10. Interconnects in CMOS Technology

Jacob Abraham

Department of Electrical and Computer Engineering
The University of Texas at Austin

VLSI Design
Fall 2020

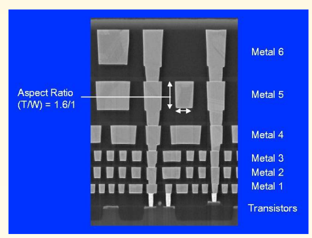
September 29, 2020

Introduction to Wires on a Chip

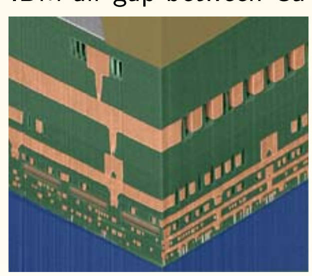
Most of chip is wires (interconnect)

- Most of the chip is covered by wires, many layers of wires
- Transistors: little things under wires
- Wires as important as transistors
  - Affect
    - Speed
    - Power
    - Noise
- Alternating layers usually run orthogonally

Intel Damascene copper

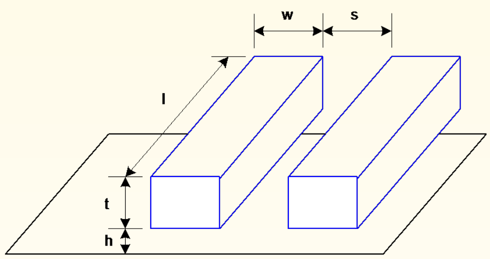


IBM air gap between Cu



Wire Geometry

- Pitch = w + s
- Aspect Ratio, AR = t/w
  - Old processes had AR << 1
  - Modern processes have AR ≈ 2 to pack in many skinny wires

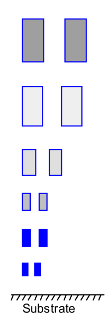


Layer Stack

- Number of metal layers has been increasing
  - AMI 0.6 mm process has 3 metal layers
  - Modern processes use 6-10+ metal layers

- Example: Intel 180 nm process
- M1: thin, narrow ($< 3\lambda$)
  - High density cells
- M2-M4: thicker
  - For longer wires
- M5-M6: thickest
  - For V_{DD}, GND, CLK

Layer	T (nm)	W (nm)	S (nm)	AR
6	1720	860	860	2.0
	1000			
5	1600	800	800	2.0
	1000			
4	1080	540	540	2.0
	700			
3	700	320	320	2.2
	700			
2	700	320	320	2.2
	700			
1	480	250	250	1.9
	800			

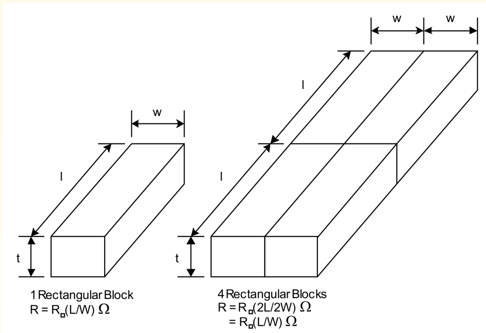


Wire Resistance

$\rho = resistivity\ (\Omega * m)$

$$R = \frac{\rho}{t}\frac{l}{w} = R_{\square}\frac{l}{w}$$

- $R_{\square} = sheet\ resistance\ (\Omega/\square)$
  - $\square$ is a dimensionless unit
- Count number of squares
  - $R = R_{\square} * (\#\ of\ squares)$



Choice of Metals

- Until the 180 nm generation, most wires were aluminum
- Modern processes often use copper
  - Cu atoms diffuse into silicon and damage FETs
  - Must be surrounded by a diffusion barrier

Metal	**Bulk Resistivity** $(\mu\Omega * cm)$
Silver (Ag)	1.6
Copper (Cu)	1.7
Gold (Au)	2.2
Aluminum (Al)	2.8
Tungsten (W)	5.3
Molybdenum (Mo)	5.3

Sheet Resistance

Typical sheet resistances in 180 nm process

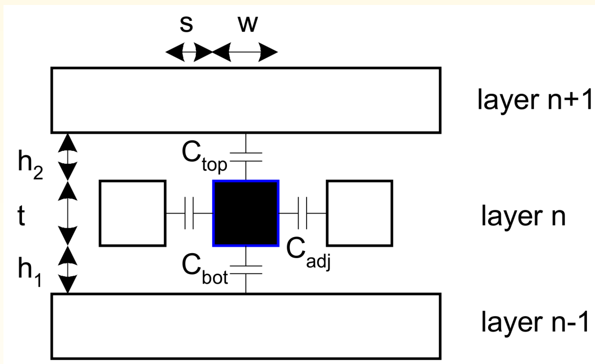
Layer	Sheet Resistance ($\Omega/\square$)
Diffusion (silicided)	3–10
Diffusion (no silicide)	50–200
Polysilicon (silicided)	3–10
Polysilicon (no silicide)	50–400
Metal1	0.08
Metal2	0.05
Metal3	0.05
Metal4	0.03
Metal5	0.02
Metal6	0.02

Contact Resistance

- Contacts and vias also have 2-20 Ω resistance
- Use many contacts for lower R
  - Many small contacts for current crowding around periphery
- Multiple contacts also help improve the yield (failure or high resistance of a contact will have only a small effect on the overall resistivity)

Wire Capacitance

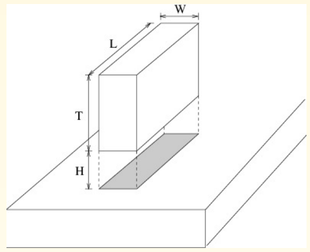
- Wire has capacitance per unit length
  - To neighbors
  - To layers above and below
- $C_{total} = C_{top} + C_{bot} + 2C_{adj}$



Capacitance Trends

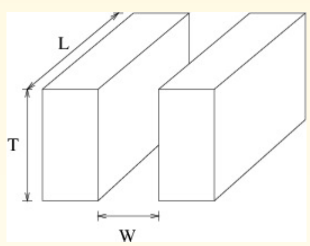
- Parallel plate equation: $C = \epsilon A/d$
  - Wires are not parallel plates, but obey trends
  - Increasing area (W, t) increases capacitance
  - Increasing distance (s, h) decreases capacitance
- Dielectric Constant
  - $\epsilon = k\epsilon_0$
- $\epsilon_0 = 8.85 \times 10^{14}$ F/cm
- $k = 3.9$ for SiO_2
- Processes are starting to use low-k dielectrics
  - $k \approx 3$ (or less) as dielectrics use air pockets

C_{top}/C_{bot} Trends



- $W >> H \Rightarrow$ Parallel Plate Model
  - $C = k \cdot \epsilon_0 \cdot \frac{W \cdot L}{H}$
- $W \leq H \Rightarrow$ Fringing Model
  - $C \; \alpha \; log(W)$
- For Deep Sub-Micron (DSM) (or nanoscale) processes, fringing model applies

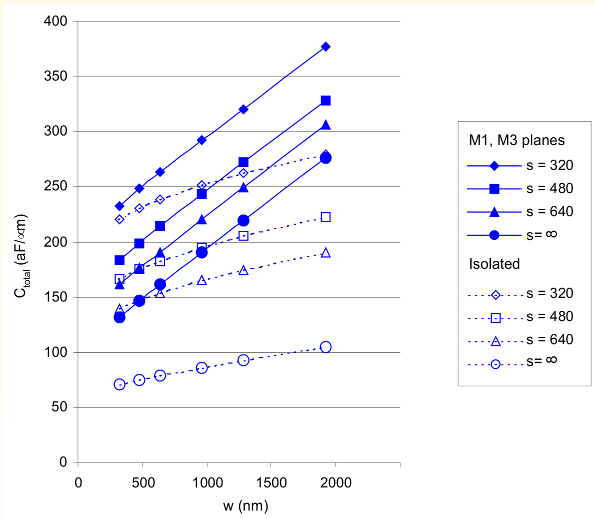
C_{adj} Trends



- $T >> W \Rightarrow$ Parallel Plate Model
  - $C = k \cdot \epsilon_0 \cdot \frac{T \cdot L}{W}$
- $T \leq W \Rightarrow$ Fringing Model
  - $C \; \alpha \; log(T)$
- For DSM processes, parallel plate model applies

M2 Capacitance Data

- Typical wires have $\approx 0.2\ fF/\mu m$
  - Compare to $2\ fF/\mu m$ for gate capacitance

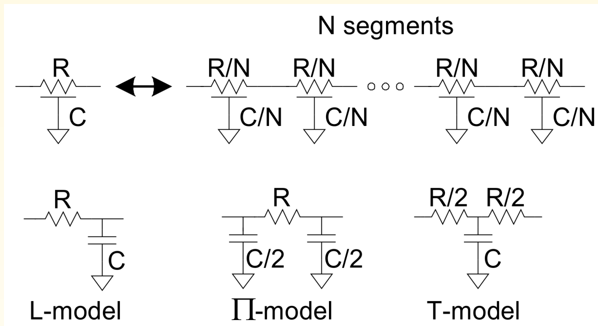


Diffusion and Polysilicon

- Diffusion capacitance is very high (about $2\ fF/\mu m$)
  - Comparable to gate capacitance
  - Diffusion also has high resistance
  - Avoid using diffusion *runners* for wires!
- Polysilicon has lower C but high R
  - Use for transistor gates
  - Occasionally for very short wires between gates

Lumped Element Models

- Wires are a distributed system
  - Approximate with lumped element models



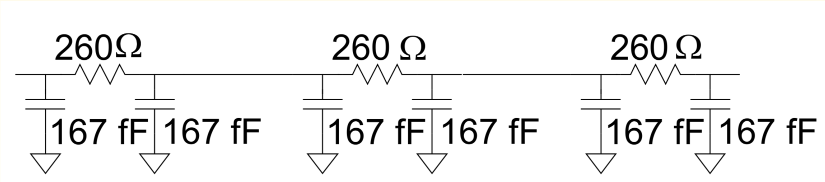
- 3-segment π-model accurate to 3% in simulation
- L-model needs 100 segments for same accuracy!
- Use single segment π-model for Elmore delay

When to use Lumped versus Distributed Models

- First find the total R and total C for the wire.
  - If RC $\gg t_r$ (or t_f) of driver then use distributed (Π or T) model
  - If RC $\leq t_r$ (or t_f) of driver then use lumped (L) model
- It is safe to use distributed model always, but this results in more circuit elements and larger simulation times.
- To find number of distributed elements to use
  - Increase the number of elements, and stop when the error between k and $k+1$ elements is acceptably small.
- Distributed RC delay is about half that of lumped RC
- This can be validated by using the Elmore model for the distributed wire (see previous slide)
- Rule of Thumb: for a distributed wire, propagation delay can be estimated as $\sim$ RC$/2$.

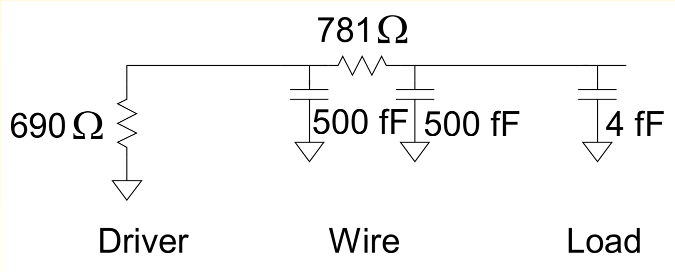
Example

- Metal2 wire in 180 nm process
  - 5 mm long
  - 0.32 μm wide
- Construct a 3-segment π-model
  - $R_{\square} = 0.05\ \Omega/\square \implies R = 781\ \Omega$
  - $C_{permicron} = 0.2 fF/\mu\text{m} \implies C = 1\ pF$



Wire RC Delay

- Estimate the delay of a 10x inverter driving a 2x inverter at the end of the 5mm wire from the previous example
  - R = 2.5 k$\Omega * \mu$m for gates
  - Unit inverter: 0.36 μm nMOS, 0.72 μm pMOS



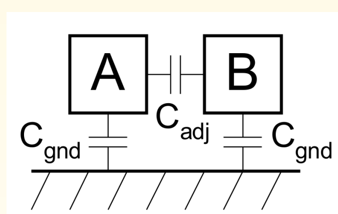
- t_{pd} = 1.1 ns

Crosstalk

- A capacitor does not like to change its voltage instantaneously
- A wire has high capacitance to its neighbor
  - When the neighbor switches from 1→0 or 0→1, the wire tends to switch too
  - Called capacitive **coupling** or **crosstalk**
- Crosstalk effects
  - **Noise** on nonswitching wires
  - Increased **delay** on switching wires

Crosstalk Delay

- Assume layers above and below on average are quiet
  - Second terminal of capacitor can be ignored
  - Model as $C_{gnd} = C_{top} + C_{bot}$
- Effective C_{adj} depends on behavior of neighbors
  - **Miller Effect**

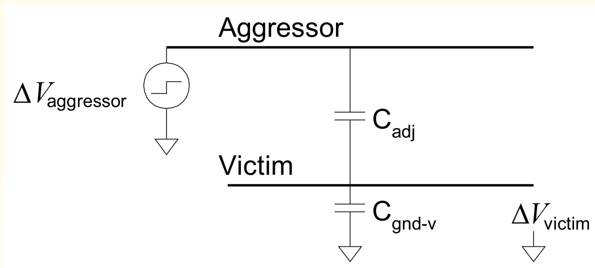


B	ΔV	$C_{eff(A)}$	MCF
Constant	V_{DD}	$C_{gnd} + C_{adj}$	1
Switching with A	0	C_{gnd}	0
Switching opposite A	$2V_{DD}$	$C_{gnd} + 2C_{adj}$	2

Crosstalk Noise

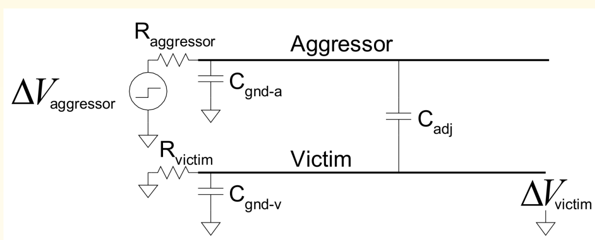
- Crosstalk causes noise on nonswitching wires
- If victim is floating:
  - model as capacitive voltage divider

$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd-v} + C_{adj}} \Delta V_{aggressor}$$



Driven Victims

- Usually victim is driven by a gate that fights noise
  - Noise depends on relative resistances
  - Victim driver is in linear region, aggressor in saturation
  - If sizes are same, $R_{aggressor} = 2 - 4 \times R_{victim}$

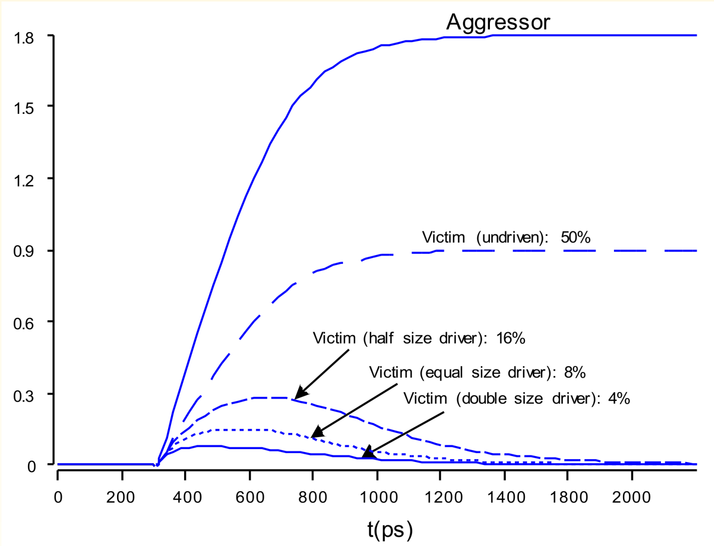


$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd-v} + C_{adj}} \frac{1}{1+k} \Delta V_{aggressor}$$

$$k = \frac{\tau_{aggressor}}{\tau_{victim}} = \frac{R_{aggressor}(C_{gnd-a} + C_{adj})}{R_{victim}(C_{gnd-v} + C_{adj})}$$

Coupling Waveforms

Simulated Coupling for $C_{adj} = C_{victim}$



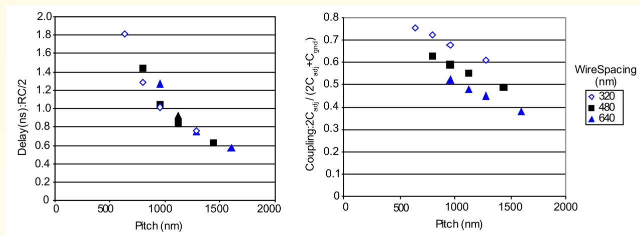
Noise Implications

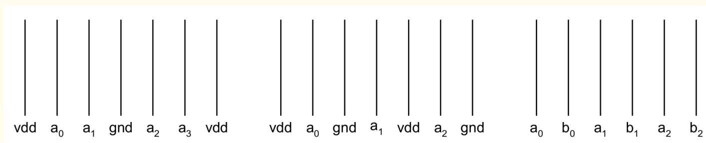
- So what if we have noise?
- If the noise is less than the noise margin, nothing happens
- Static CMOS logic will eventually settle to correct output even if disturbed by large noise spikes
  - But glitches cause extra delay
  - Also cause extra power from false transitions
- Dynamic logic never recovers from glitches
- Memories and other sensitive circuits also can produce the wrong answer

Wire Engineering

Goal: achieve delay, area, power goals with acceptable noise

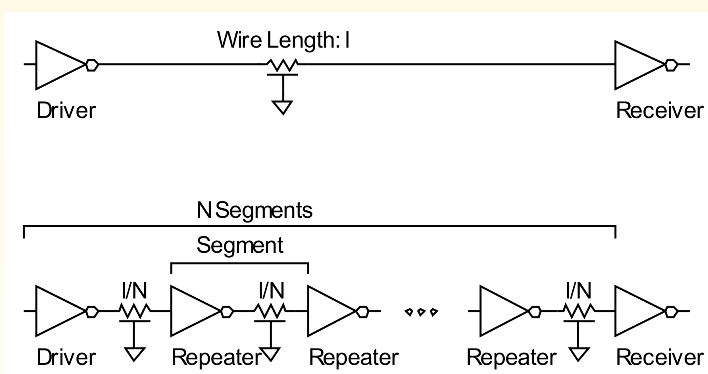
- Degrees of freedom
  - Width
  - Spacing
  - Layer
  - Shielding





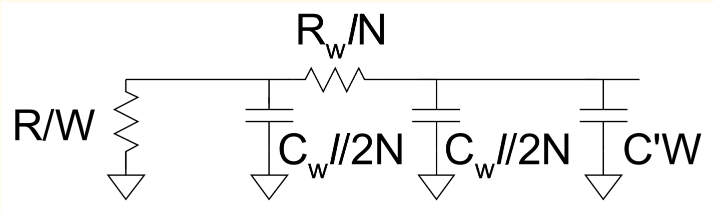
Repeaters

- R and C are proportional to l
- RC delay is proportional to l^2
  - Unacceptably great for long wires
- Break long wires into N shorter segments
  - Drive each one with an inverter or buffer



Repeater Design

- How many repeaters should we use?
- How large should each one be?
- Equivalent Circuit
  - Wire length l
    - Wire Capacitance $C_w * l$, Resistance $R_w * l$
  - Inverter width W (nMOS = W, pMOS = 2W)
    - Gate Capacitance C'*W, Resistance R/W



Repeater Results

- Write equation for Elmore Delay
  - Differentiate with respect to W and N
  - Set equal to 0, solve

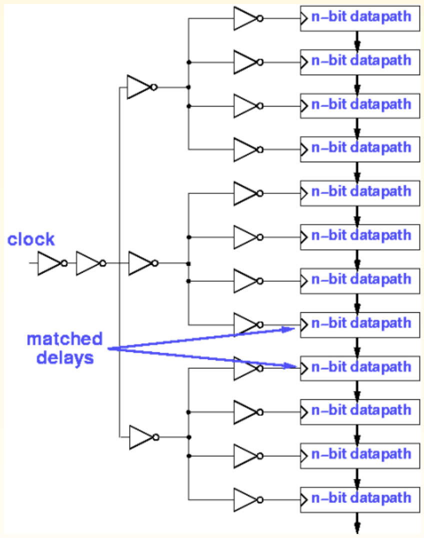
$$\frac{l}{N} = \sqrt{\frac{2RC'}{R_w C_w}}$$

$$\frac{t_{pd}}{l} = \left(2 + \sqrt{2}\right) \sqrt{RC' R_w C_w}$$

$\sim$ 60–80 ps/mm in 0.18μ process

$$W = \sqrt{\frac{RC_w}{R_w C'}}$$

Clock Distribution

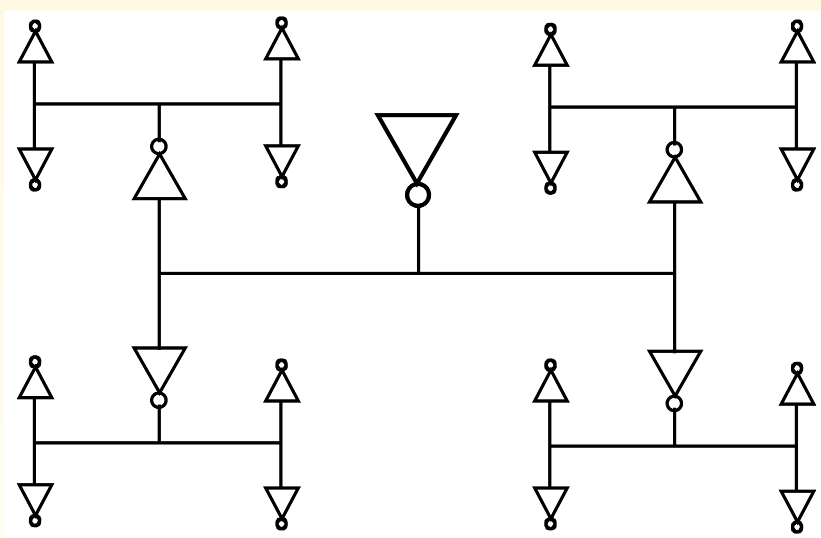


High peak currents to drive typical clock loads (≈ 1000 pF)

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

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

H-Trees

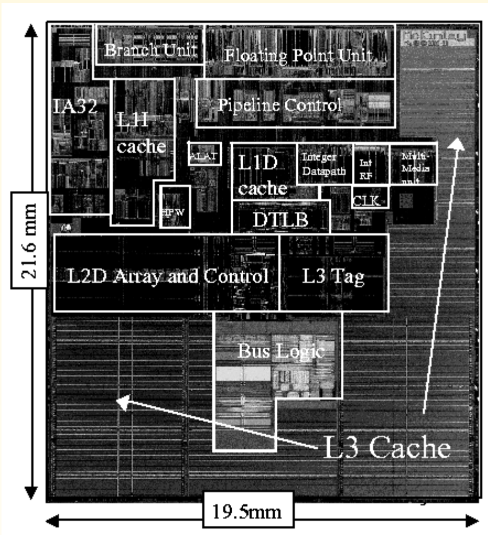


Matching Delays in Clock Distribution

- 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
- Tools to support clock tree design

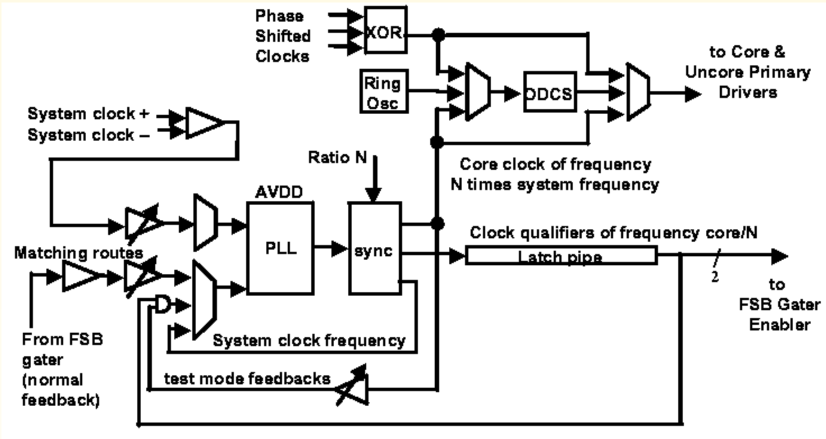
Clocking in the Itanium Processor

- 0.18μ technology
- 1GHz core clock
- 200 MHz system clk
- Core clocking
  - 260 mm^2
  - 1 primary driver
  - 5 repeaters
  - 157,000 clocked latches

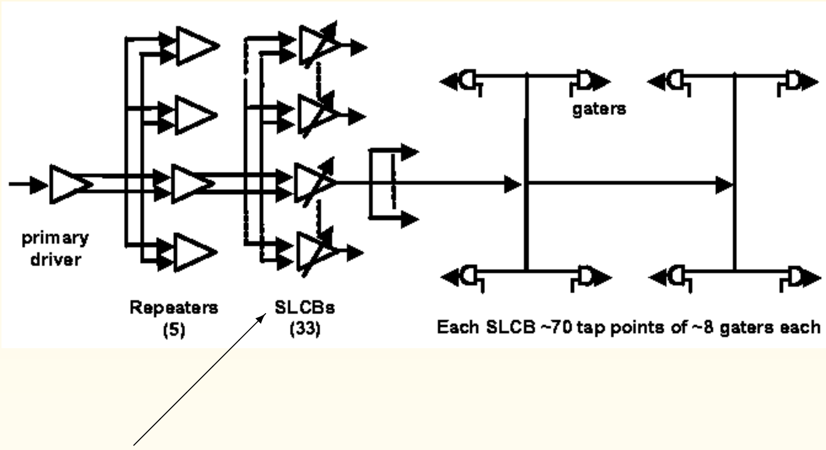


Source for the slides on Itanium: Intel/HP

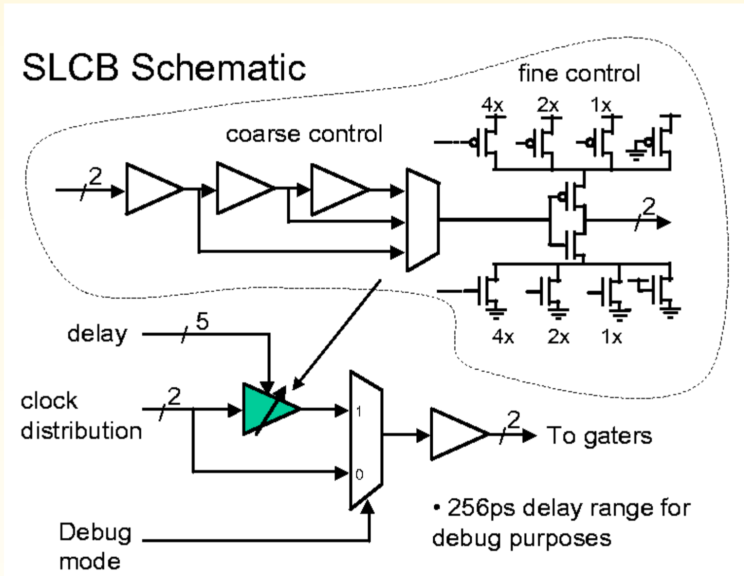
Clock Generation



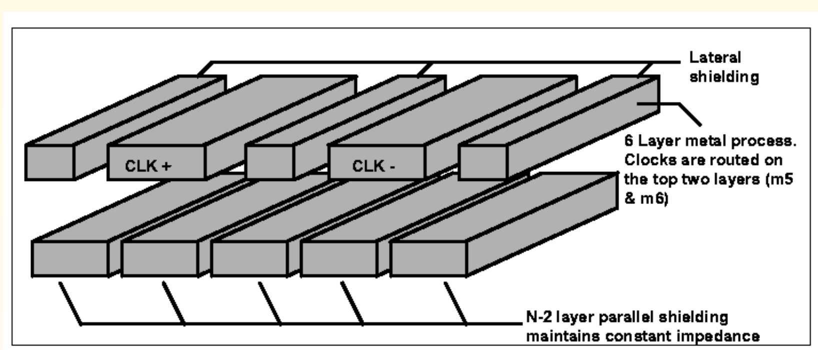
Core Clock Distribution



Second Level Clock Buffer (SLCB)



First Level Route Geometry



Measured Skew