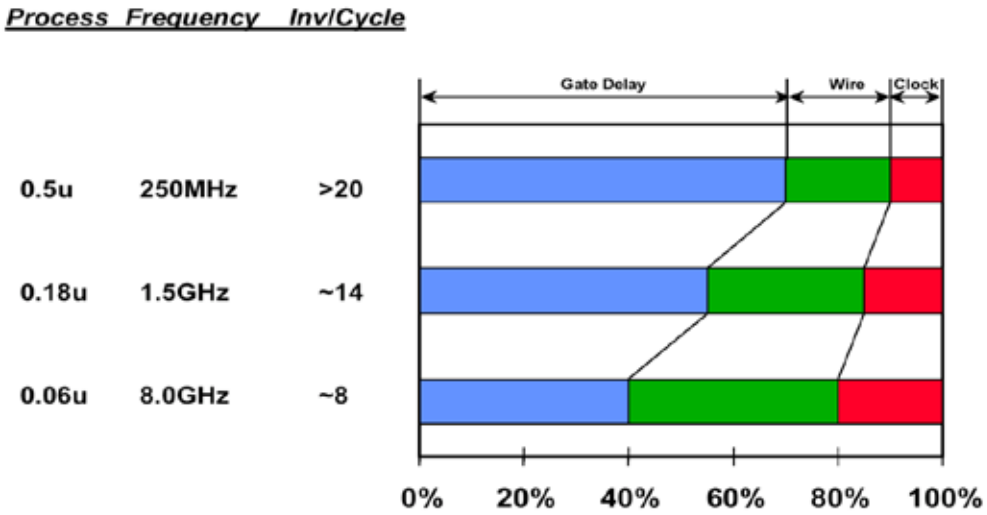
- Clocking overhead % increases as the frequency is increased.

Logic Transactions and Clock Dependence

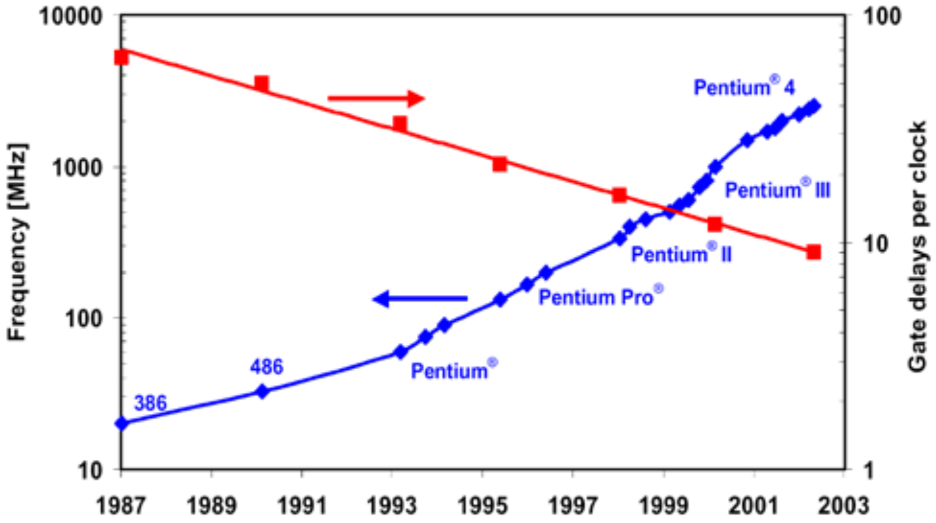

Output(s)
Input(s)
Internal logic & wire delay from input pin to register
Internal logic bound by F2F timing
Internal logic & wire delay from register to output pin
Din Delay
DFF
COMB
DFF
Dout Delay
CLK
CLK'
Clock Skew = CLK - CLK'
Clk
Tcycle
Tuncertainty
Clk'

Clocking Overhead per Technology Generation

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



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

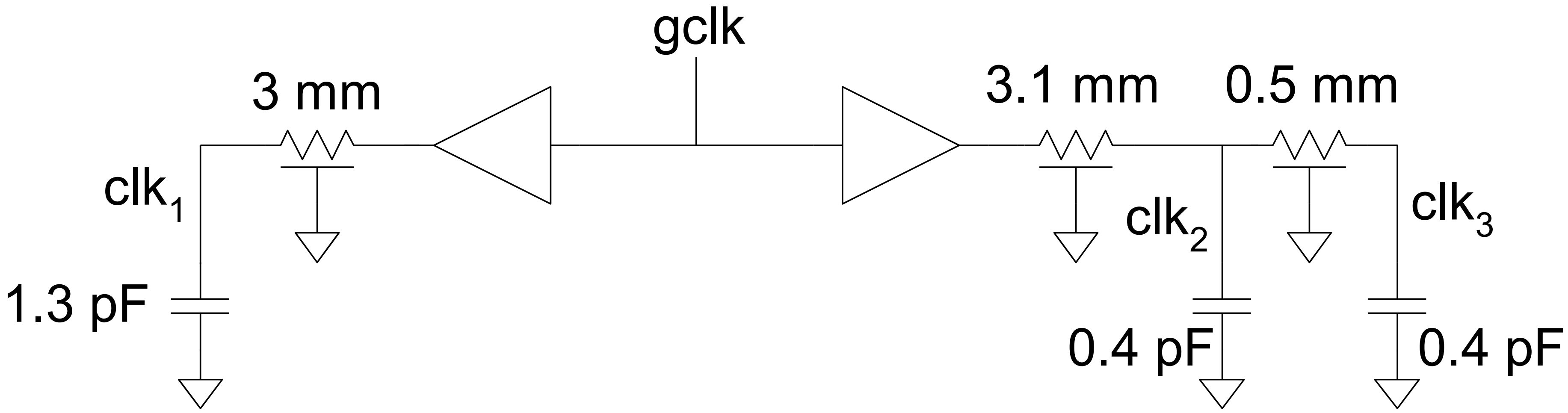


Clock Distribution

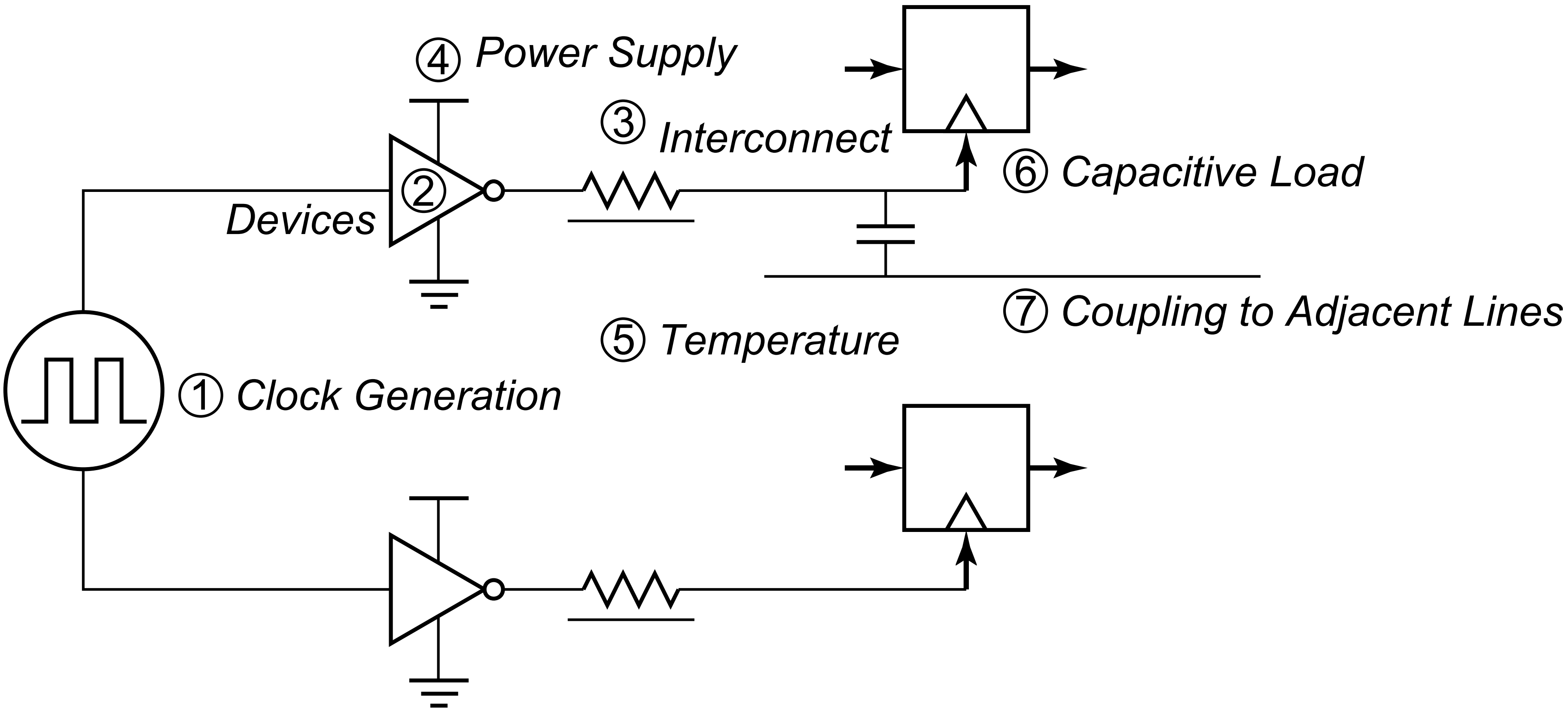
- On a small chip, the clock distribution network is just a wire
  - And possibly an inverter for clk’
- On practical chips, the RC delay of the wire resistance and gate load is very long
  - Variations in this delay cause clock to get to different elements at different times
  - This is called clock skew
- Most chips use repeaters to buffer the clock and equalize the delay
  - Reduces but does not eliminate skew

Example

- **Skew comes from differences in gate and wire delay**
  - With right buffer sizing, clk_1 and clk_2 could ideally arrive at the same time.
  - But power supply noise changes buffer delays
  - clk_2 and clk_3 will always see RC skew



Sources of Clock Uncertainties

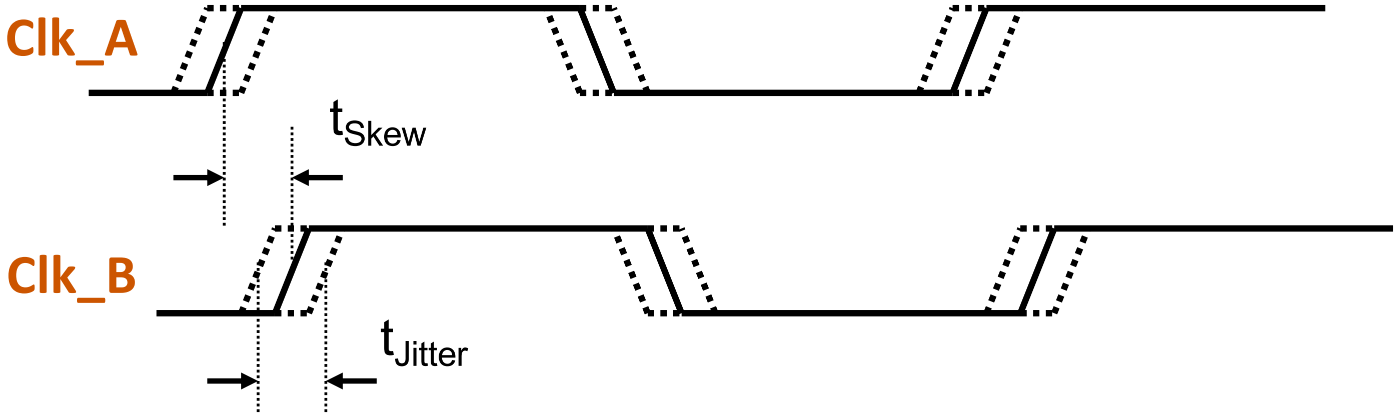


© J. Rabaey, 2002

Clock Non-idealities

- Clock skew
  - Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}
- Clock jitter
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- Variation of the pulse width
  - Important for level sensitive clocking

Clock Skew and Jitter

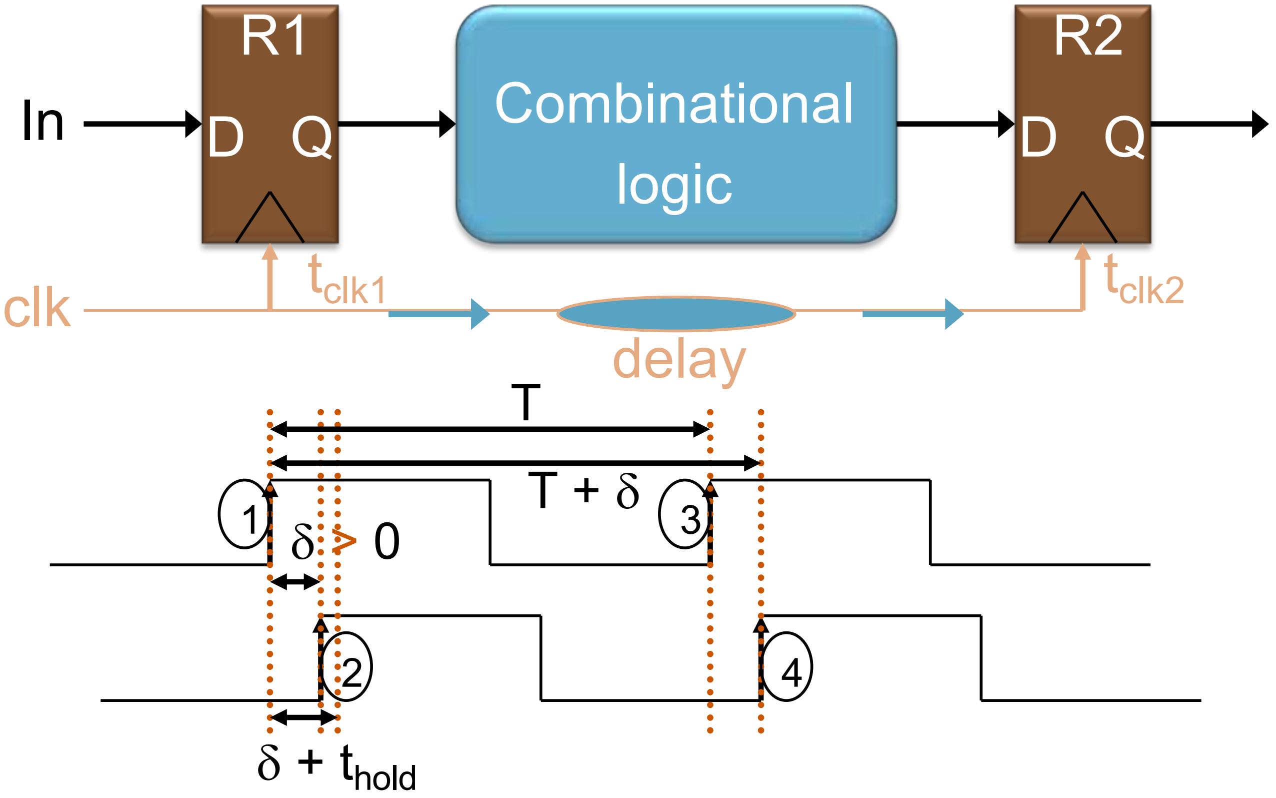


Both skew and jitter affect the effective cycle time

Only skew affects the race margin assuming jitter tracks in the same direction

Positive Clock Skew

Clock and data flow in the same direction



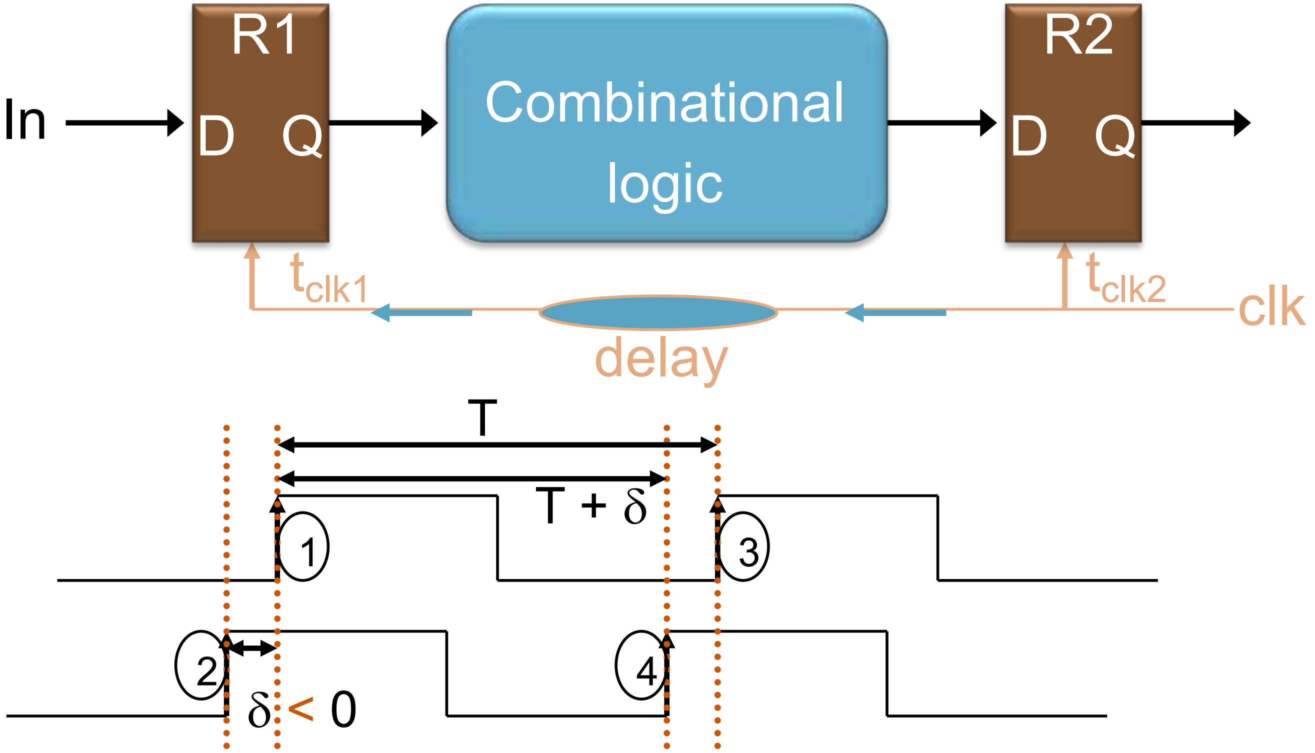
$$T: \quad T + \delta \geq t_{c\text{-}q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c\text{-}q} + t_{plogic} + t_{su} - \delta$$

$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

δ > 0: Improves performance, but makes t_{hold} harder to meet. If t_{hold} is not met (race conditions), the circuit malfunctions independent of the clock period!

Negative Clock Skew

Clock and data flow in opposite directions



$$T: \quad T + \delta \geq t_{c\text{-}q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c\text{-}q} + t_{plogic} + t_{su} - \delta$$

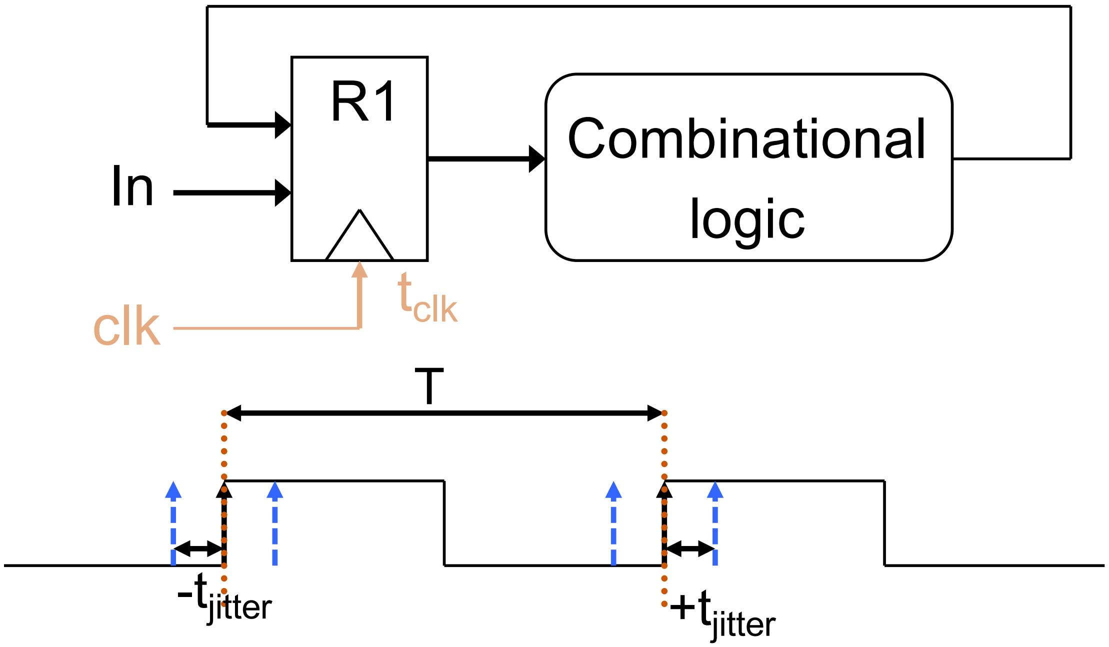
$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

δ < 0: Degrades performance (t_{setup}), but t_{hold} is easier to meet (eliminating race conditions)

© J. Rabaey, 2002

Clock Jitter

Jitter causes T to vary on a cycle-by-cycle basis



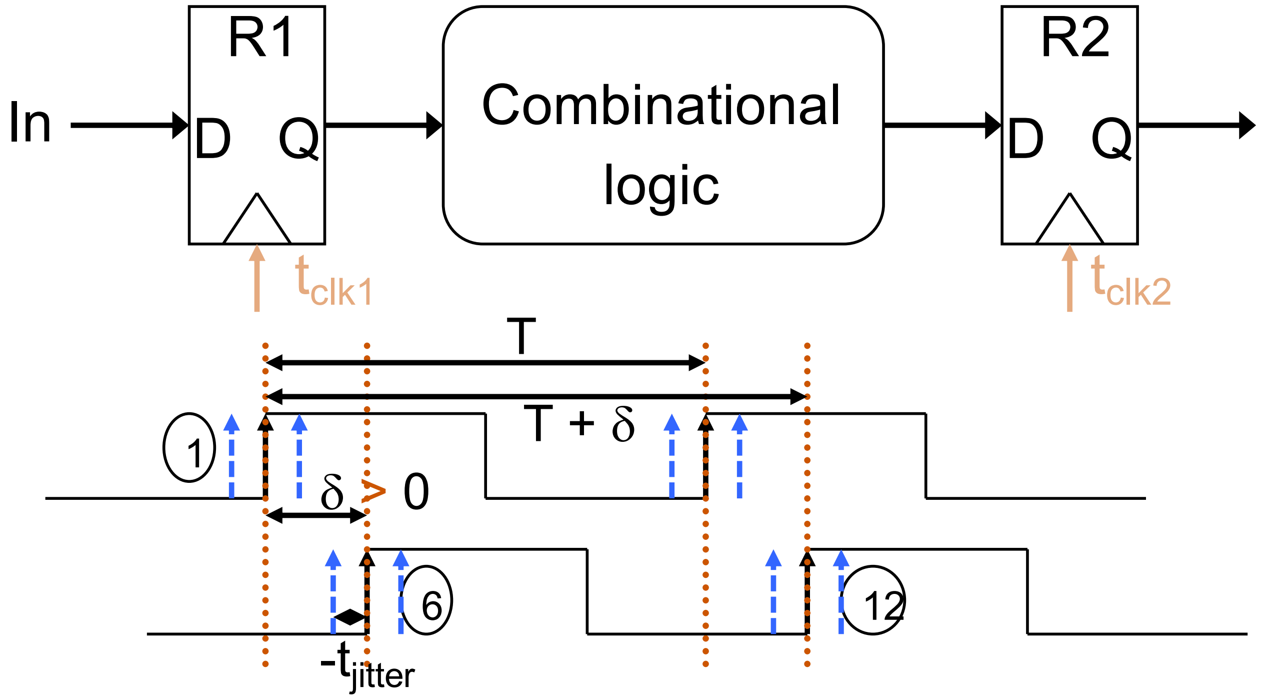
T : $T - 2t_{jitter} \geq t_{c\text{-}q} + t_{plogic} + t_{su}$ so $T \geq t_{c\text{-}q} + t_{plogic} + t_{su} + 2t_{jitter}$

Jitter directly reduces the performance of a sequential circuit

© J. Rabaey, 2002

Combined Impact of Skew and Jitter

Constraints on the minimum clock period (δ > 0)



$T \geq t_{c\text{-}q} + t_{plogic} + t_{su} - \delta + 2t_{jitter}$

$t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta - 2t_{jitter}$

δ > 0 with jitter: Degrades performance, and makes t_{hold} even *harder* to meet. (The acceptable skew is reduced by jitter.)

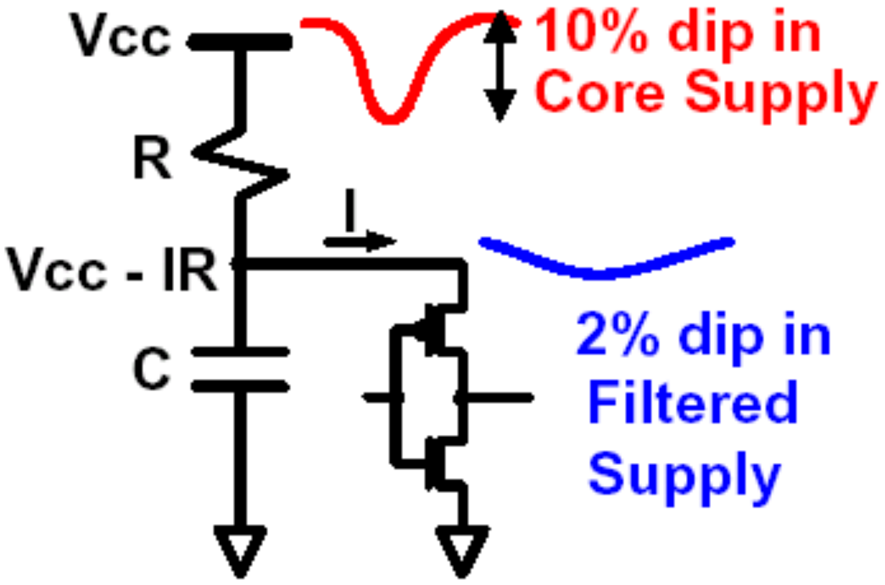
Clock Skew Solutions

- **Reduce clock skew**
  - Careful clock distribution network design
  - Plenty of metal wiring resources
  - Supply Filtering
  - Active de-skewing

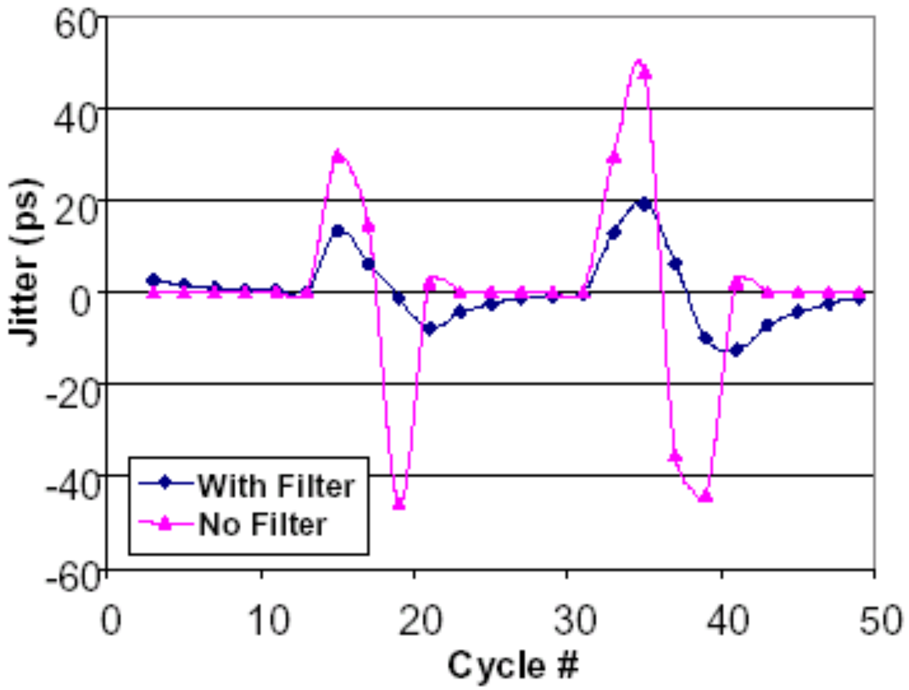
- **Analyze clock skew**
  - Only budget actual, not worst case skews
  - Local vs. global skew budgets

- **Tolerate clock skew**
  - Choose circuit structures insensitive to skew
  - Take advantage of "useful skew"

Jitter reduction using local supply filtering

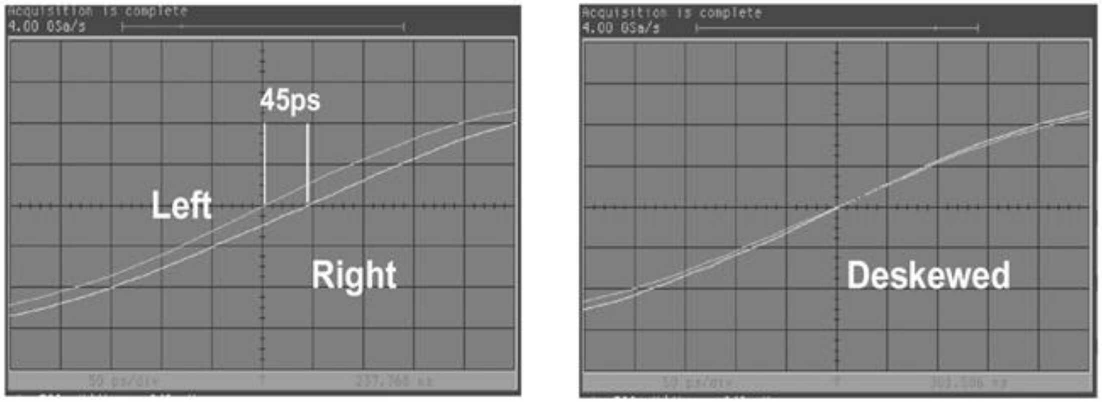


Kurd, JSSC-2001

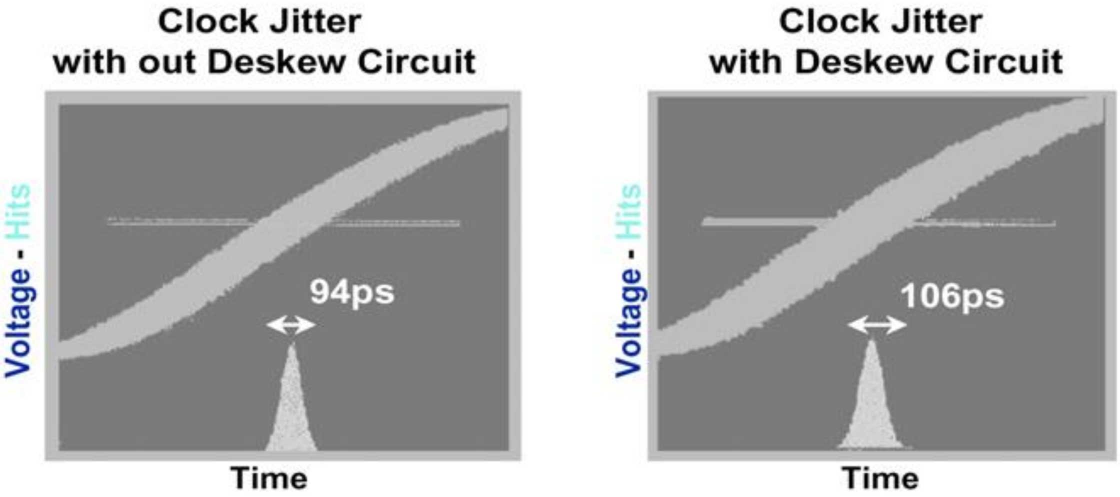


Active Clock De-skewing

- **Active clock de-skewing 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

Clock Distribution

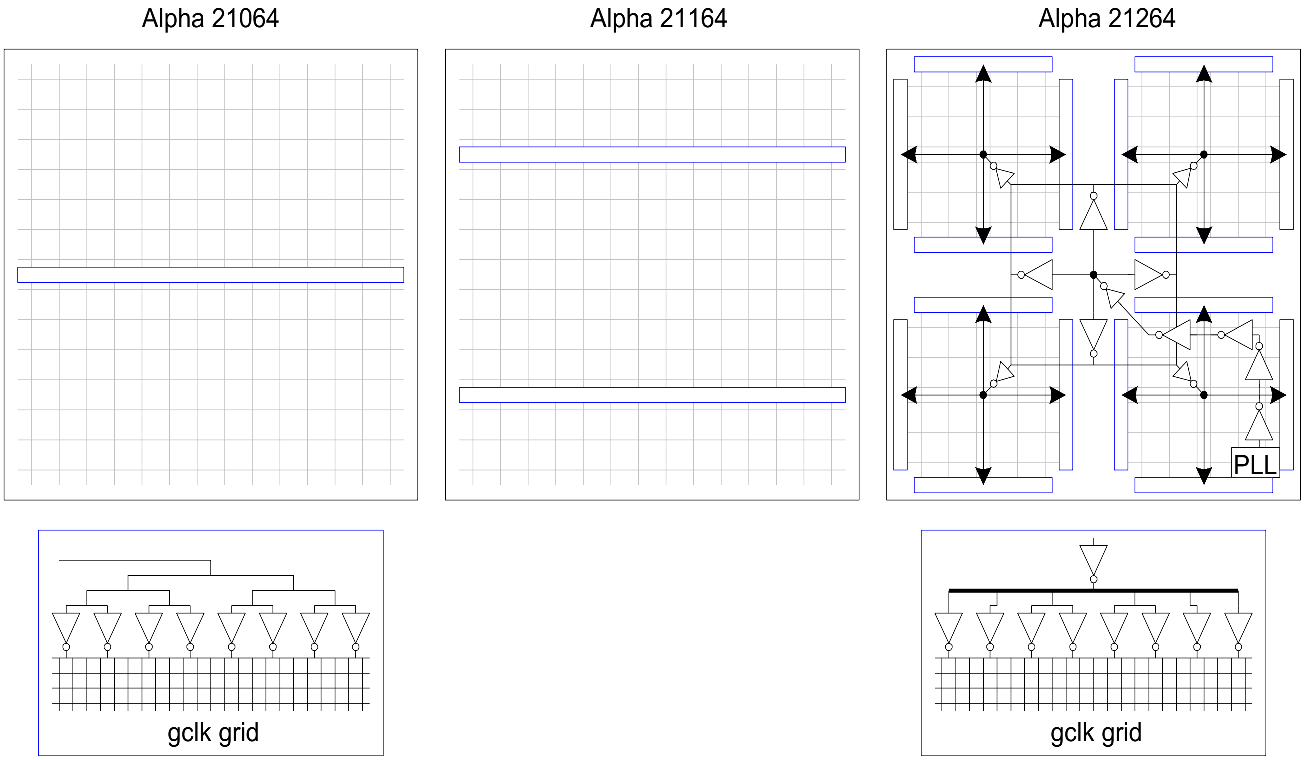
- There are four basic types of clock distribution networks used in high performance processor designs:
  - Tree: IBM and Freescale PowerPC, HP PA-RISC
  - Grid: SPARC, Alpha
  - Serpentine: Pentium-III
  - Spine: Alpha, Pentium-4
- Each technique has advantages and disadvantages:

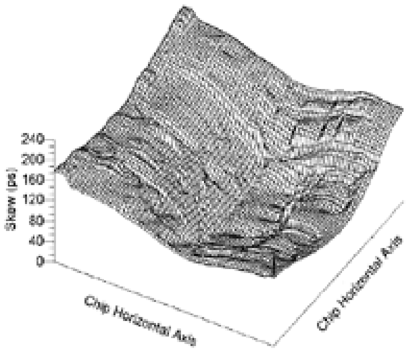
	Wire Cap	Delay	Skew
Grid	High – 15x	Low – sub100 ps	Low-Med
Trees	Low – 1x	High – 100's ps	Low
Serpentine	Very High – 30x	High – 100's ps	Low
Spine	High – 10x	Low-sub100ps	Med

Clock Grids

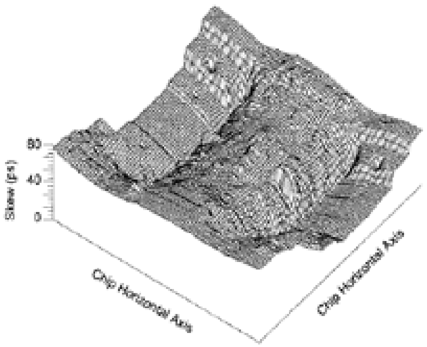
- Use grid on two or more levels to carry clock
- Make wires wide to reduce RC delay
- Ensures low skew between nearby points
- But possibly large skew across die

Alpha Clock Grids

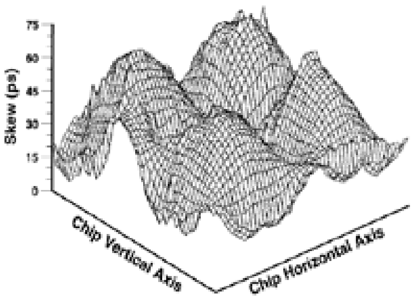




Alpha 21064



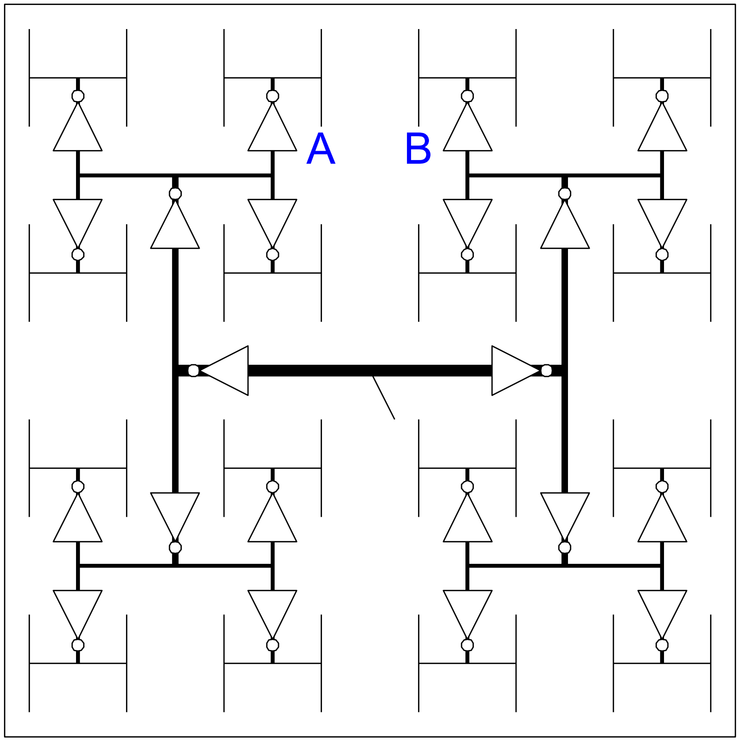
Alpha 21164



Alpha 21264

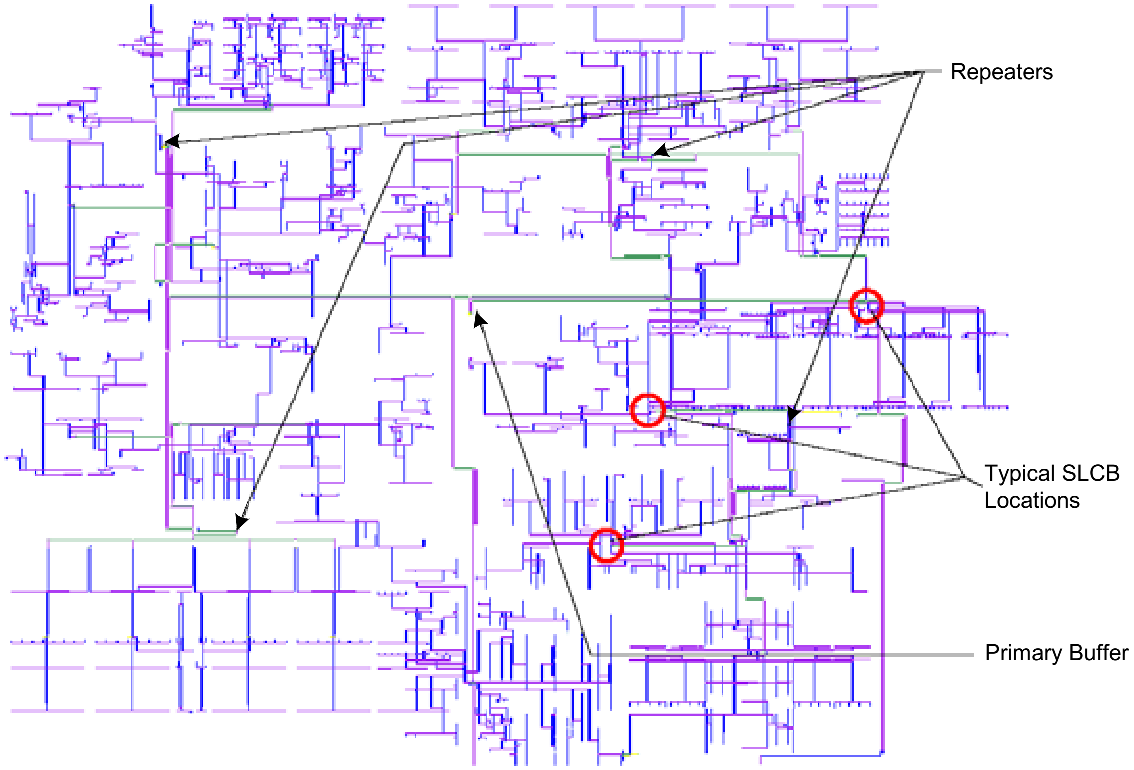
H-Trees

- Fractal structure
  - Gets clock arbitrarily close to any point
  - Matched delay along all paths
- Delay variations cause skew
- A and B might see big skew



Itanium 2 H-Tree

- **Four levels of buffering:**
  - Primary driver
  - Repeater
  - Second-level clock buffer
  - Gater
- **Route around obstructions**



Hybrid Networks

- Use H-tree to distribute clock to many points
- Tie these points together with a grid

- Ex: IBM Power4, PowerPC
  - H-tree drives 16-64 sector buffers
  - Buffers drive total of 1024 points
  - All points shorted together with grid

Dealing with Clock Skew and Jitter

- To minimize skew, balance clock paths using H-tree or matched-tree clock distribution structures.
- If possible, route data and clock in opposite directions; eliminates races at the cost of performance.
- The use of gated clocks to help with dynamic power consumption make jitter worse.
- Shield clock wires (route power lines – VDD or GND – next to clock lines) to minimize/eliminate coupling with neighboring signal nets.

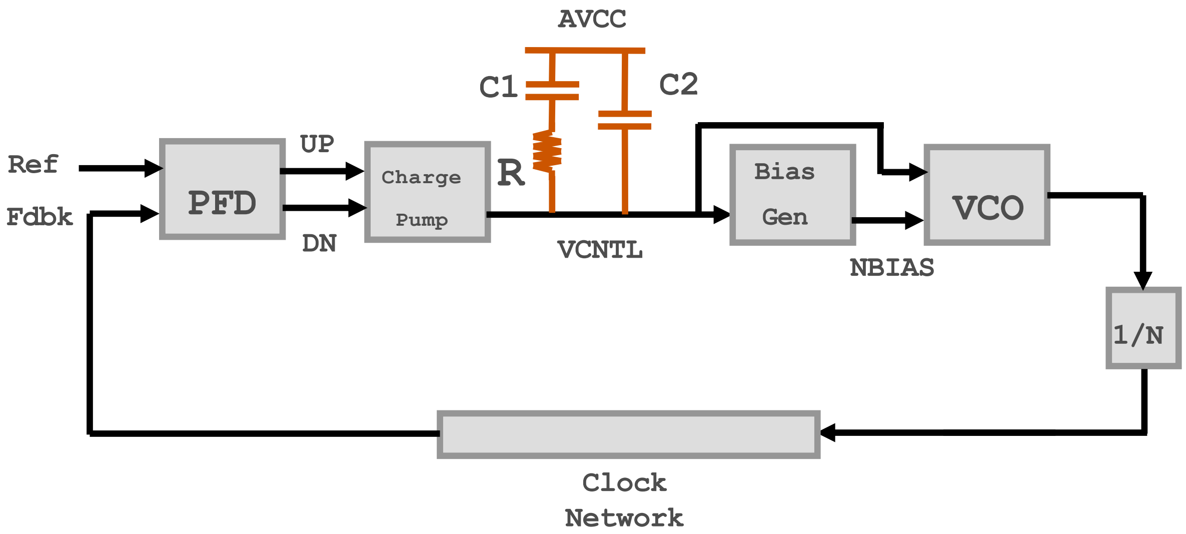
Dealing with Clock Skew and Jitter

- Use dummy fills to reduce skew by reducing variations in interconnect capacitances due to interlayer dielectric thickness variations.

- Beware of temperature and supply rail variations and their effects on skew and jitter. Power supply noise fundamentally limits the performance of clock networks.

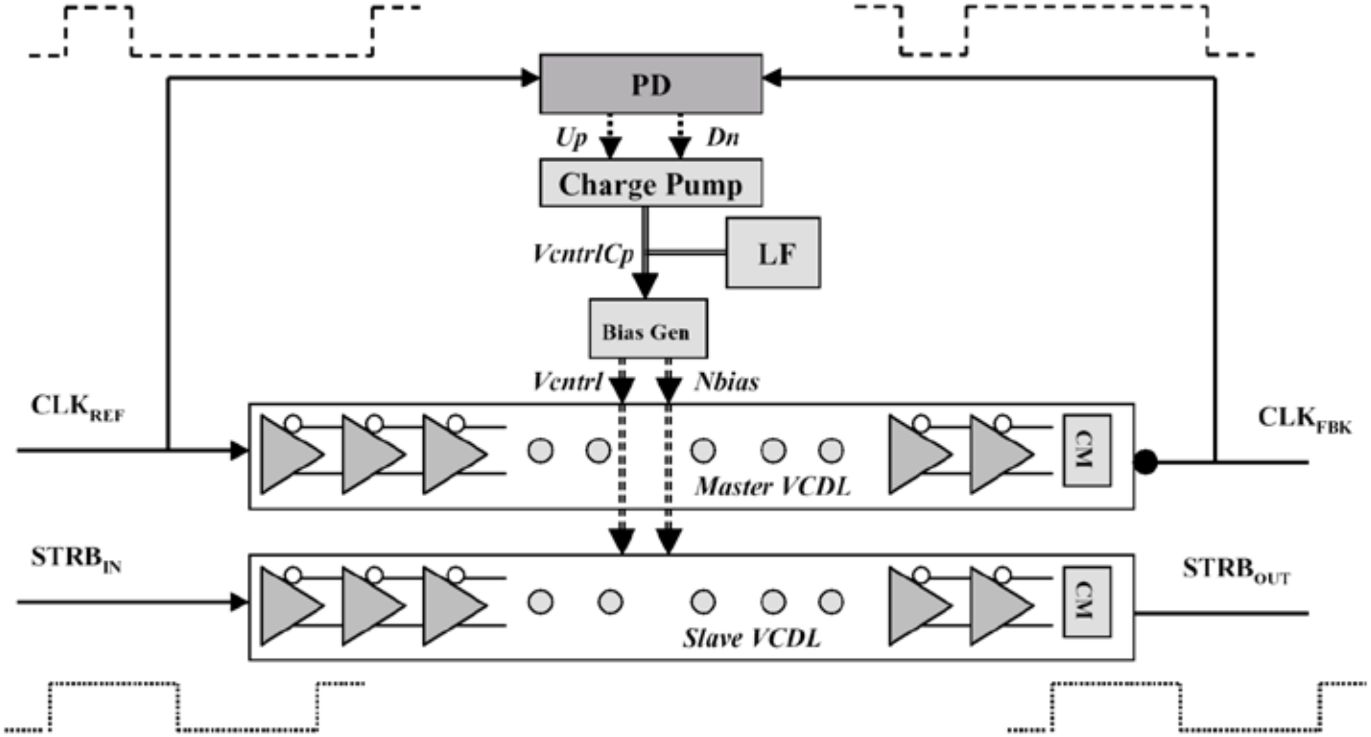
PLL Synchronization

- 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.

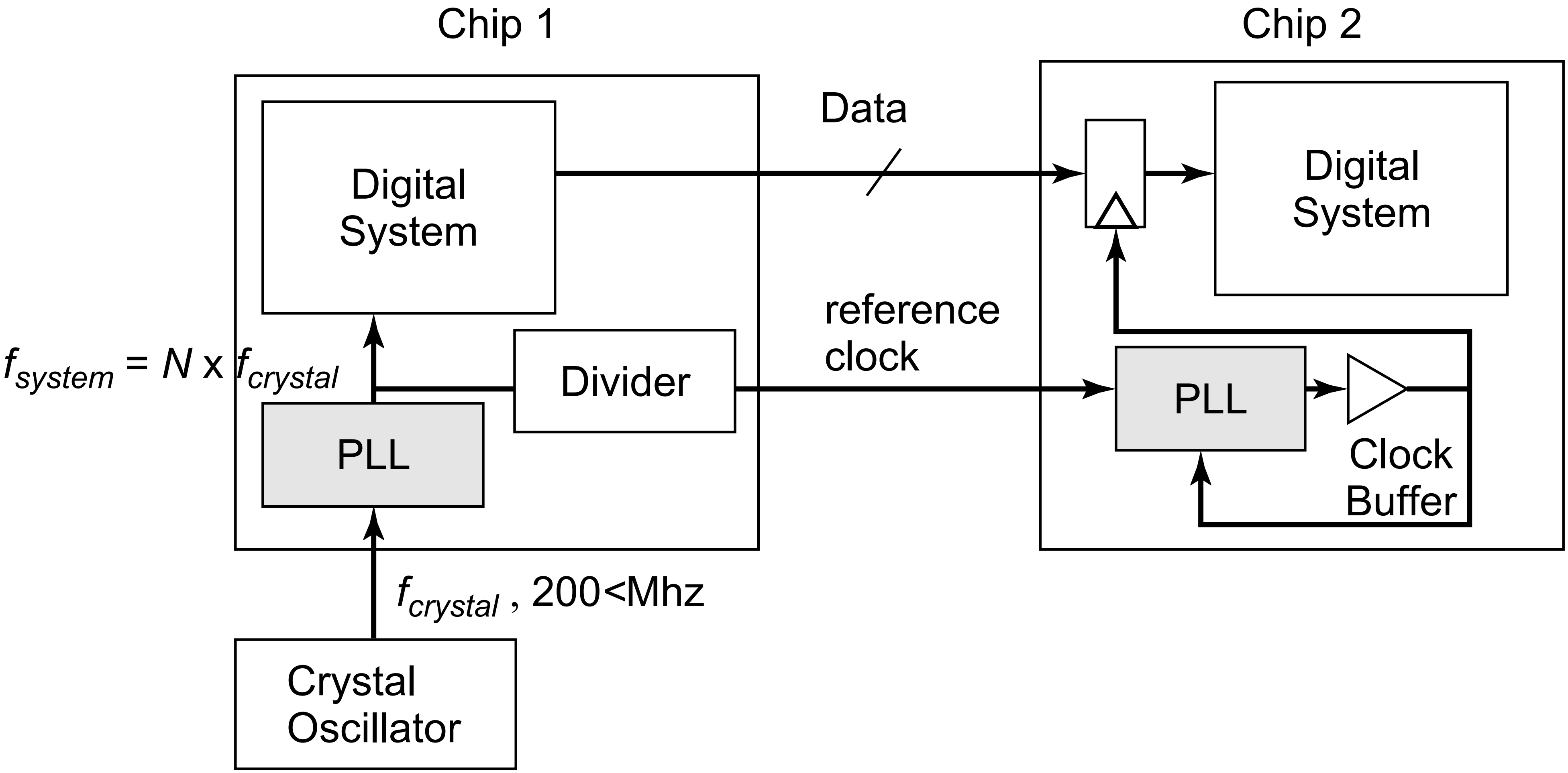


DLL Synchronization

- 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.

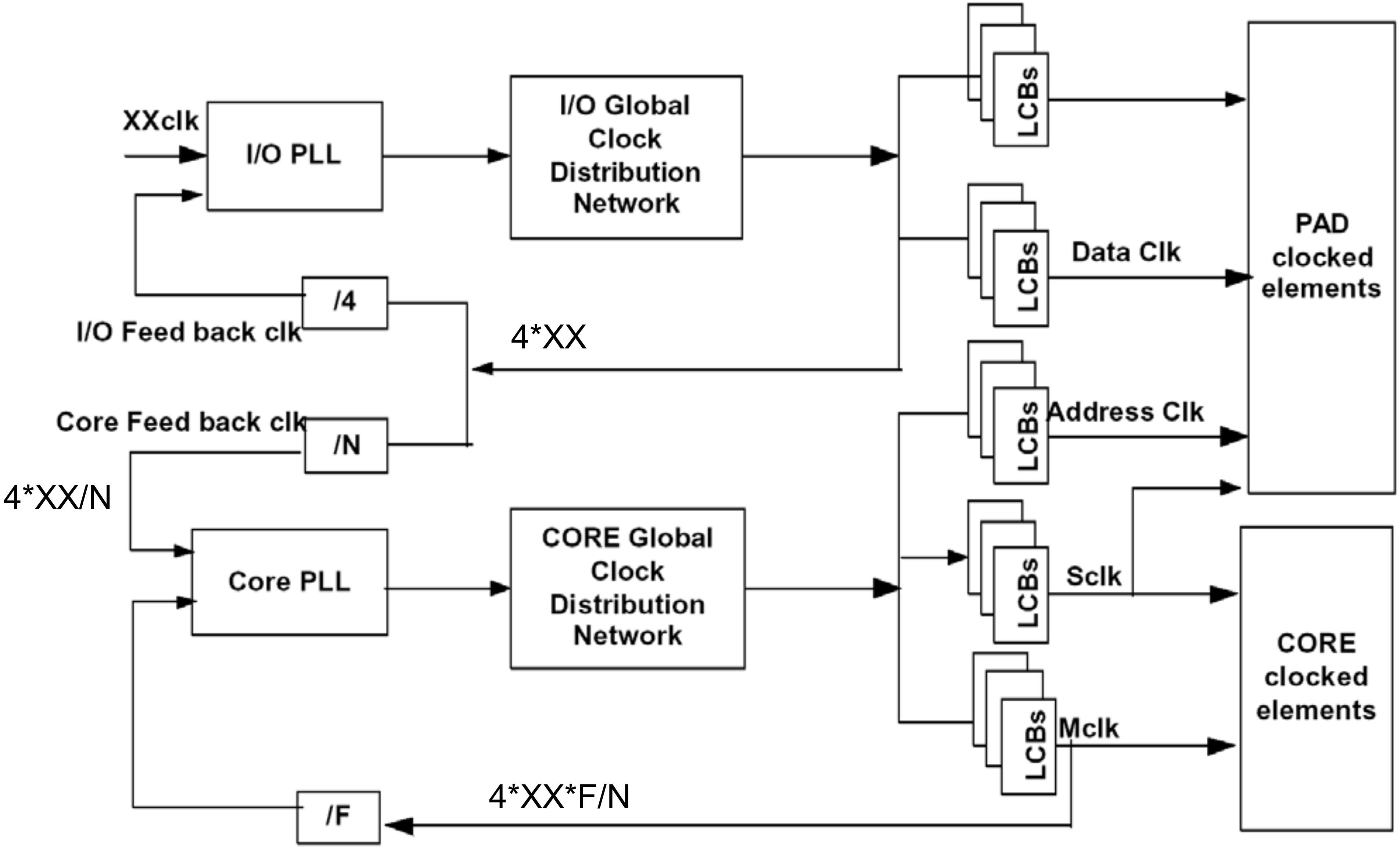


Chip-2-Chip PLL Synchronization

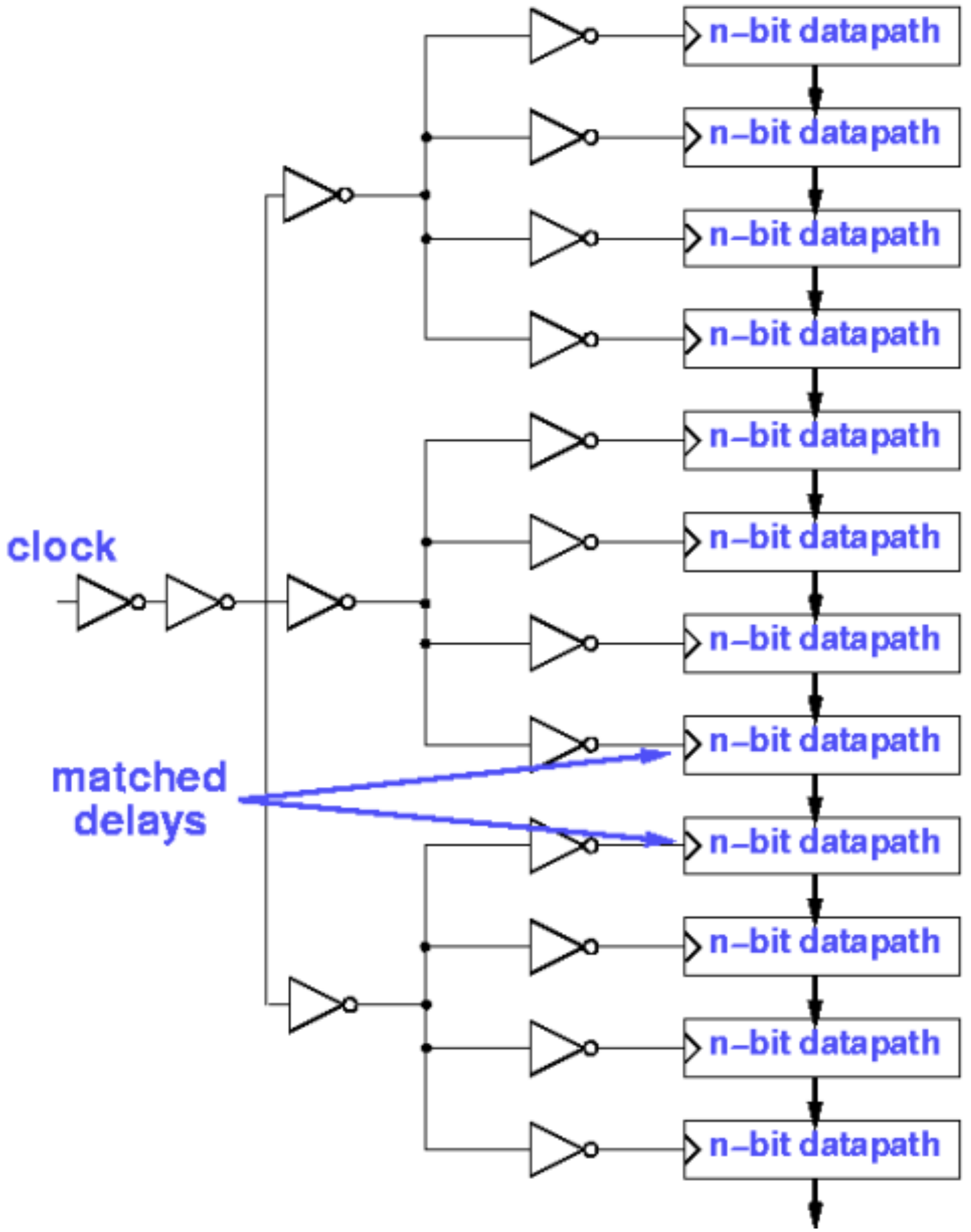


© J. Rabaey, 2002

High Performance Processor Clock Network



Clock Distribution tree for a Datapath



High peak currents to drive typical clock loads (¼ 1000 pF)

$$I_{peak} = C\frac{dV}{dt},$$

$$P_d = CV_{DD}^2 f$$

Matching Delays in Clock Distribution

- Balance delays of paths

- Match buffer and wire delays to minimize skew

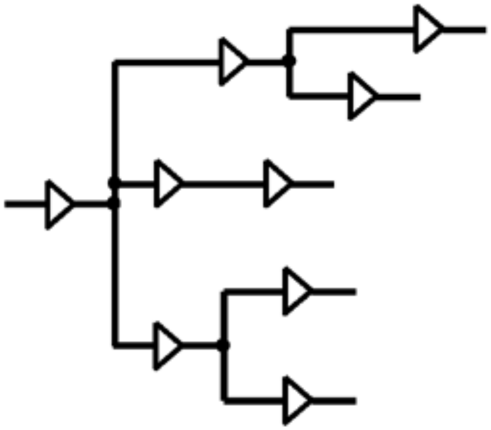
- Issues
  - Load of latch (driven by clock) is data-dependent (capacitance depends on source voltage)
  - Process variations
  - IR drops and temperature variations

- Need tools to support clock tree design

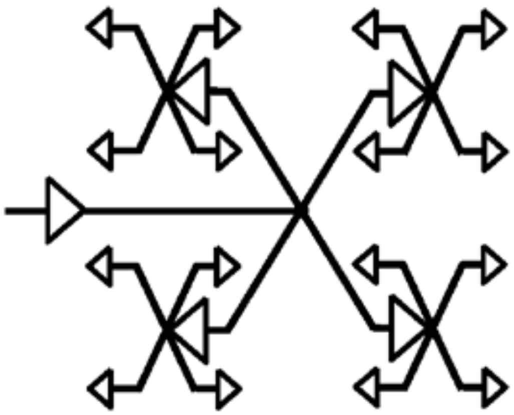
Backup

Variations of tree distribution networks

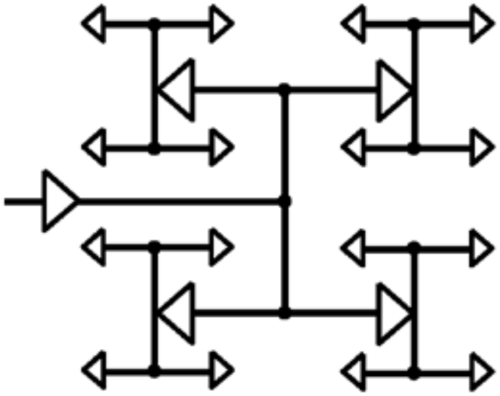
- *Target: Metallization and Gate topology uniformity*



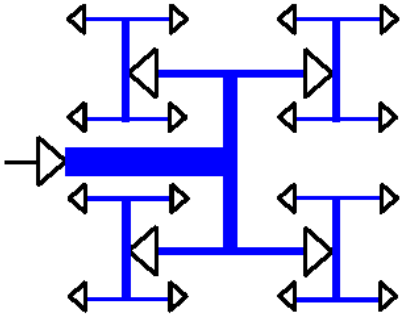
Tree



X-Tree

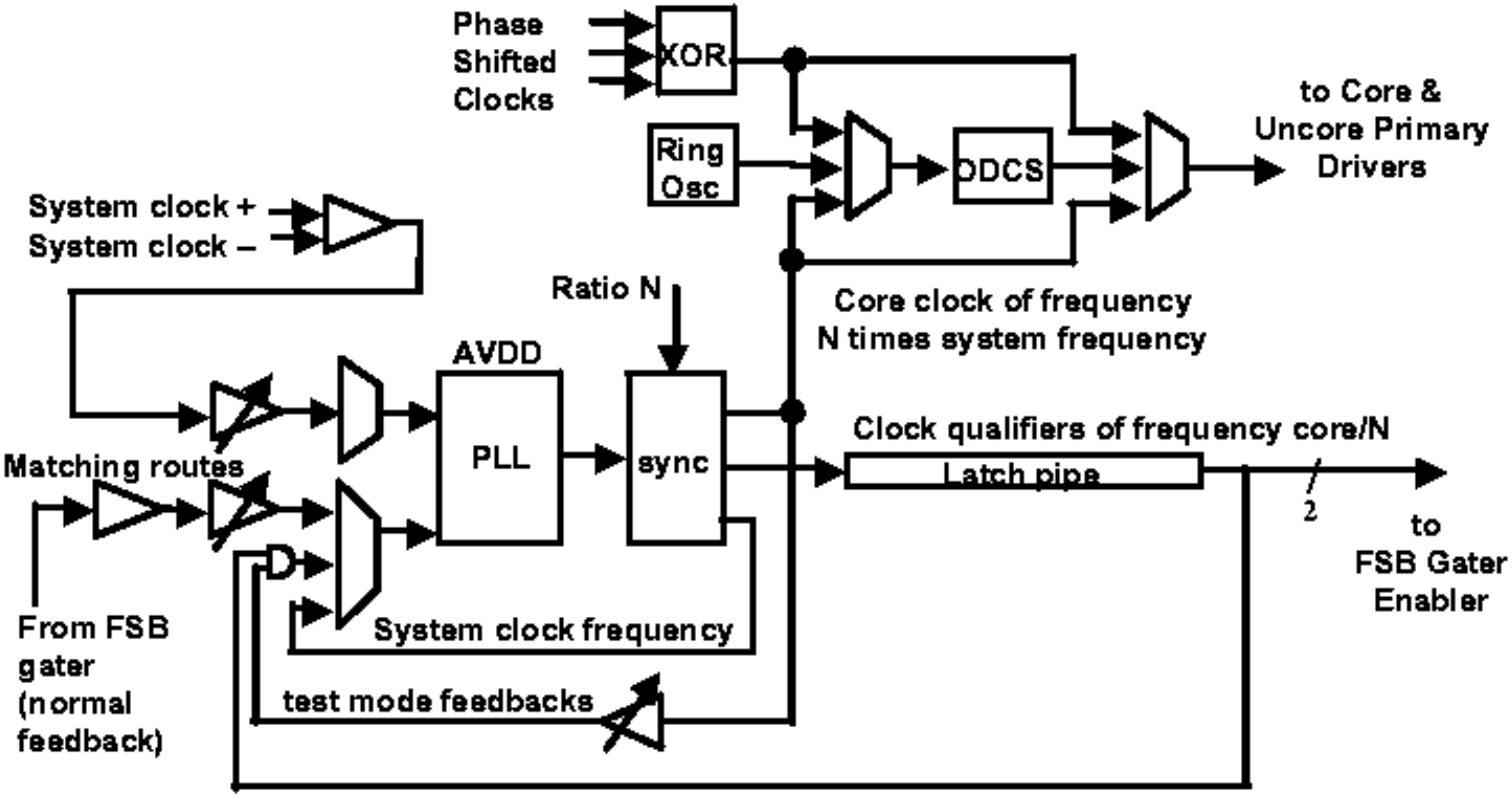


H-Tree

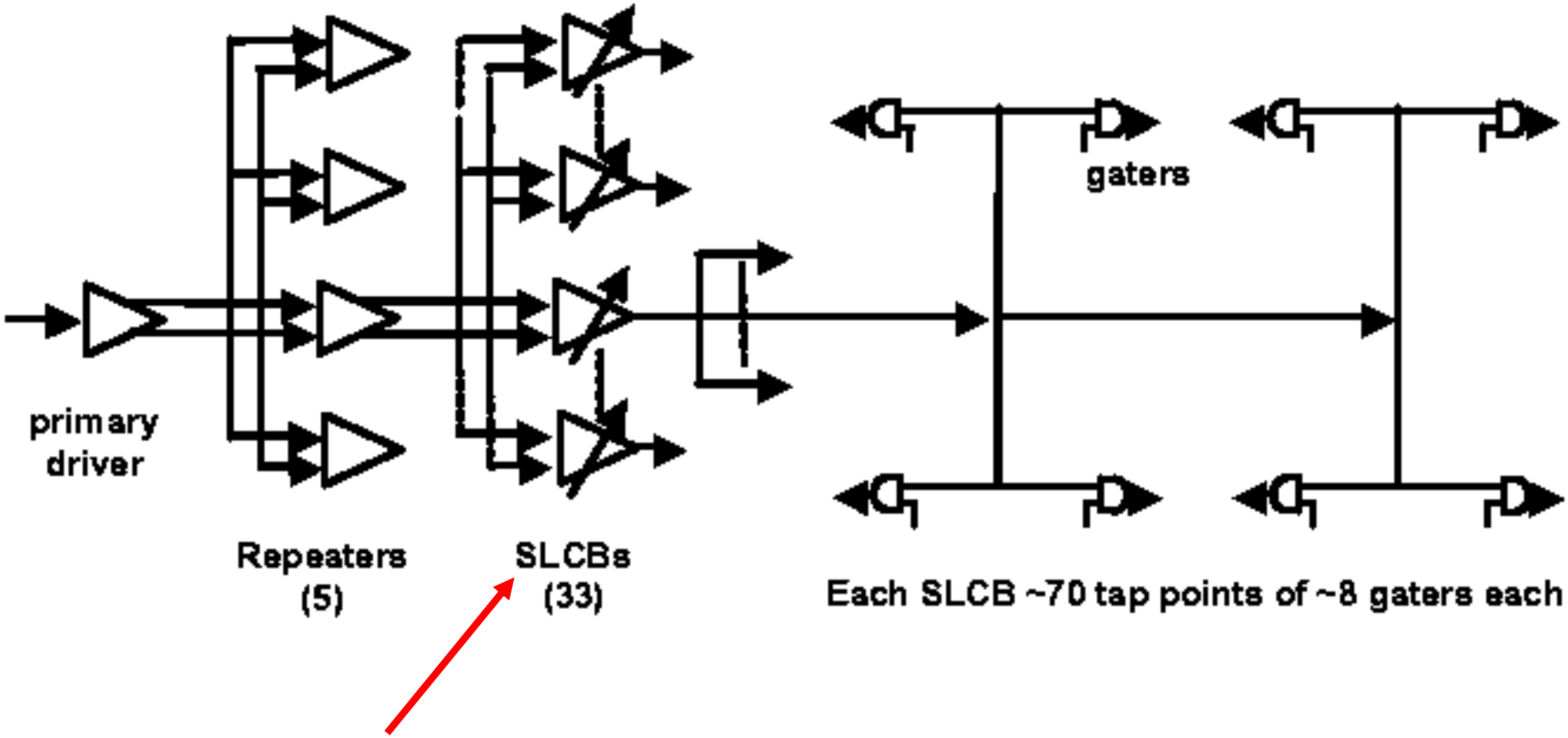


- Tapered H-Tree

Clock Generation

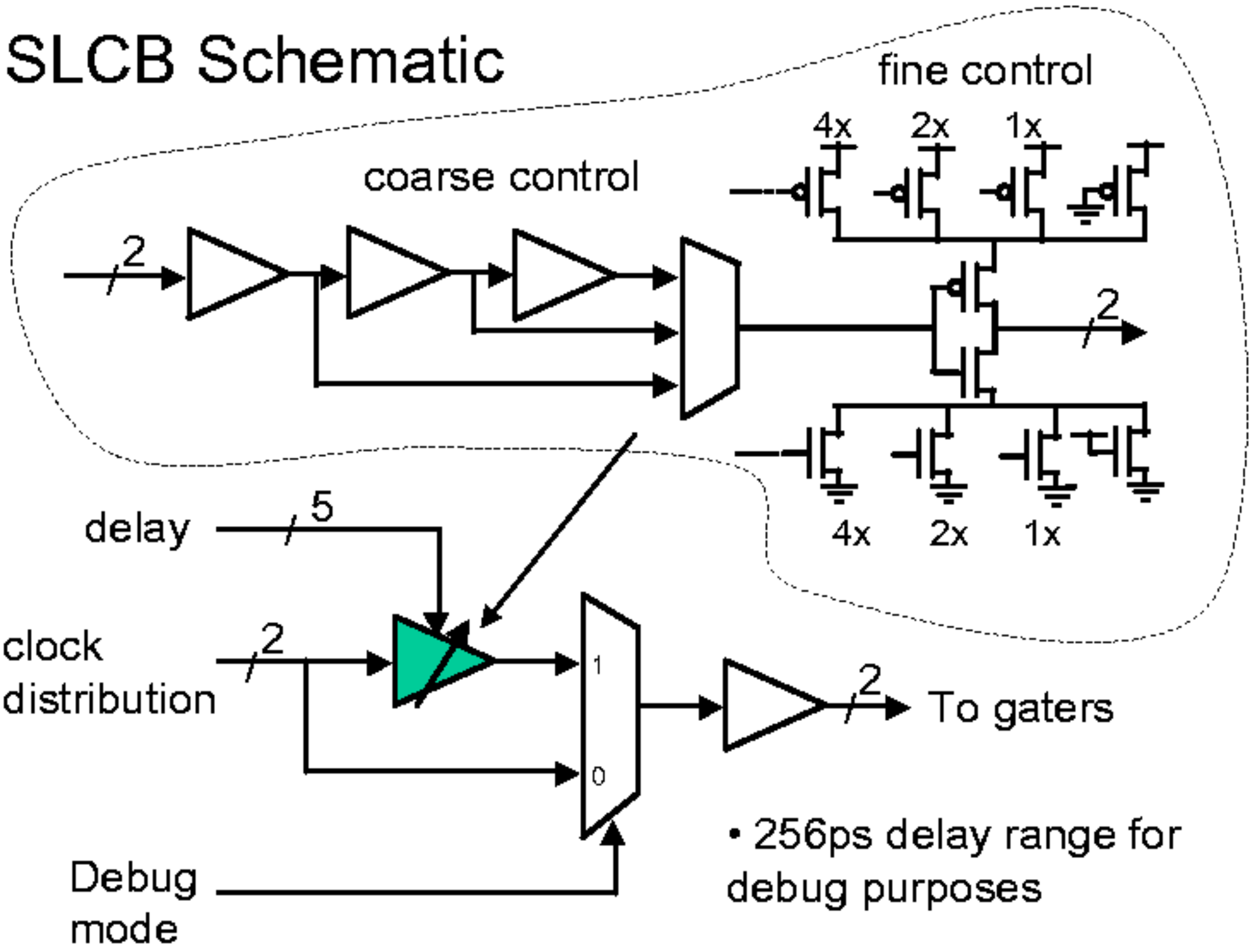


Core Clock Distribution



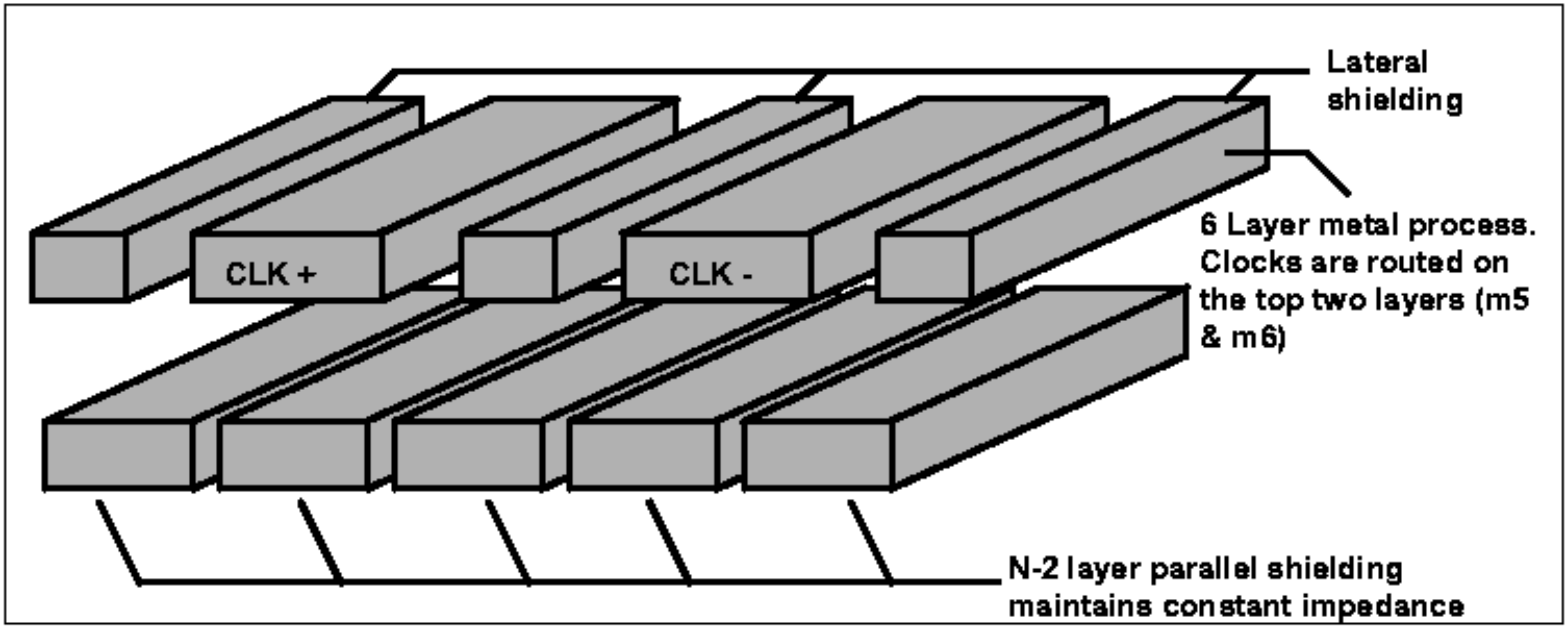
Adjustable delay buffer

SLCB (Second Level Clock Buffer)



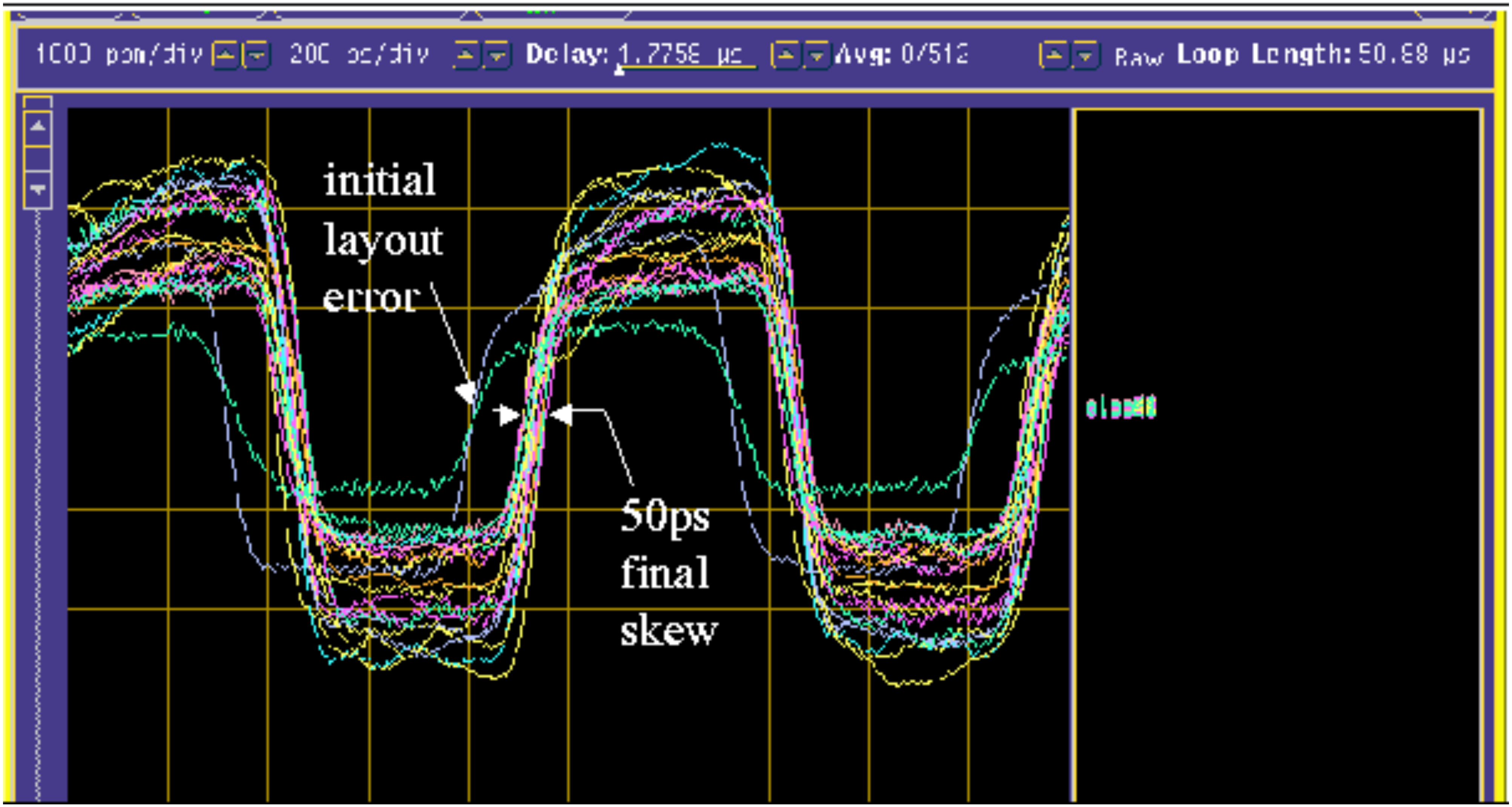
- 256ps delay range for debug purposes

First Level Route Geometry


Lateral shielding
CLK +
CLK -
6 Layer metal process. Clocks are routed on the top two layers (m5 & m6)
N-2 layer parallel shielding maintains constant impedance

Measured Skew

Laser probed SLCB skew (non-loading, valid DC levels)

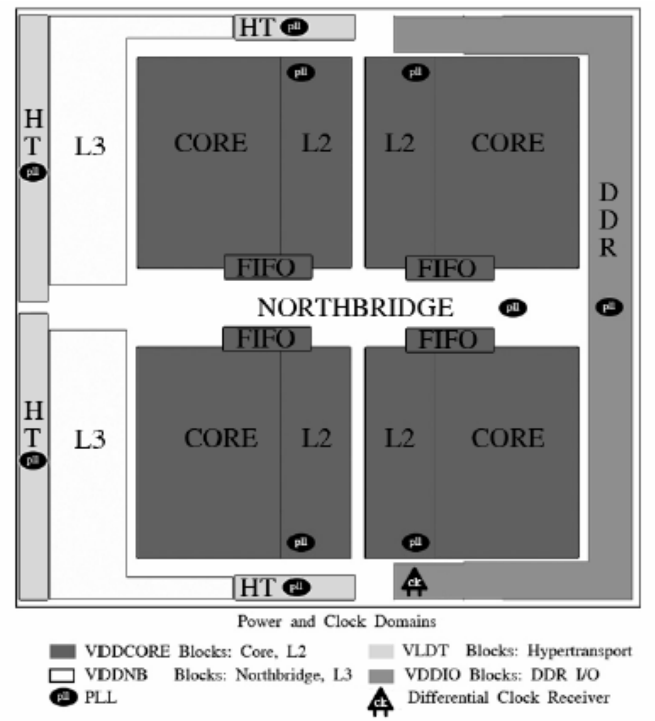


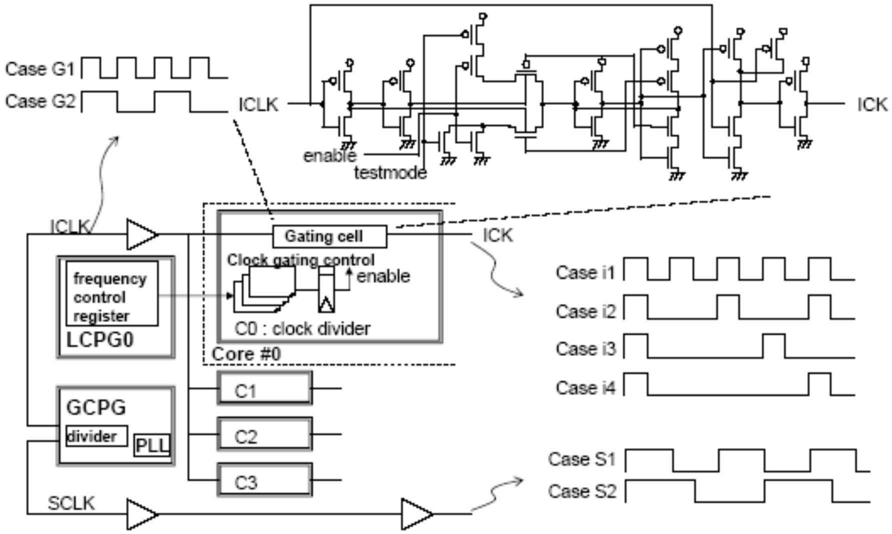
1000 ppm/div 200 ps/div Delay: 1.7758 µs Avg: 0/512 Raw Loop Length: 50.68 µs

initial layout error

50ps final skew

ISSCC 2007: Multi-Core Clocking Approach Asynchronous Communication and Independent Core Frequencies

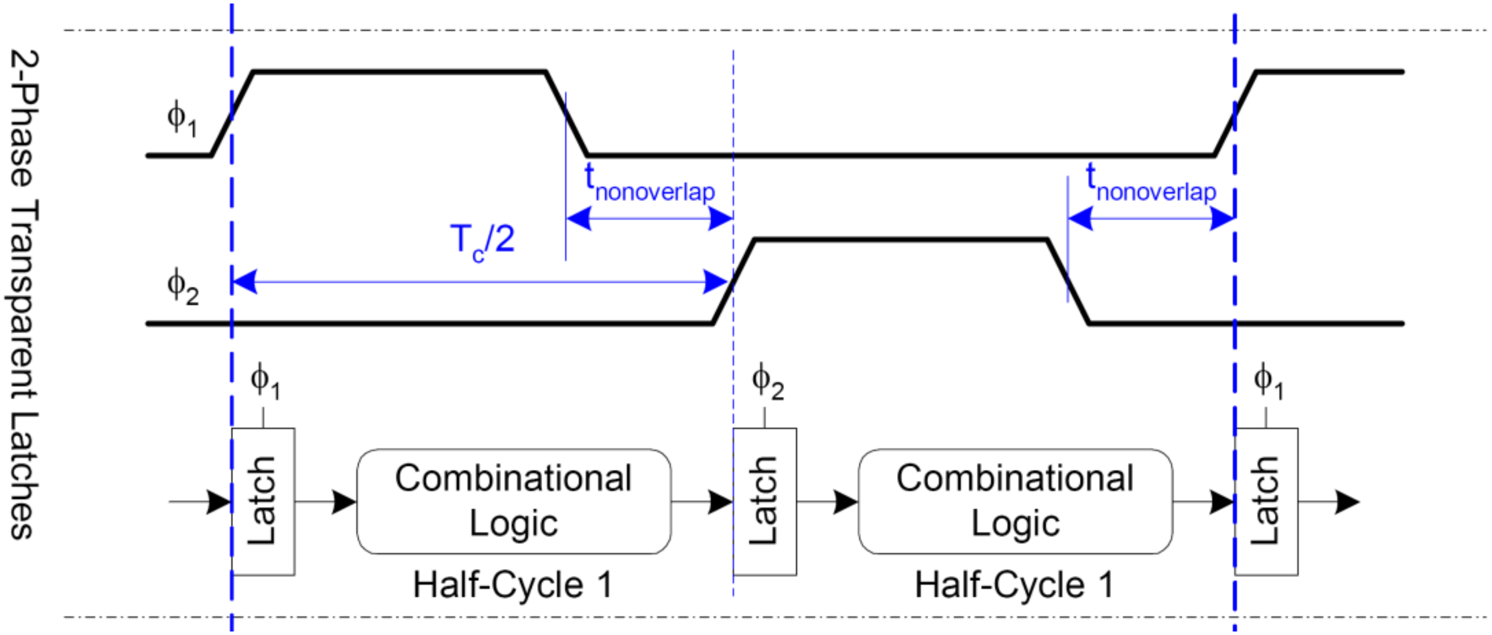


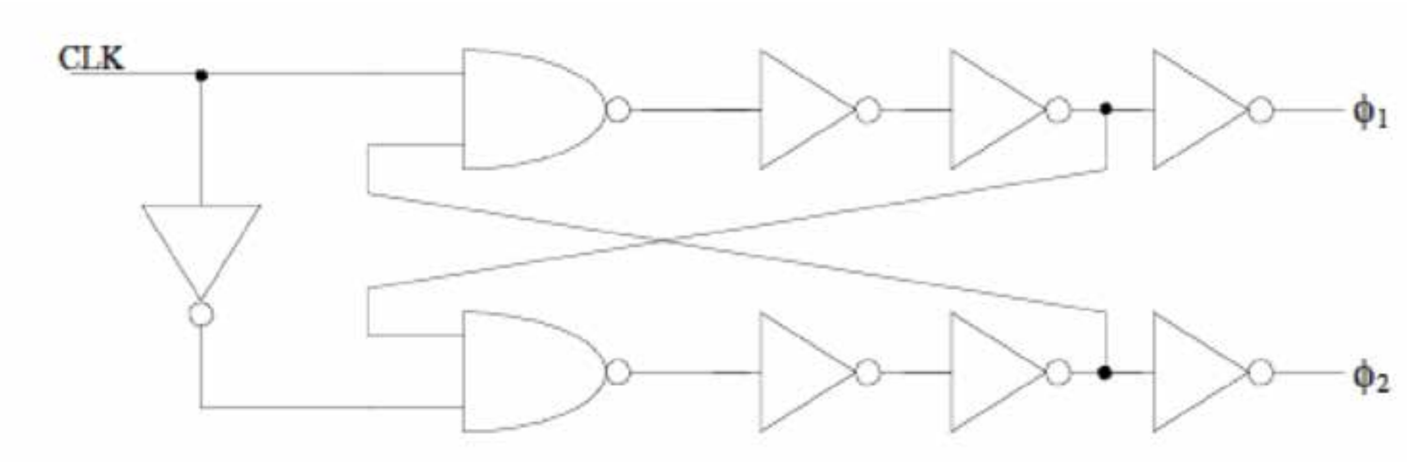


- •Source: An Integrated Quad-Core Opteron™ Processor
- •ISSCC 2007

- •Source: A 4320MIPS Four-Processor Core SMP/AMP with
- •Individually Managed Clock Frequency for Low
- •Power Consumption, ISSCC 2007

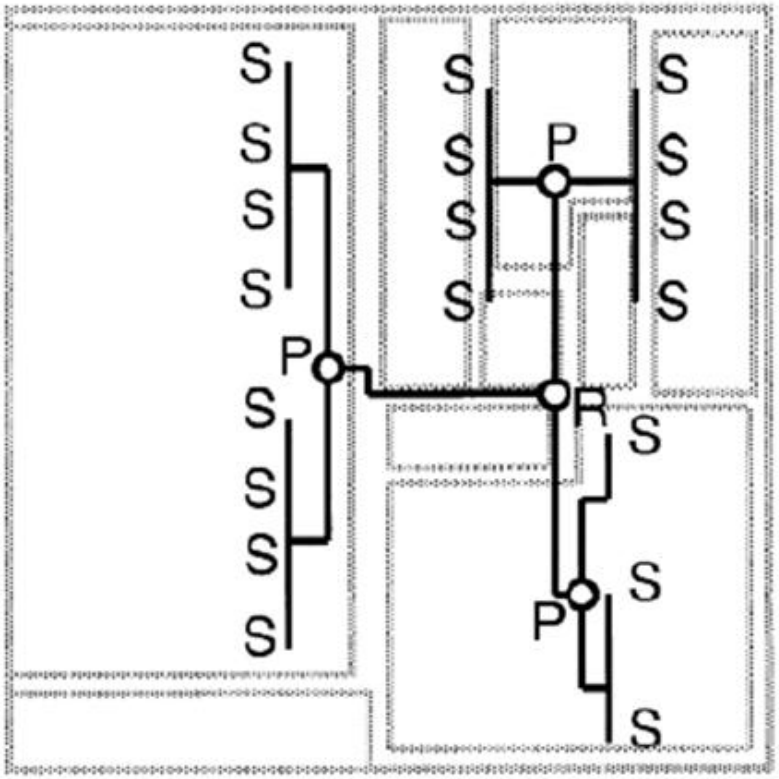
Non-overlapping clock generator





HP PA-RISC Clock Distribution

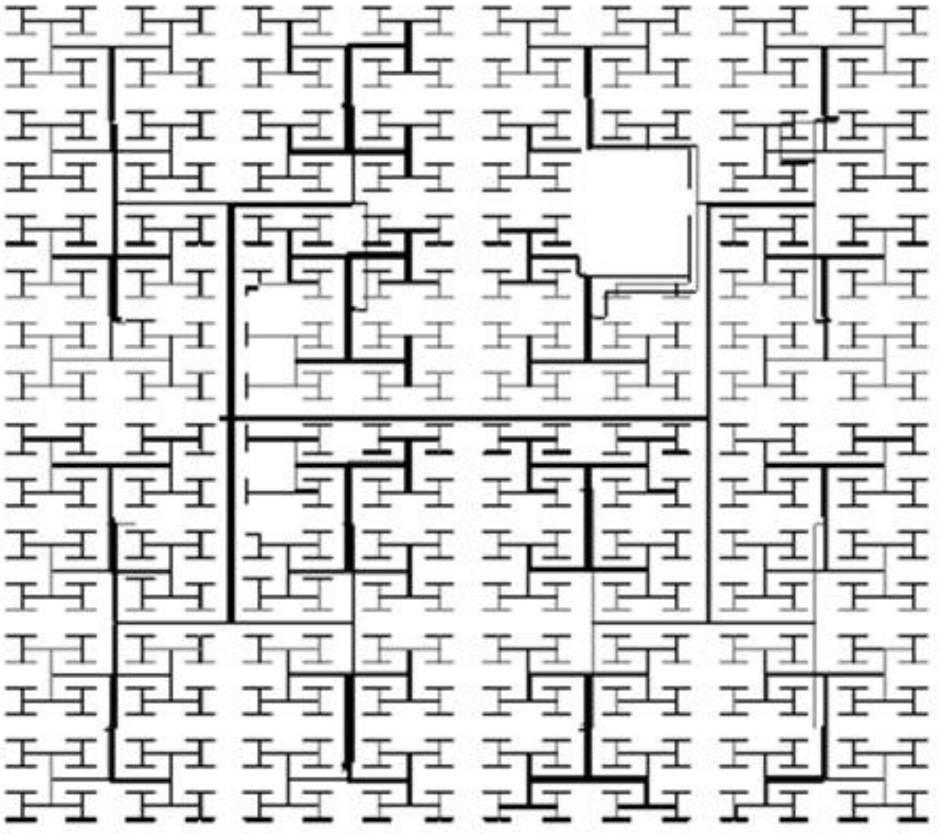
- Plan the clock usage points to minimize the distribution area
- Large caches do not require the clock to be distributed over them
- Example from the HP RISC design:
  - 1 Receiver Buffer (R)
  - 3 Primary Buffers (P)
  - 19 Secondary Buffers (S)
  - Balanced Tree Distribution
  - Die Size 21.3 x 22.0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



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

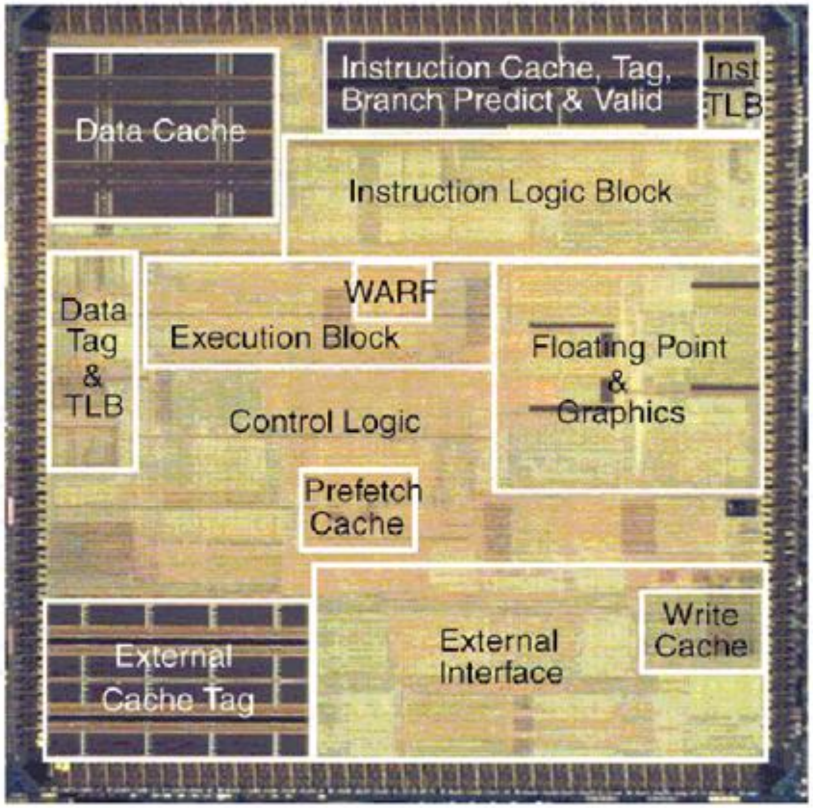
IBM PowerPC Clock Distribution

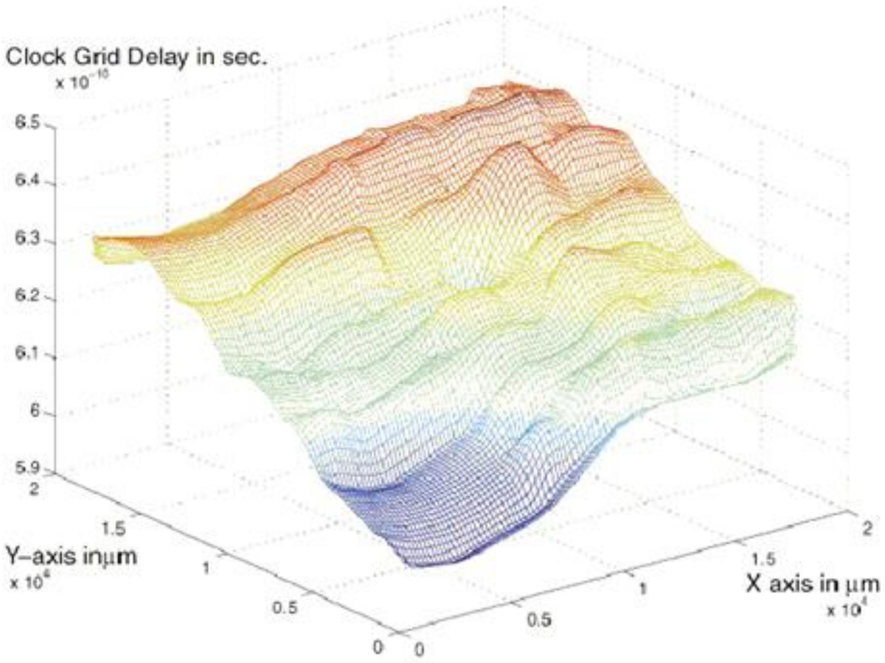
- 0.22µm technology
- 17mm x 17mm die size
- 19M transistors
- 6 level metal with copper interconnect technology
- Clock tree on top 2 metal levels
- 1 GHz clock frequency
- Almost symmetric H-tree
- Simulated clock skew under 15ps



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

Sun Microsystems UltraSparc III



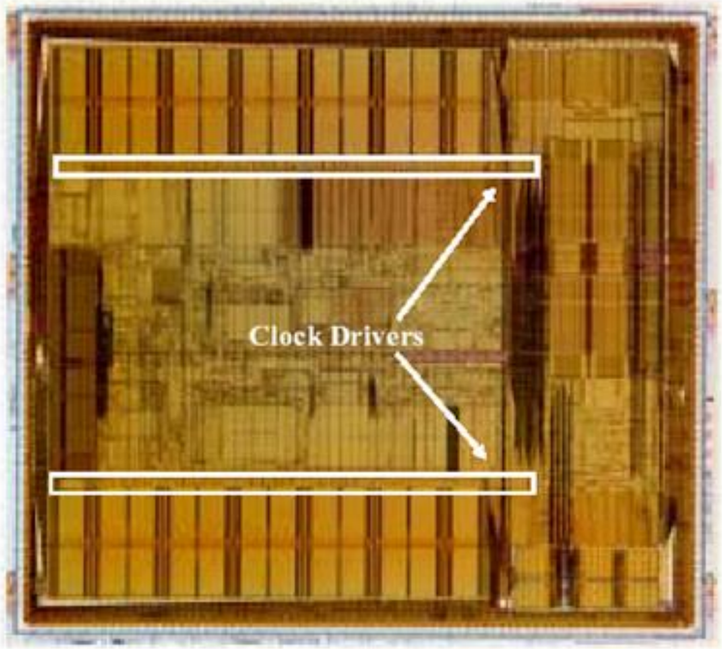


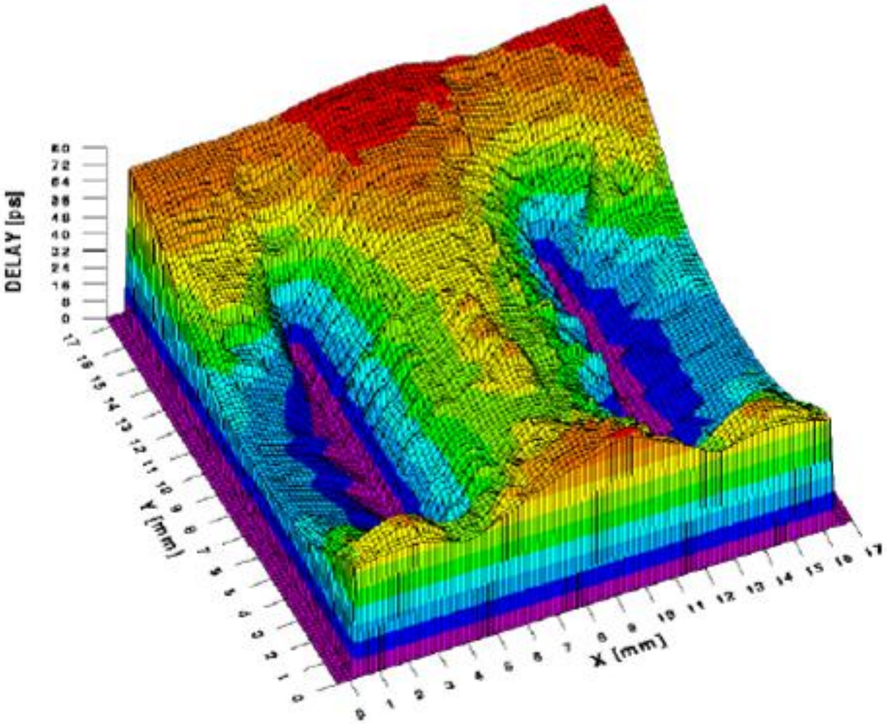
- ◆15.0 x 15.5 mm die size, 23M transistors
- ◆Overall clock skew <80ps

DEC Alpha 21164 Clock Distribution

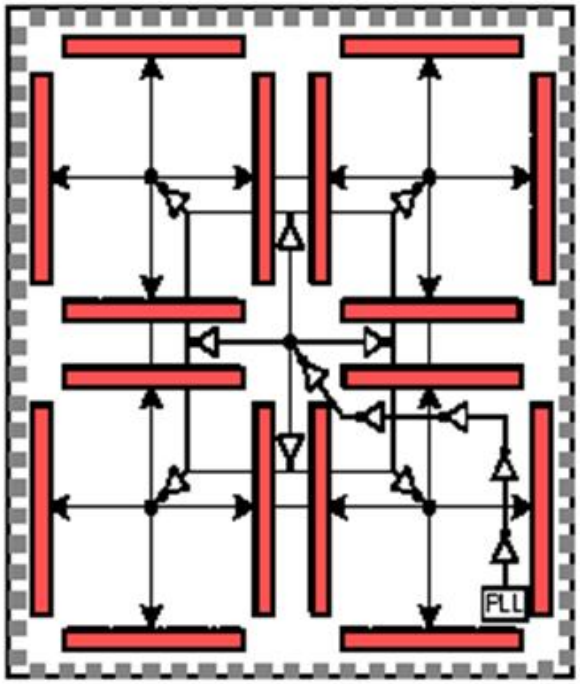
- Clock Frequency: 300MHz
- 9.3 Million transistors
- Total Clock Load: 3.75nF
- Power in Clock Distribution network: 20W (out of 50 total)

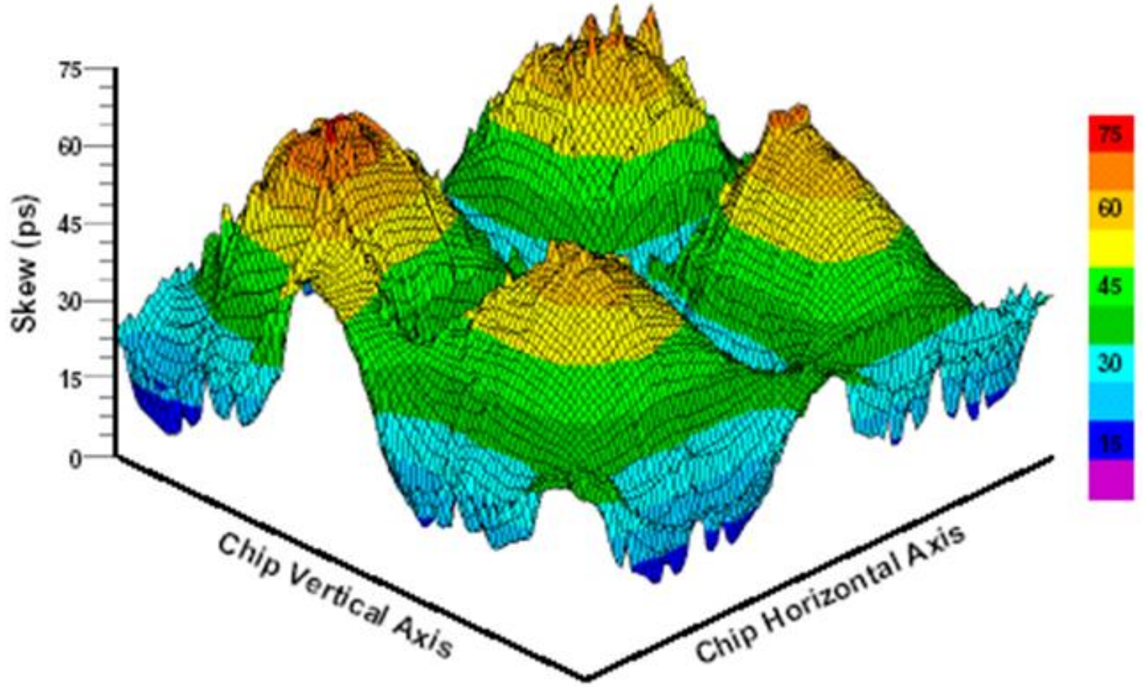
- Uses two level clock distribution
  - Single 6-stage driver at center of chip
  - Secondary buffers drive left and right side clock grid in M3 and M4
  - Total driver size: 58cm !!





DEC Alpha 21264 Clock Distribution



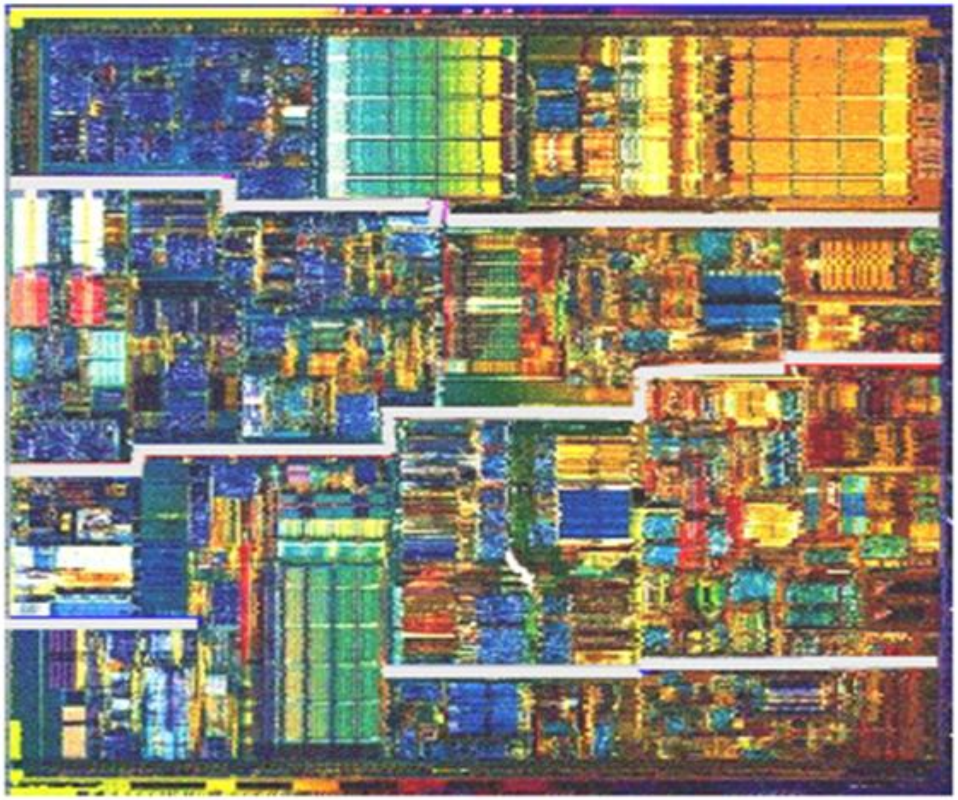


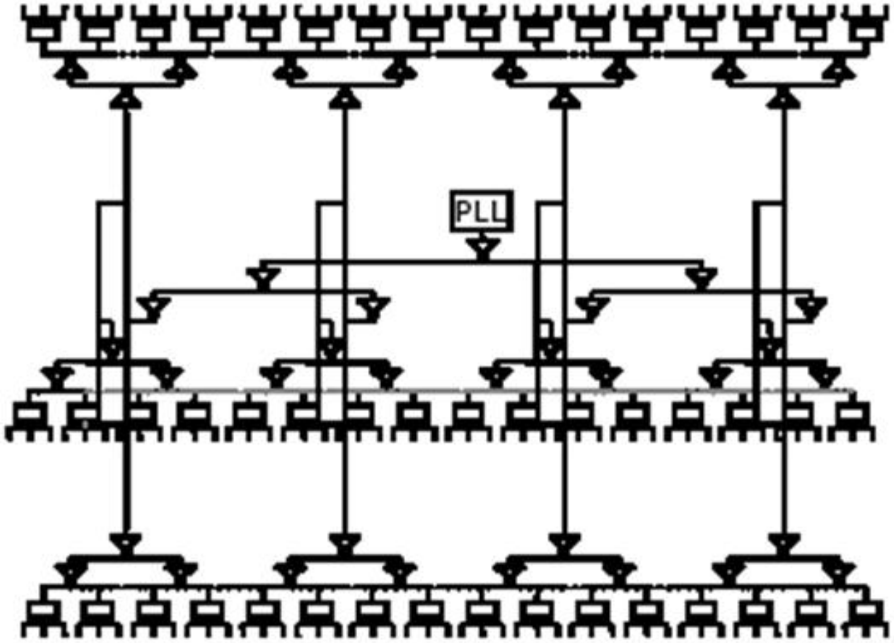
- 600MHz, 9.3M transistors, 0.35µm CMOS, 6 metals, 72W power dissipation at 2.0V
- Total clock load: 2.8nF

Gronowski, JSSC 1998

Pentium® 4 Processor Clock Network

- **2GHz triple-spine clock distribution**



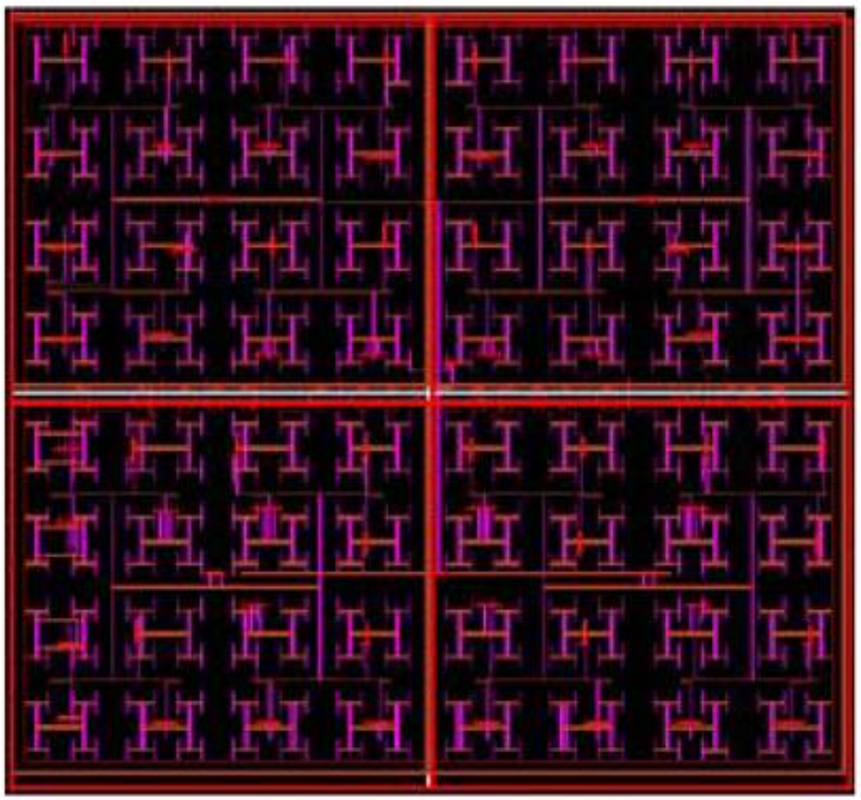


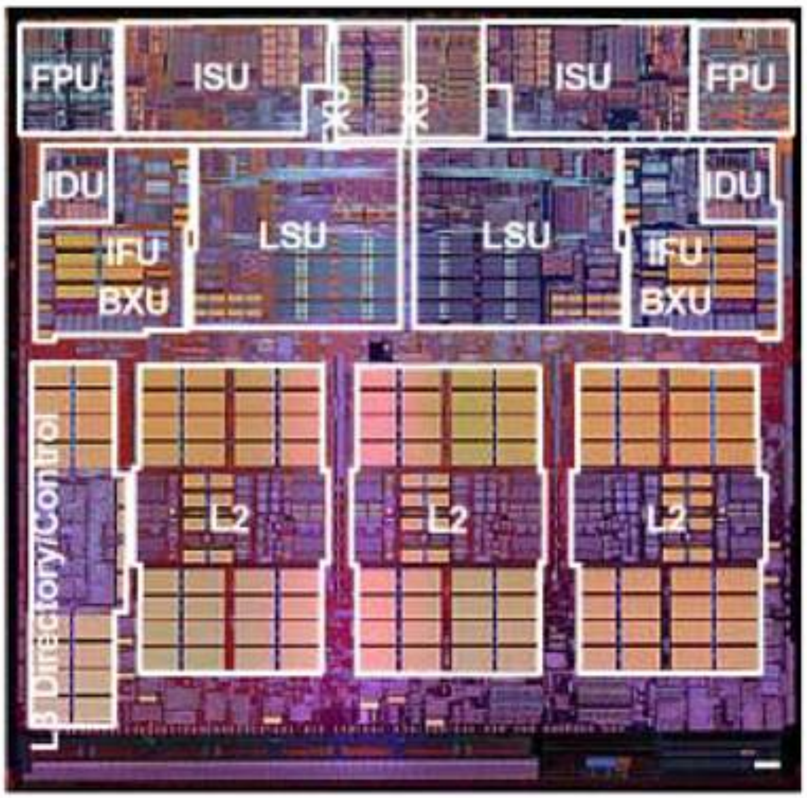
Kurd, JSSC-2001

IBM POWER-4 in .18u SOI

- H-Tree and Grid Distribution
- Clock skew: <25ps
- ~70% power in clock and

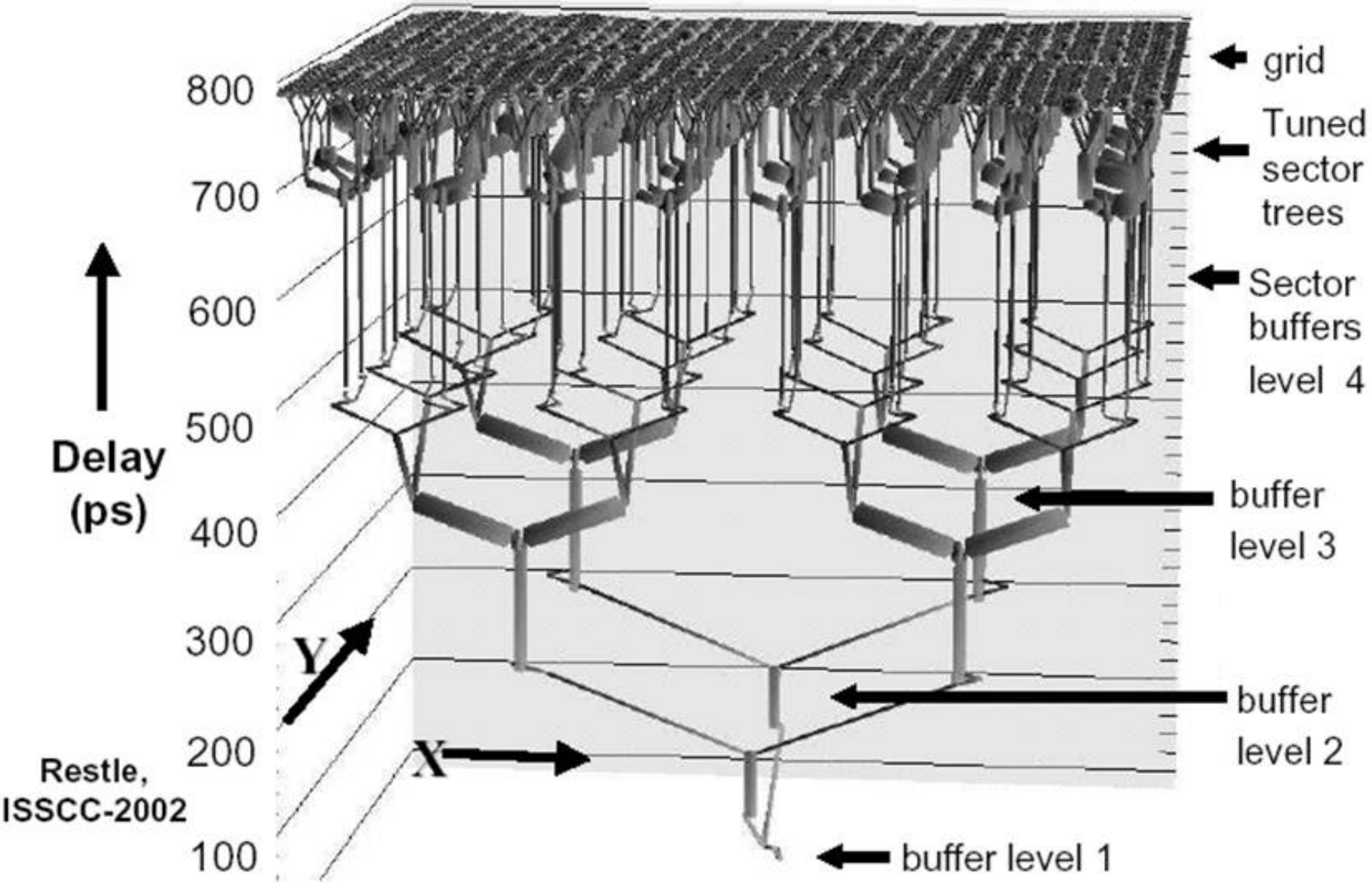
- Dual core, shared L2
- 174M transistors
- 115W at 1.1GHz, 1.5V





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

IBM POWER-4 3D Skew Visualization



Restle, ISSCC-2002

Clock Skew across CELL Processor Global Clock Distribution

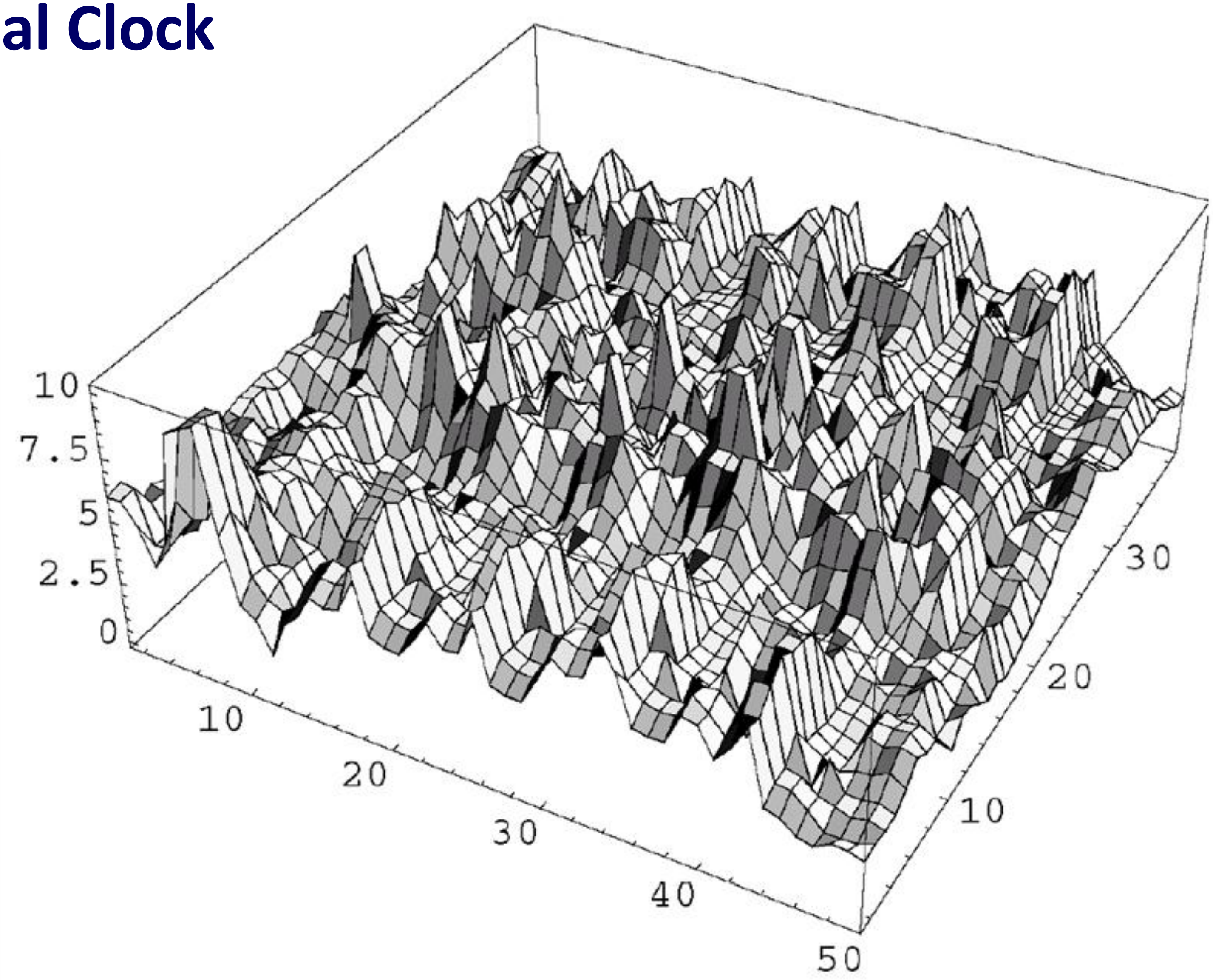


Figure 10.2.2: Clock-skew map.