

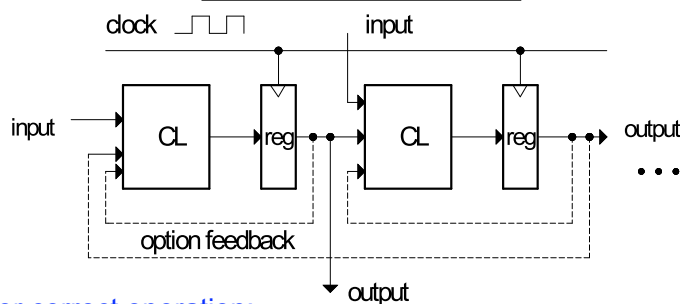
EECS150 - Digital Design

Lecture 18 - Circuit Timing (2)

March 17, 2010

John Wawrzynek

In General ...



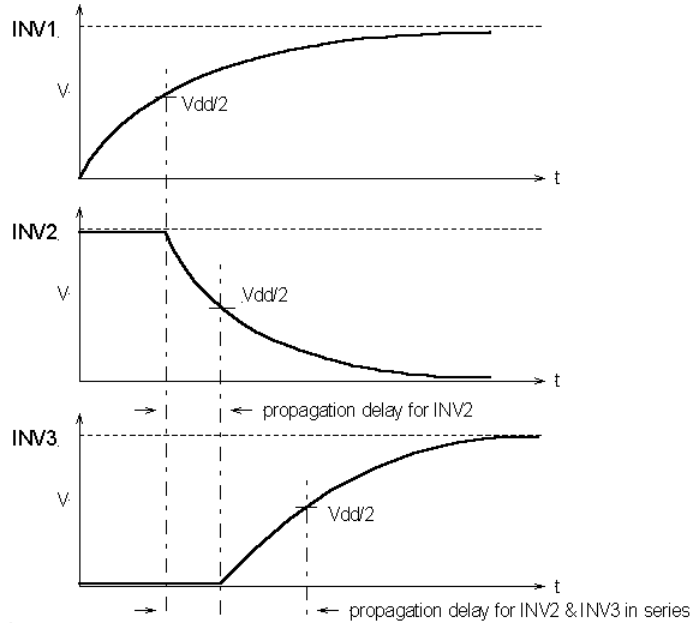
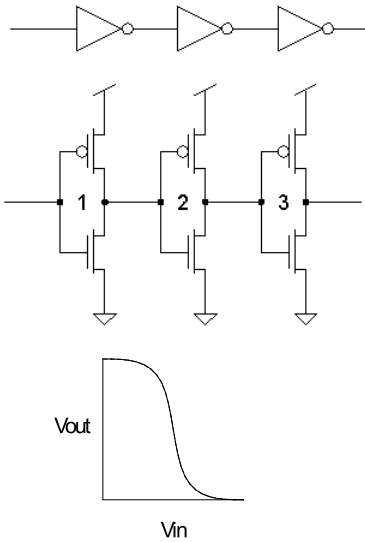
For correct operation:

$$T \geq \tau_{\text{clk} \rightarrow \text{Q}} + \tau_{\text{CL}} + \tau_{\text{setup}} \quad \text{for all paths.}$$

- How do we enumerate **all** paths?
 - Any circuit input or register output to any register input or circuit output?
- Note:
 - "setup time" for outputs is a function of what it connects to.
 - "clk-to-q" for circuit inputs depends on where it comes from.

Gate Delay is the Result of Cascading

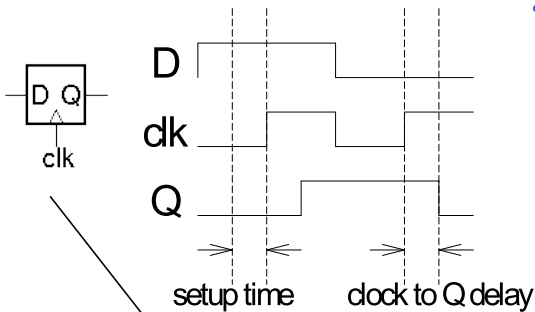
- Cascaded gates:



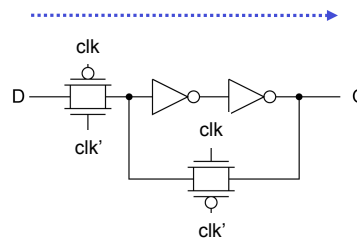
"transfer curve" for inverter.

In general:
 prop. delay = sum of individual prop. delays of gates in series.

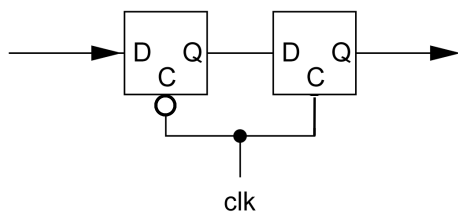
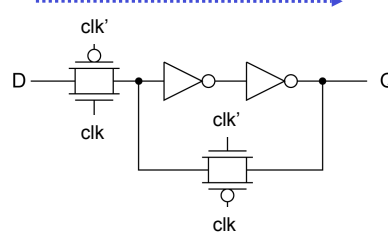
Delay in Flip-flops



- Setup time results from delay through *first* latch.



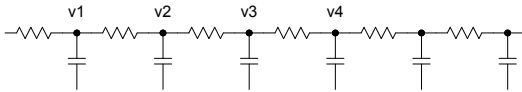
- Clock to Q delay results from delay through *second* latch.



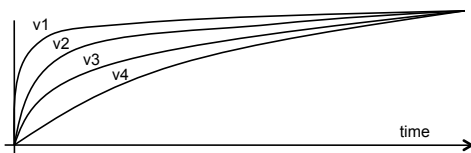
Wire Delay

- Even in those cases where the transmission line effect is negligible:

- Wires possess distributed resistance and capacitance



- Time constant associated with distributed RC is proportional to the square of the length

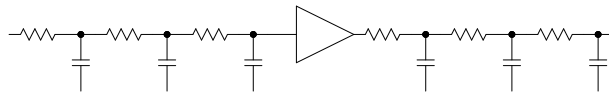


- For **short wires** on ICs, resistance is insignificant (relative to effective R of transistors), but C is important.

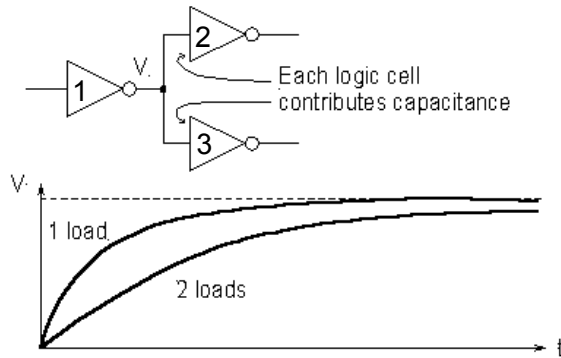
- Typically around half of C of gate load is in the wires.

- For **long wires** on ICs:

- busses, clock lines, global control signal, etc.
- Resistance is significant, therefore distributed RC effect dominates.
- signals are typically “rebuffered” to reduce delay:



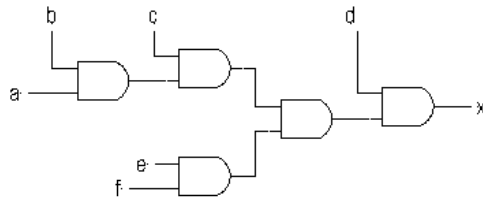
Delay and “Fan-out”



- The delay of a gate is proportional to its output capacitance. Connecting the output of gate one increases its output capacitance. Therefore, it takes increasingly longer for the output of a gate to reach the switching threshold of the gates it drives as we add more output connections.
- Driving wires also contributes to fan-out delay.
- What can be done to remedy this problem in large fan-out situations?

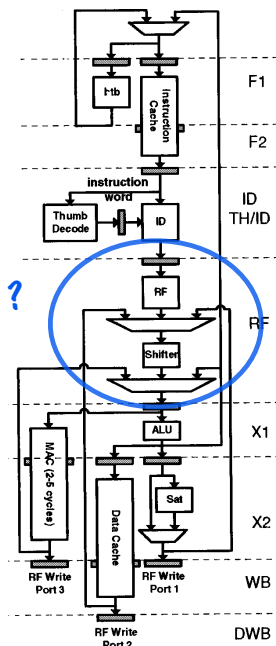
“Critical” Path

- *Critical Path*: the path in the entire design with the maximum delay.
 - This could be from state element to state element, or from input to state element, or state element to output, or from input to output (unregistered paths).
- For example, what is the critical path in this circuit?



- Why do we care about the *critical path*?

Searching for processor critical path



Must consider all connected register pairs, paths from input to register, register to output. Don't forget the controller.

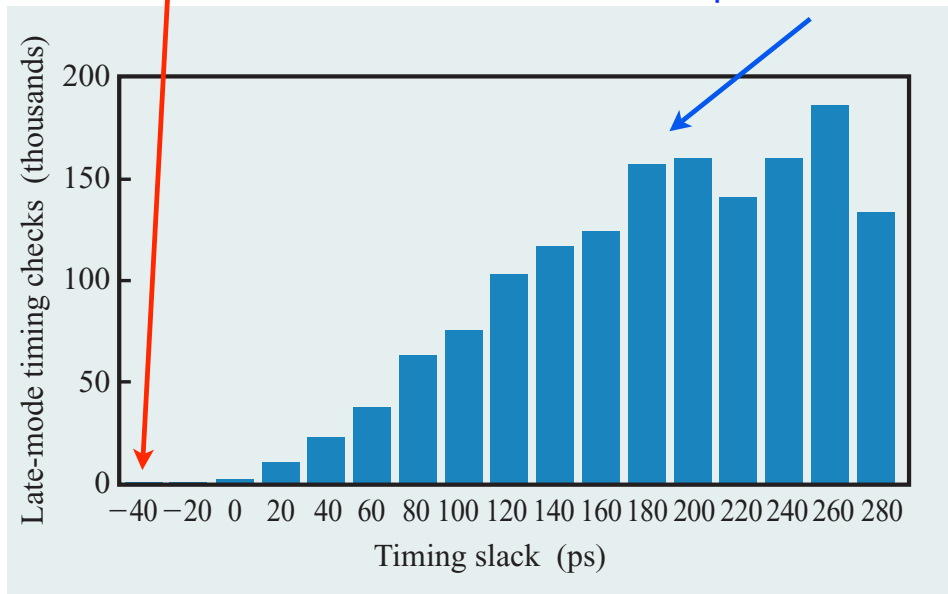
- Design tools help in the search.
 - **Synthesis tools** report delays on paths,
 - Special **static timing analyzers** accept a design netlist and report path delays,
 - and, of course, **simulators** can be used to determine timing performance.

Tools that are expected to **do something** about the timing behavior (such as synthesizers), also include provisions for specifying input arrival times (relative to the clock), and output requirements (set-up times of next stage).

Real Stuff: Timing Analysis

The critical path

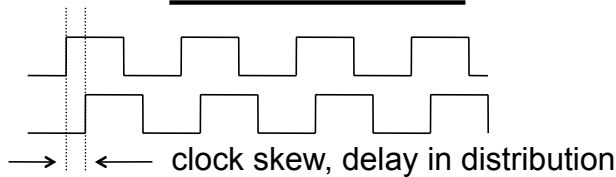
Most paths have hundreds of picoseconds to spare.



From "The circuit and physical design of the POWER4 microprocessor", IBM J Res and Dev, 46:1, Jan 2002, J.D. Warnock et al. Spring 2010 EECS150 - Lec18-timing(2)

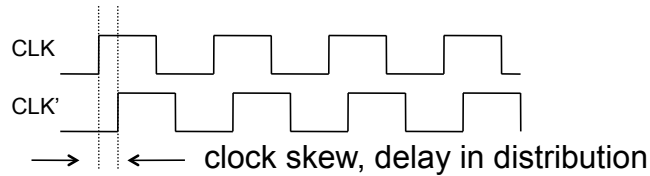
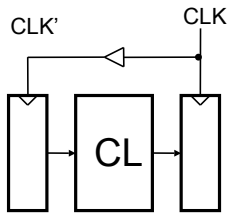
Page 9

Clock Skew



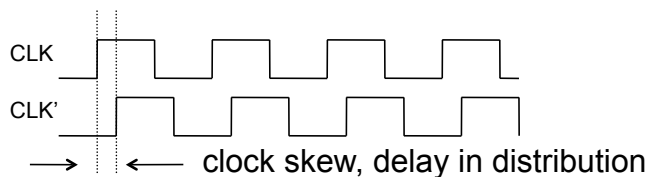
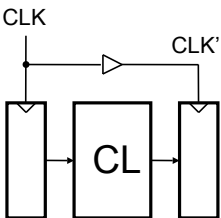
- Unequal delay in distribution of the clock signal to various parts of a circuit:
 - if not accounted for, can lead to erroneous behavior.
 - Comes about because:
 - clock wires have delay,
 - circuit is designed with a different number of clock buffers from the clock source to the various clock loads, or
 - buffers have unequal delay.
 - All synchronous circuits experience some clock skew:
 - more of an issue for high-performance designs operating with very little extra time per clock cycle.

Clock Skew (cont.)



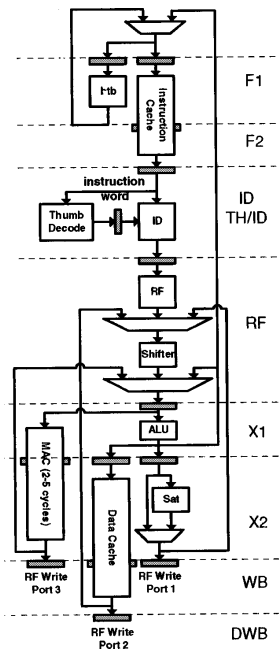
- If clock period $T = T_{CL} + T_{setup} + T_{clk \rightarrow Q}$, circuit will fail.
- Therefore:
 1. Control clock skew
 - a) Careful clock distribution. Equalize path delay from clock source to all clock loads by controlling wires delay and buffer delay.
 - b) don't "gate" clocks in a non-uniform way.
 2. $T \geq T_{CL} + T_{setup} + T_{clk \rightarrow Q} + \text{worst case skew}$.
- Most modern large high-performance chips (microprocessors) control end to end clock skew to a small fraction of the clock period.

Clock Skew (cont.)

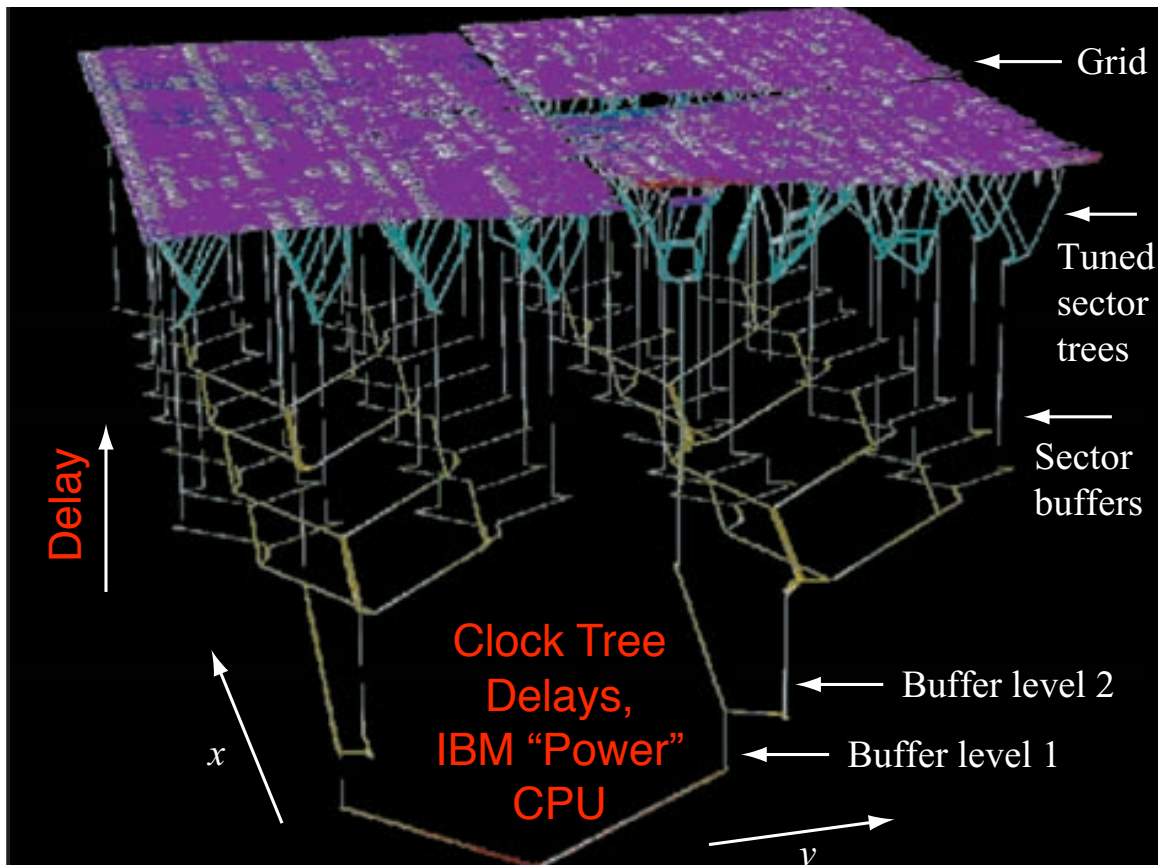
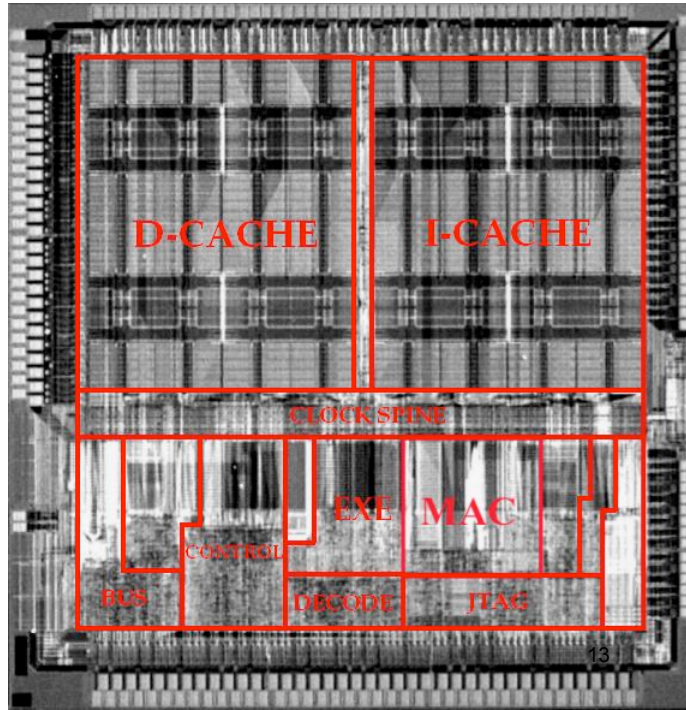


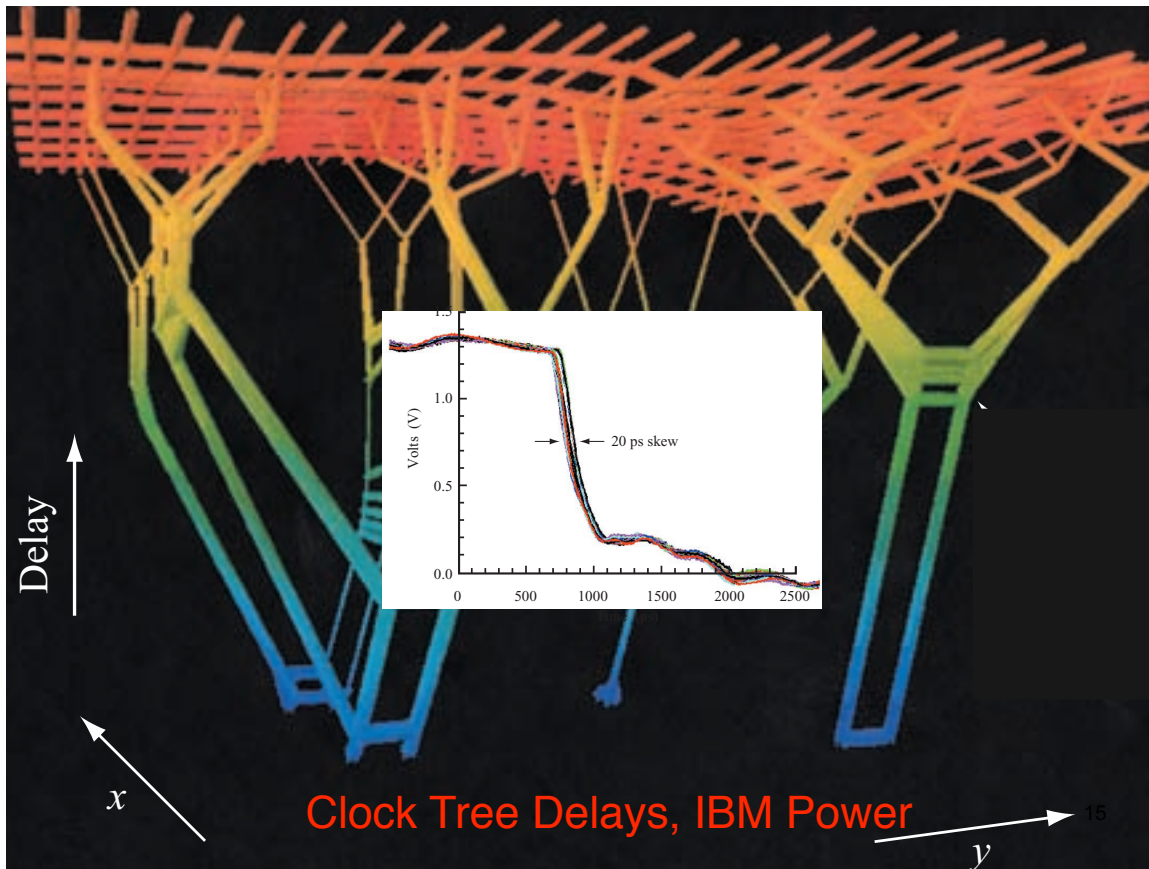
- **Note reversed buffer.**
- In this case, clock skew actually provides *extra time* (adds to the effective clock period).
- This effect has been used to help run circuits as higher clock rates. Risky business!

Real Stuff: Floorplanning Intel XScale 80200



Spring 2010

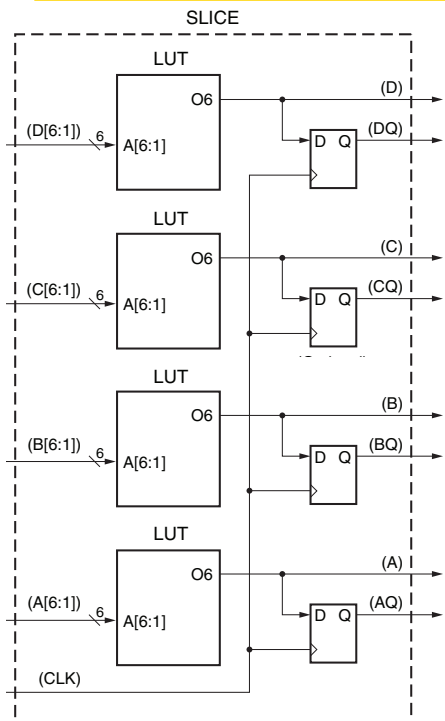




Timing in Xilinx Designs



From earlier lecture: Virtex-5 slice



6-LUT delay is **0.9 ns**

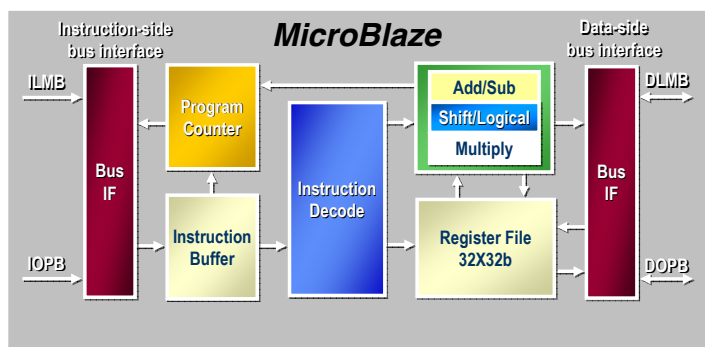
1.1 GHz toggle speed

128 x 32b LUT RAM
access time is **1.1 ns**

0.909 GHz toggle speed

But yet ...

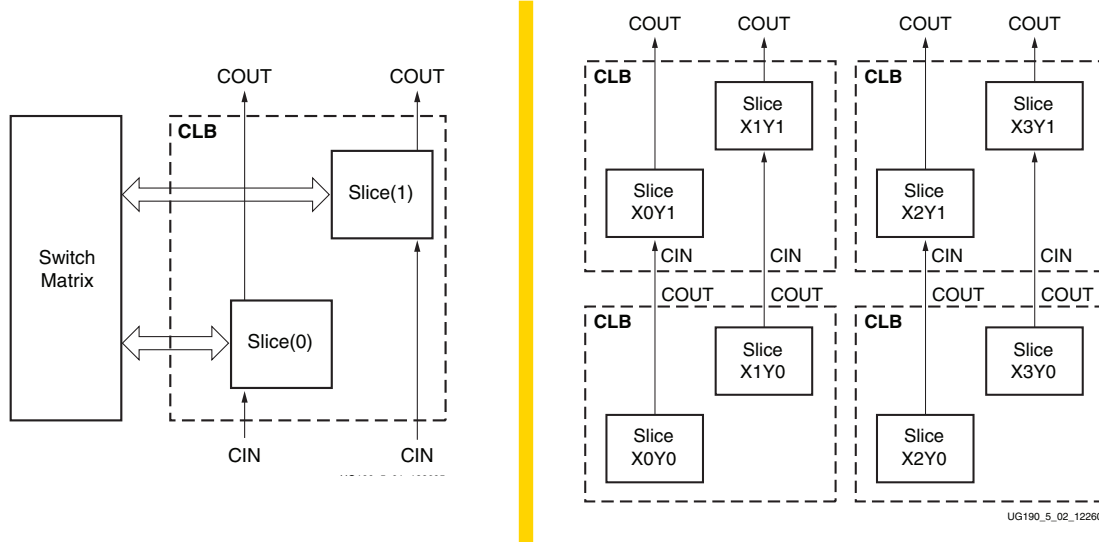
Xilinx CPU runs at 201 MHz ... 4.5x slower



- Better use of new LUTs
 - 1269 LUT4s in Virtex-4, MB 4.0
 - 1400 LUT6s in Virtex-5, MB 5.0
- from 3 stage -> 5 stage pipeline
- new processor: from 0.92 DMips/MHz to 1.14 DMips/MHz
- 180MHz -> 201 MHz
- 166 -> 230 Dhrystone Mips

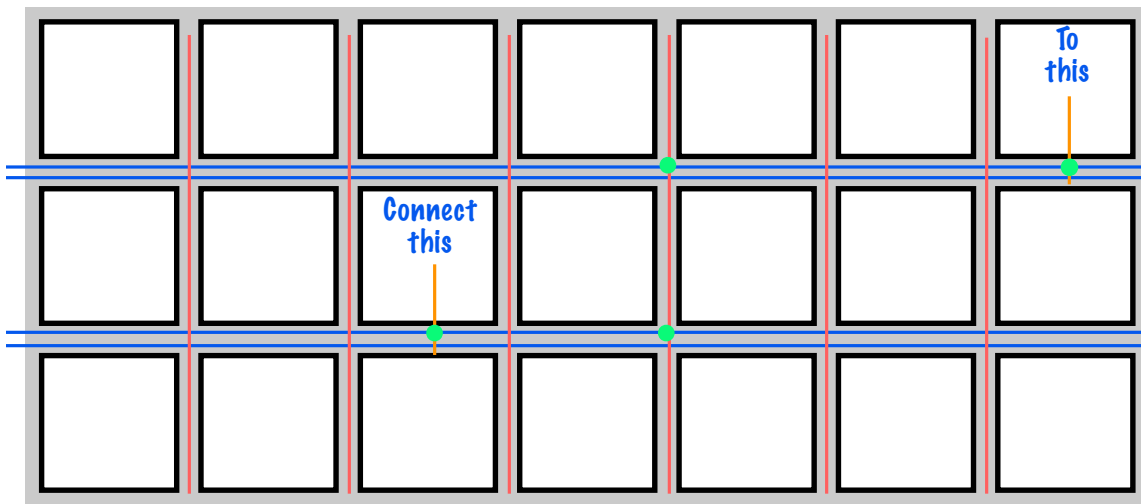
Major delay source: Interconnect

Slices define regular connections to the switching fabric, and to slices in CLBs above and below it on the die.



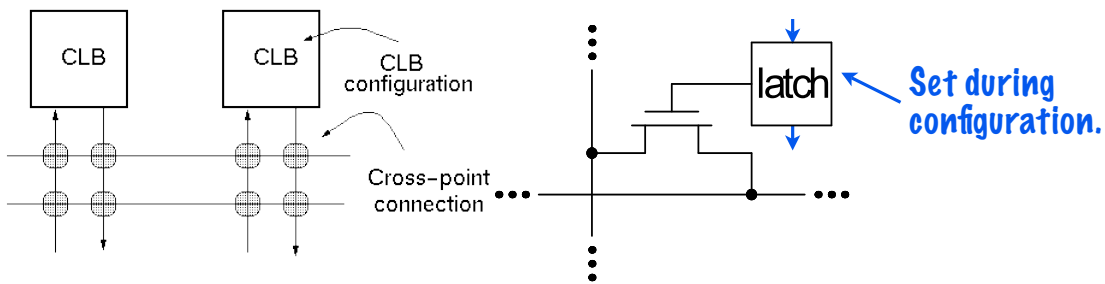
Simplified model of interconnect ...

Wires are slow because (1) each green dot is a transistor switch (2) path may not be shortest length (3) all wires are too long!



Delay in FPGA designs are particularly layout sensitive. Placement and routing tools spend most of their cycles in timing optimization. When Xilinx designs FPGA chips, wiring channels are optimized for (2) & (3).

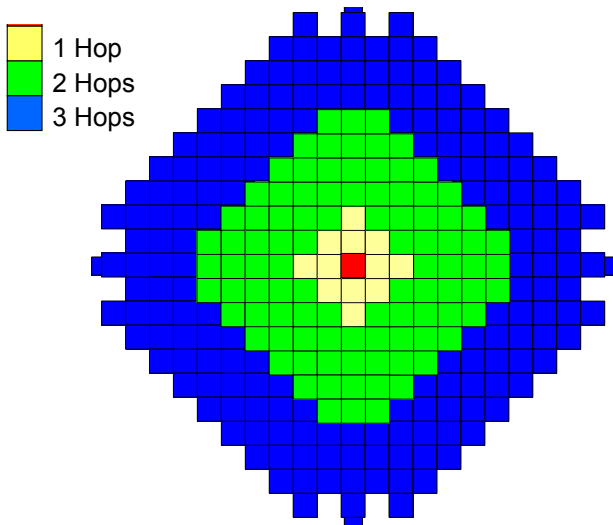
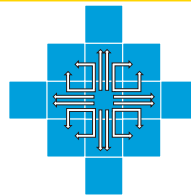
What are the green dots?



One flip-flop and a pass gate for each switch point. In order to have enough wires in the channels to wire up CLBs for most circuits, we need a lot of switch points! Thus, "80%+ of FPGA is for wiring".

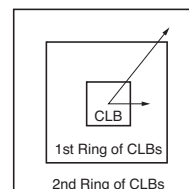
More realistic Virtex-5 model ...

1-hop wires to nearest neighbors



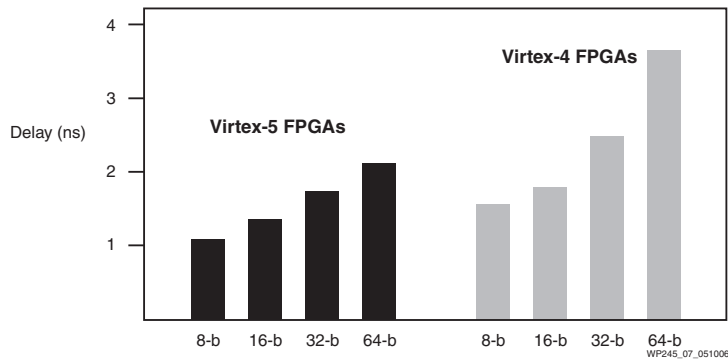
Virtex-5	
● 1st Ring	665 ps
● 2nd Ring	723 ps

WP245_04_050106



Timing for small building blocks ...

	Virtex-4 FPGA	Virtex-5 FPGA
6-Input Function ⁽¹⁾	1.1 ns	0.9 ns
Adder, 64-bit	3.5 ns	2.5 ns
Ternary Adder, 64-bit	4.3 ns	3.0 ns
Barrel Shifter, 32-bit	3.9 ns	2.8 ns
Magnitude Comparator, 48-bit	2.4 ns	1.8 ns
LUT RAM, 128 x 32-bit	1.4 ns	1.1 ns



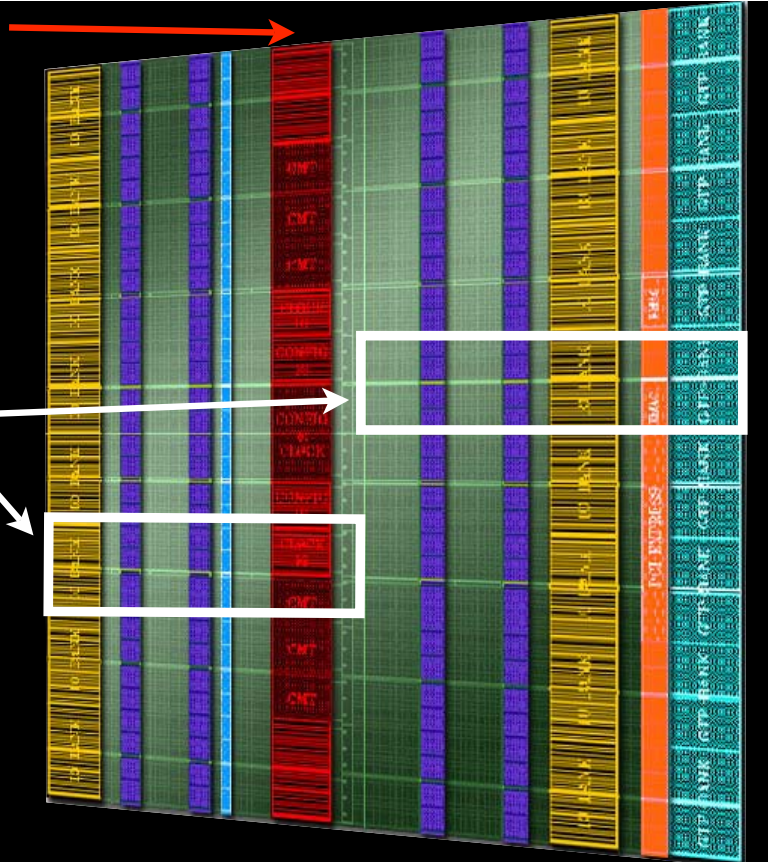
Clocking

Clock circuits live in **center** column.

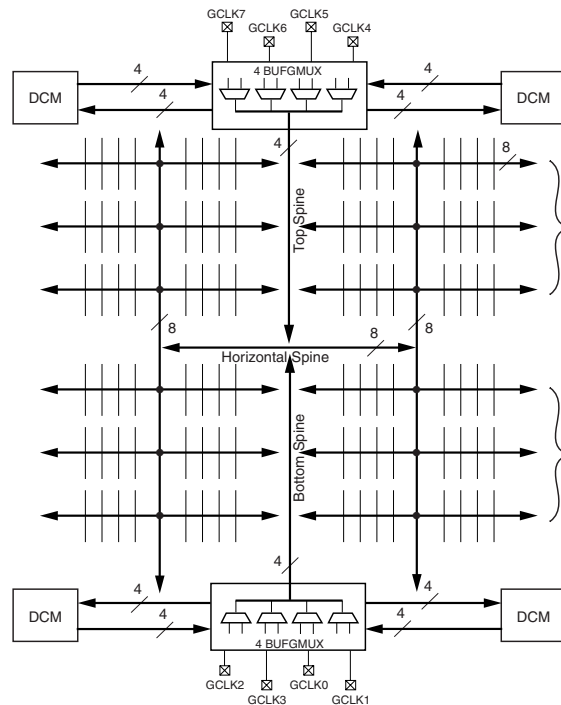
32 global clock wires go down the red column.

Any 10 may be sent to a clock region.

Also, 4 regional clocks (restricted functionality).



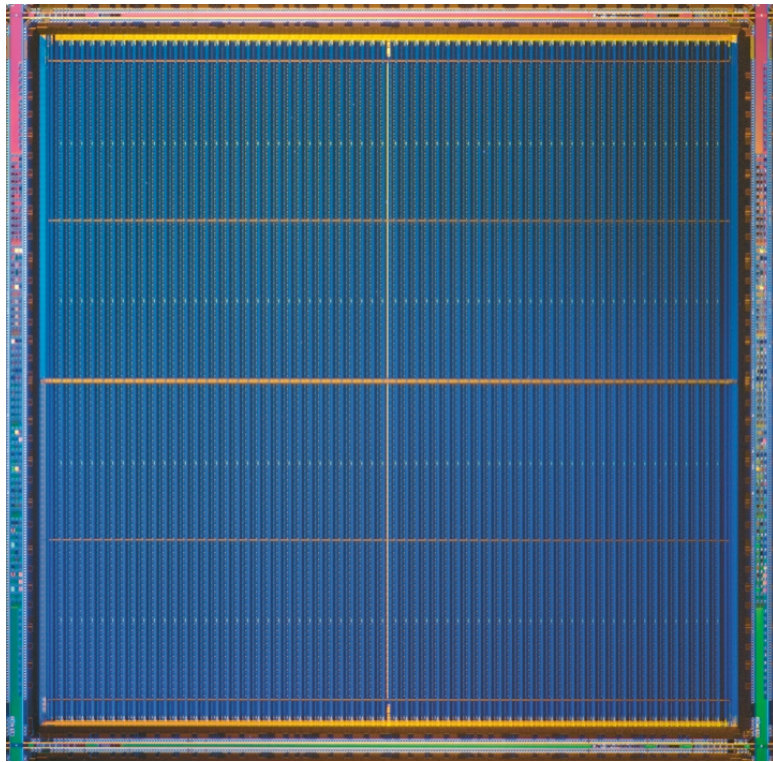
Clocks have dedicated wires (low skew)



From: Xilinx Spartan 3 data sheet. Virtex is similar.

Die photo:
Xilinx
Virtex

Gold wires
are the
clock tree.

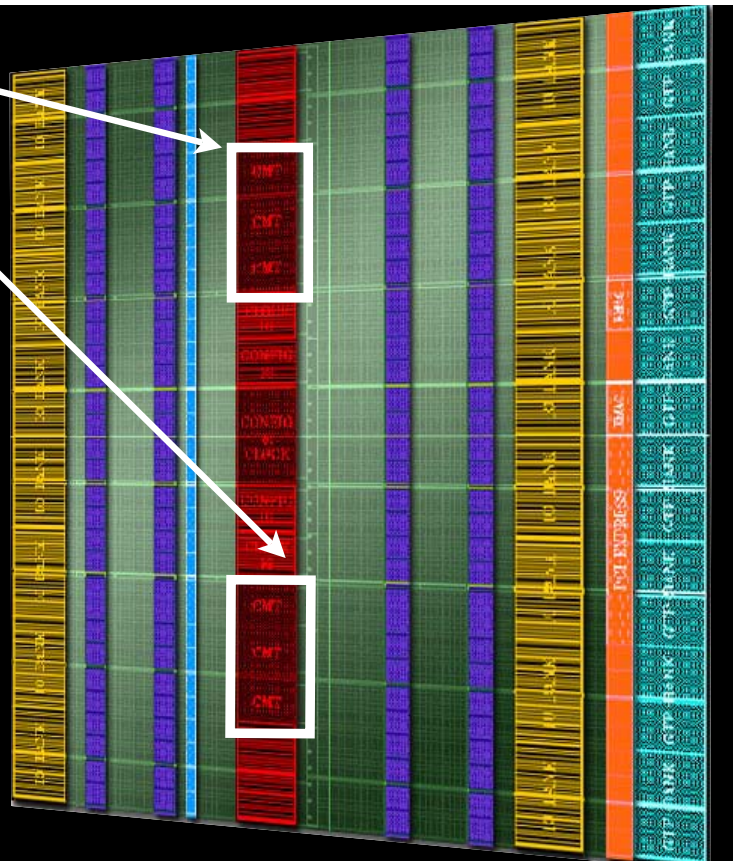


LX110T:

12
Digital Clock
Managers (DCM)

6 Phase Locked
Loops (PLL)

20 Clock I/O Pads



DCM: Clock deskew, clock phasing

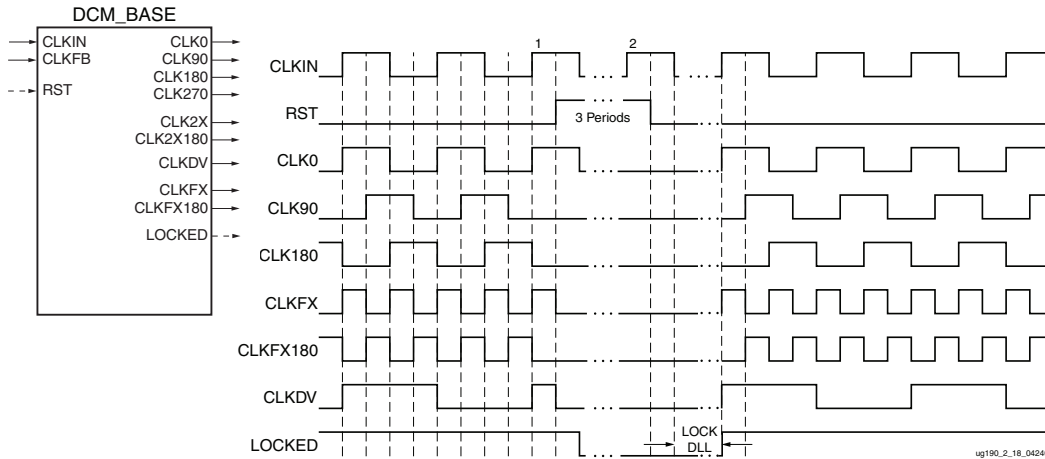


Figure 2-17: RESET/LOCK Example

DCM adjusts its output delay to synchronize the clock signal at the feedback clock input (CLKFB) to the clock signal at the input clock (CLKIN).

Important use is in “deskewing” on-chip clock distribution relative to input (board level) clock signal.

How it works: Delay-line feedback

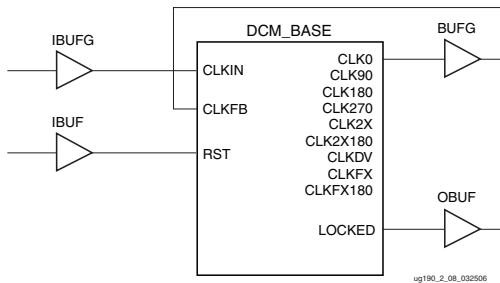


Figure 2-8: Standard Usage

