

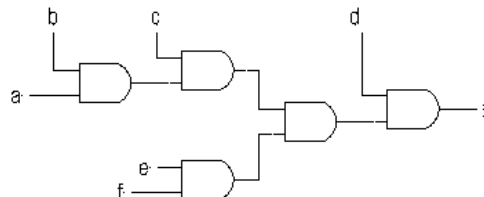
EECS150 - Digital Design

Lecture 17 - Circuit Timing (2)

March 19, 2013
John Wawrzynek

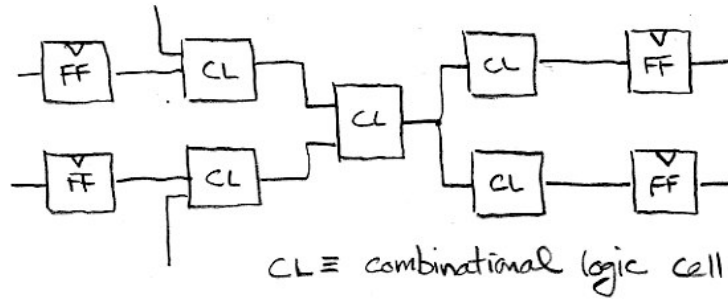
“Critical” Path

- *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*?

Components of Path Delay

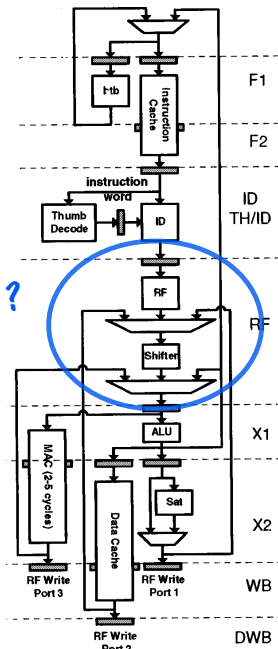


1. # of levels of logic
2. Internal cell delay
3. wire delay
4. cell input capacitance
5. cell fanout
6. cell output drive strength

Who controls the delay?

	Silicon foundry engineer	Cell Library Developer, FPGA-chip designer	CAD Tools (logic synthesis, place and route)	Designer (you)
1. # of levels			synthesis	RTL
2. Internal cell delay	physical parameters	cell topology, trans sizing	cell selection	
3. Wire delay	physical parameters		place & route	layout generator
4. Cell input capacitance	physical parameters	cell topology, trans sizing	cell selection	
5. Cell fanout			synthesis	RTL
6. Cell drive strength	physical parameters	transistor sizing	cell selection	instantiation (ASIC)

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.

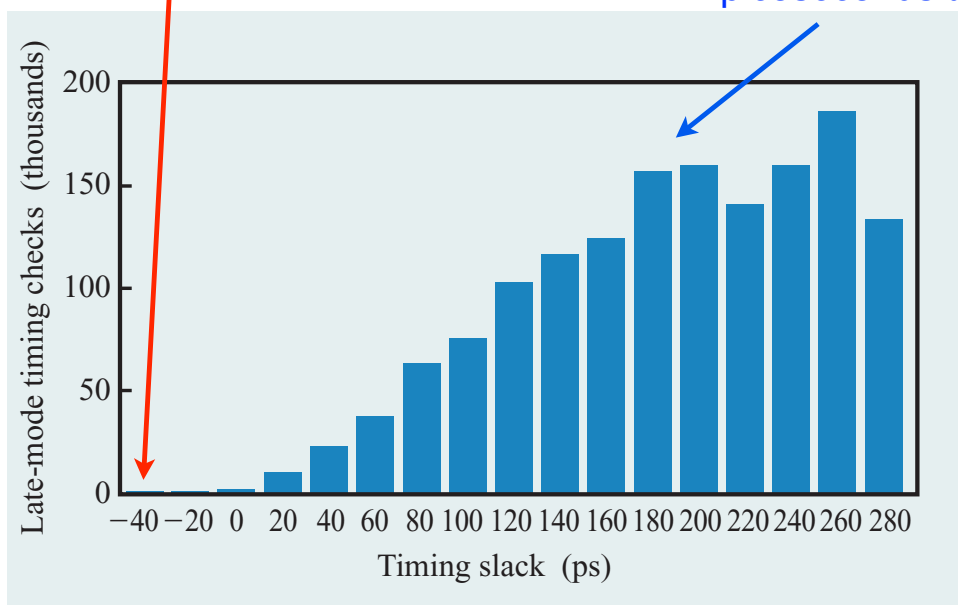
Tools that are expected to **do something** about the timing behavior (such as synthesizers), also include provisions for specifying input arrival times (relative to the clock), and output requirements (set-up times of next stage).

5

Real Stuff: Timing Analysis

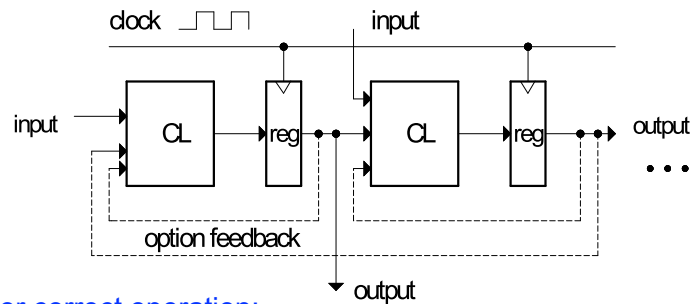
The critical path

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.

In General ...



For correct operation:

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

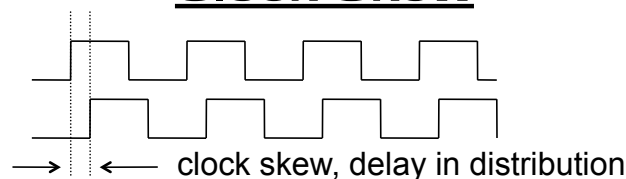
- How do we enumerate **all** paths?
 - Any circuit input or register output to any register input or circuit output?
- Note:
 - "setup time" for outputs is a function of what it connects to.
 - "clk-to-q" for circuit inputs depends on where it comes from.

Spring 2013

EECS150 - Lec17-timing[2]

Page 7

Clock Skew



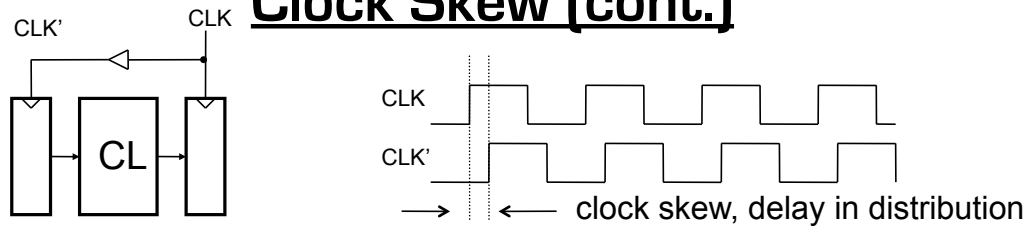
- Unequal delay in distribution of the clock signal to various parts of a circuit:
 - if not accounted for, can lead to erroneous behavior.
 - Comes about because:
 - clock wires have delay,
 - circuit is designed with a different number of clock buffers from the clock source to the various clock loads, 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.

Spring 2013

EECS150 - Lec17-timing[2]

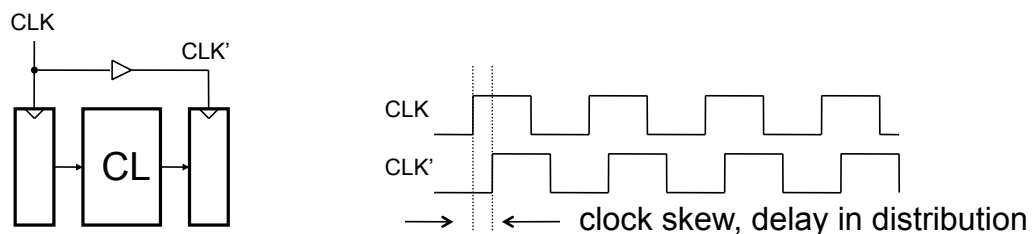
Page 8

Clock Skew (cont.)



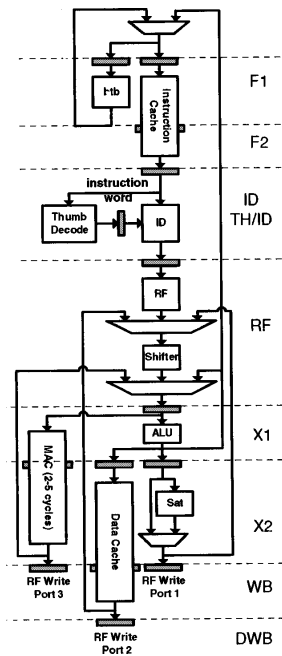
- If clock period $T = T_{CL} + T_{\text{setup}} + T_{\text{clk} \rightarrow Q}$, circuit will fail.
- Therefore:
 1. Control clock skew
 - a) Careful clock distribution. Equalize path delay from clock source to all clock loads by controlling wires delay and buffer delay.
 - b) don't "gate" clocks in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

Clock Skew (cont.)

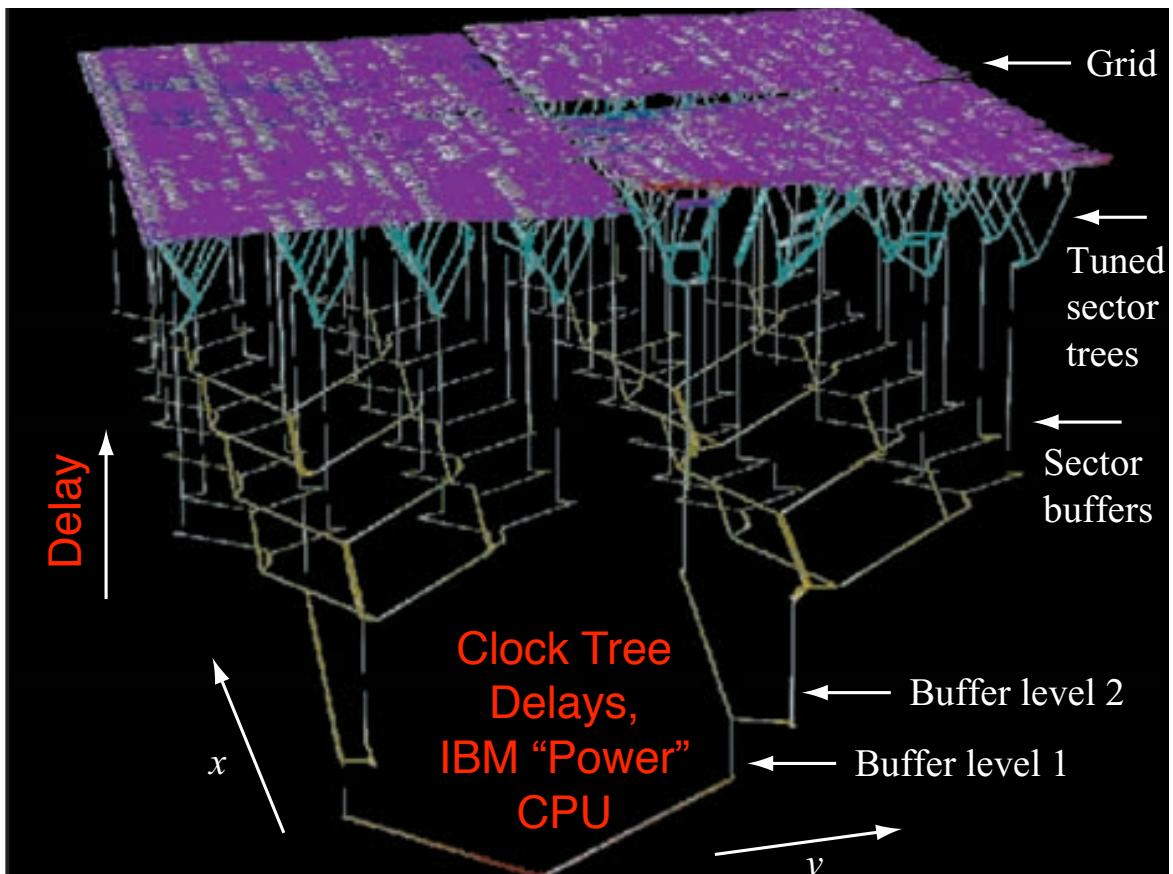
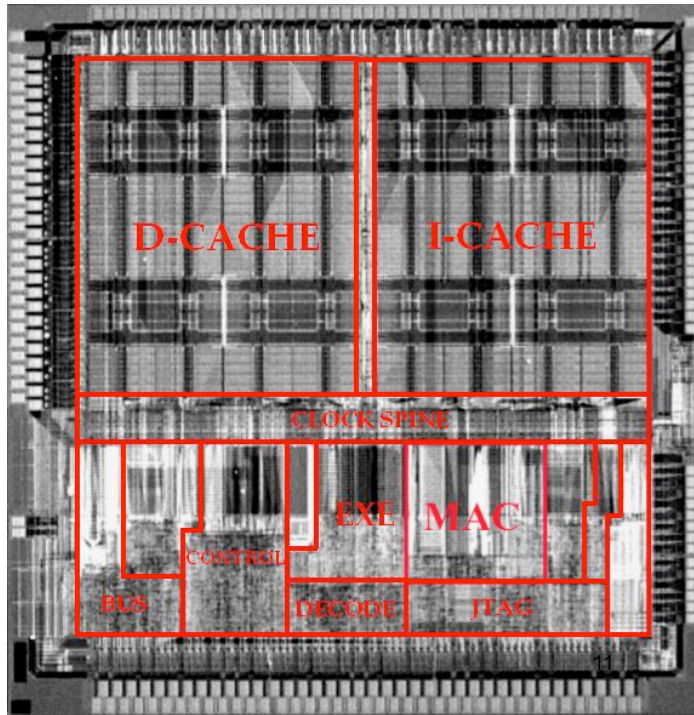


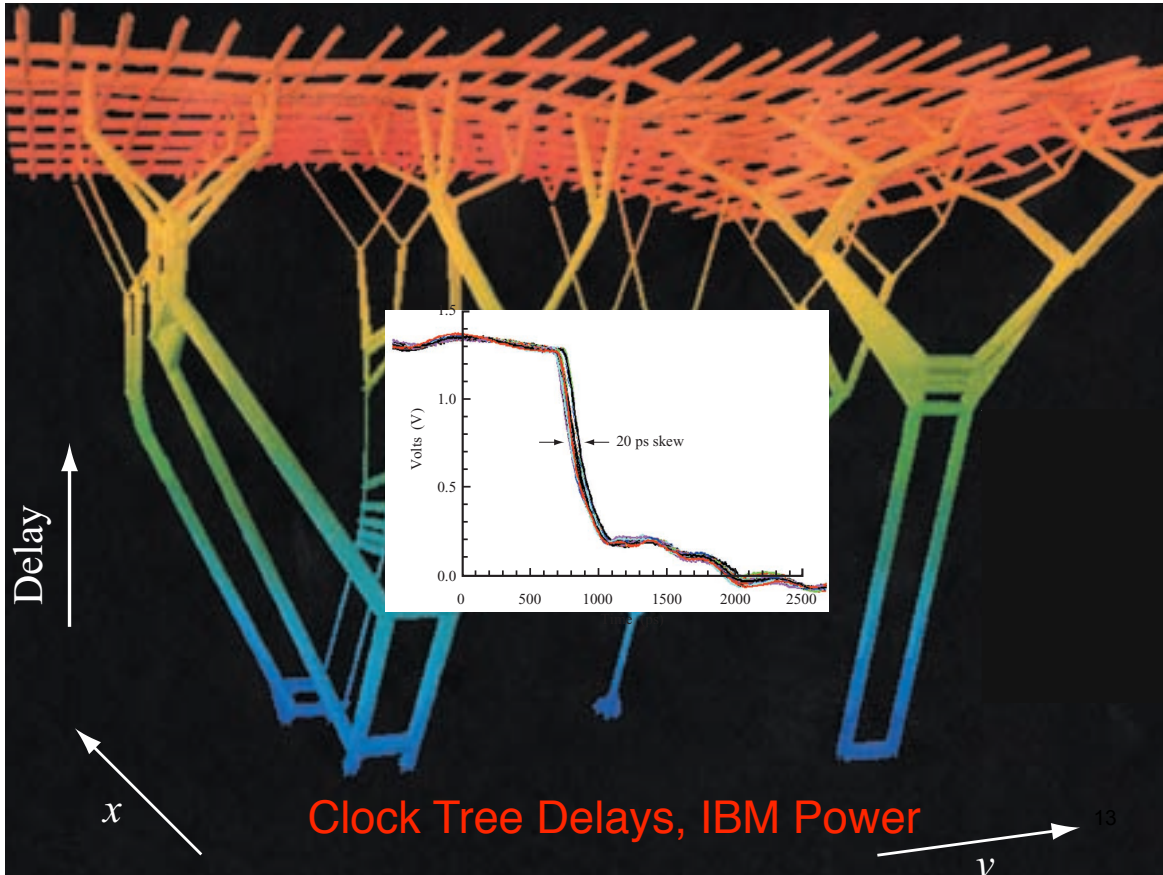
- **Note reversed buffer.**
- In this case, clock skew actually provides *extra time* (adds to the effective clock period).
- This effect has been used to help run circuits as higher clock rates. Risky business!

Real Stuff: Floorplanning Intel XScale 80200



Spring 2013

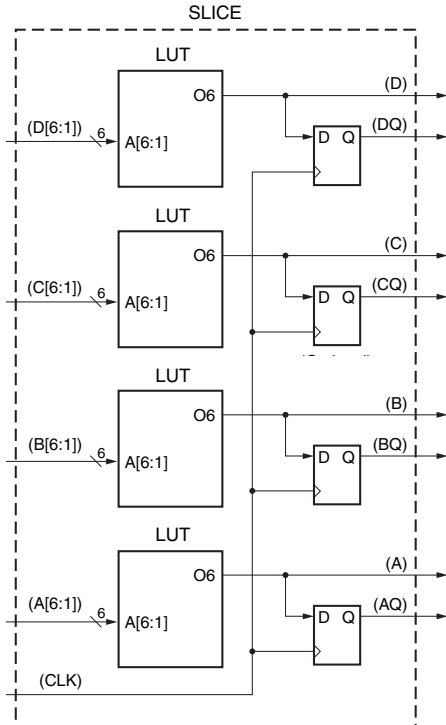




Timing in Xilinx Designs



From earlier lecture: Virtex-5 slice



6-LUT delay is **0.9 ns**

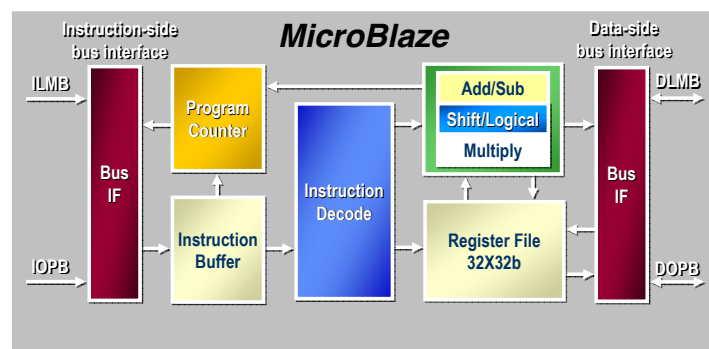
1.1 GHz toggle speed

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

0.909 GHz toggle speed

But yet ...

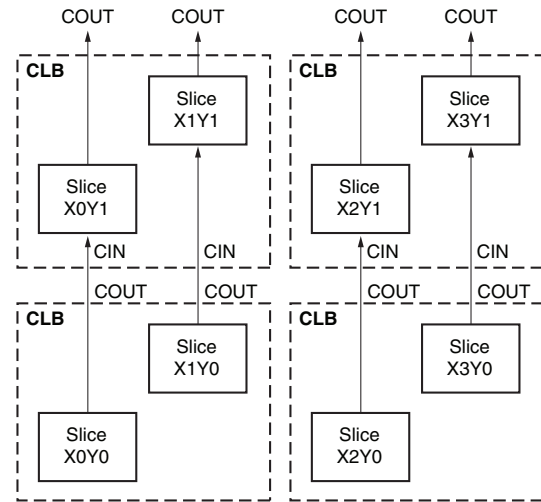
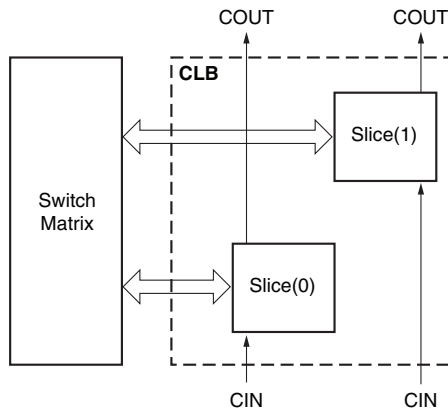
Xilinx CPU runs at 201 MHz ... 4.5x slower



- 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

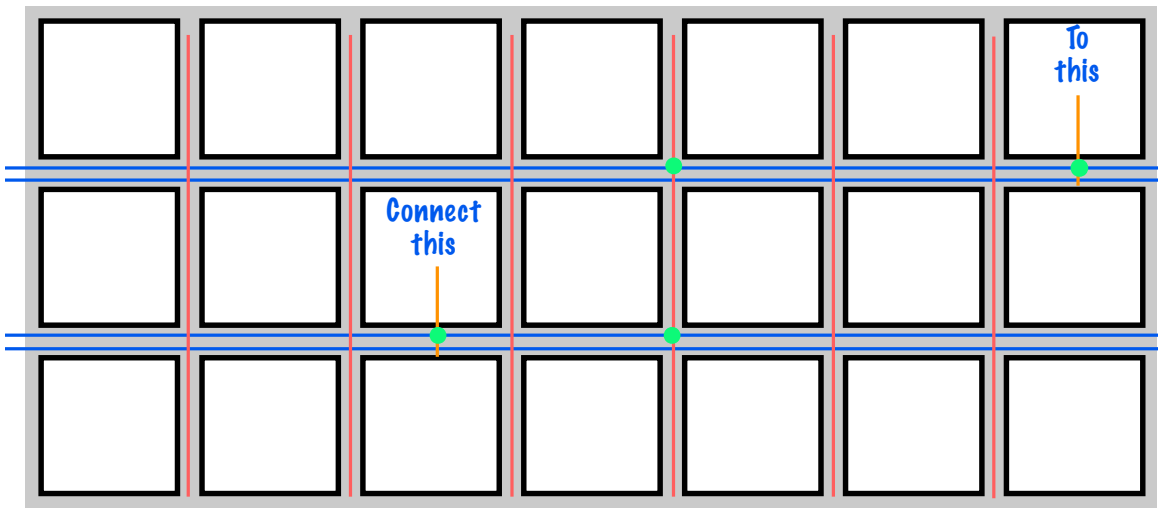
Slices define regular connections to the switching fabric, and to slices in CLBs above and below it on the die.



UG190_5_02_122605

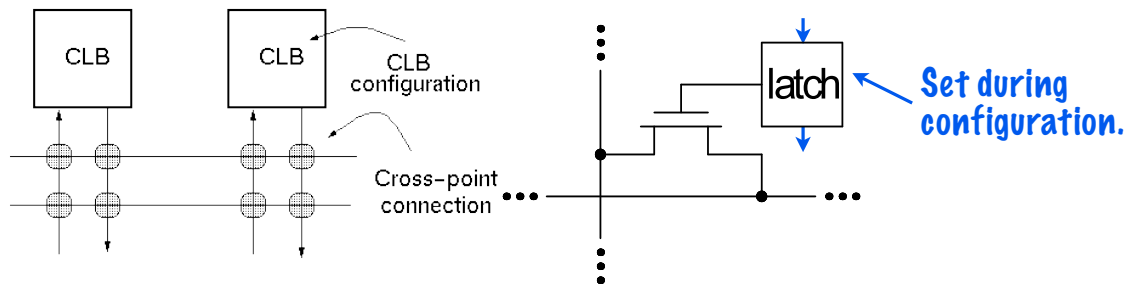
Simplified model of interconnect ...

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 there cycles in timing optimization. When Xilinx designs FPGA chips, wiring channels are optimized for (2) & (3).

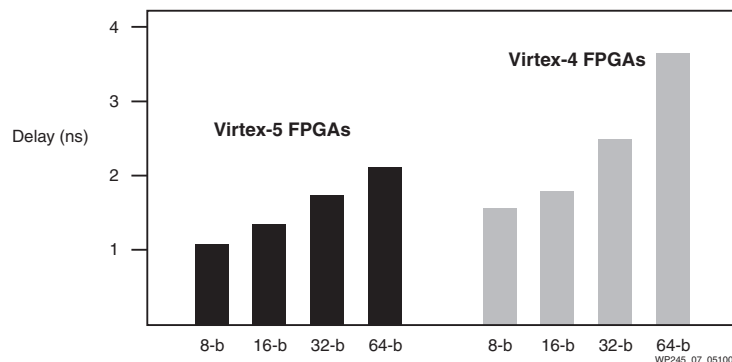
What are the green dots?



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".

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

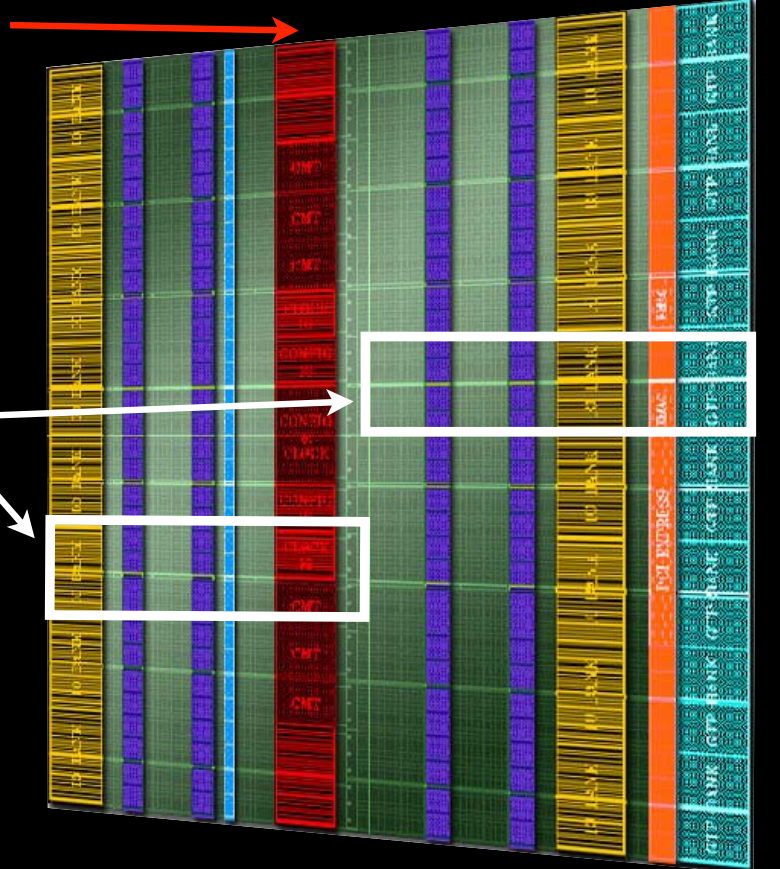


Clock circuits live in **center** column.

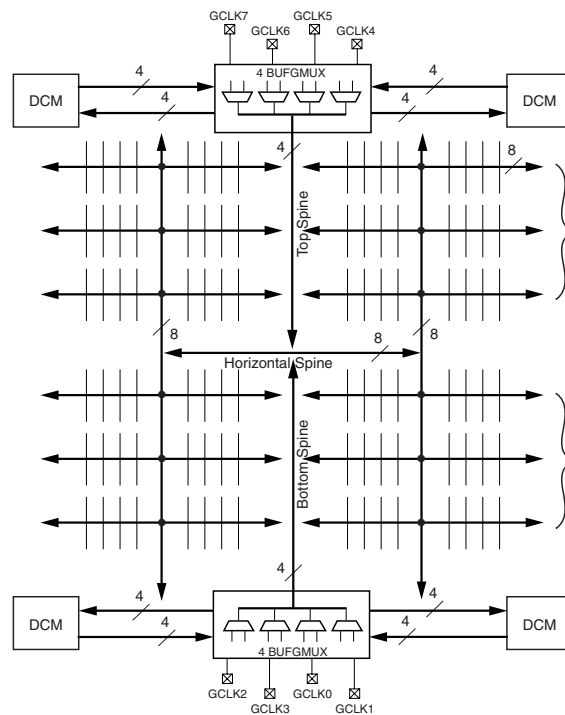
32 global clock wires go down the red column.

Any 10 may be sent to a clock region.

Also, 4 regional clocks (restricted functionality).



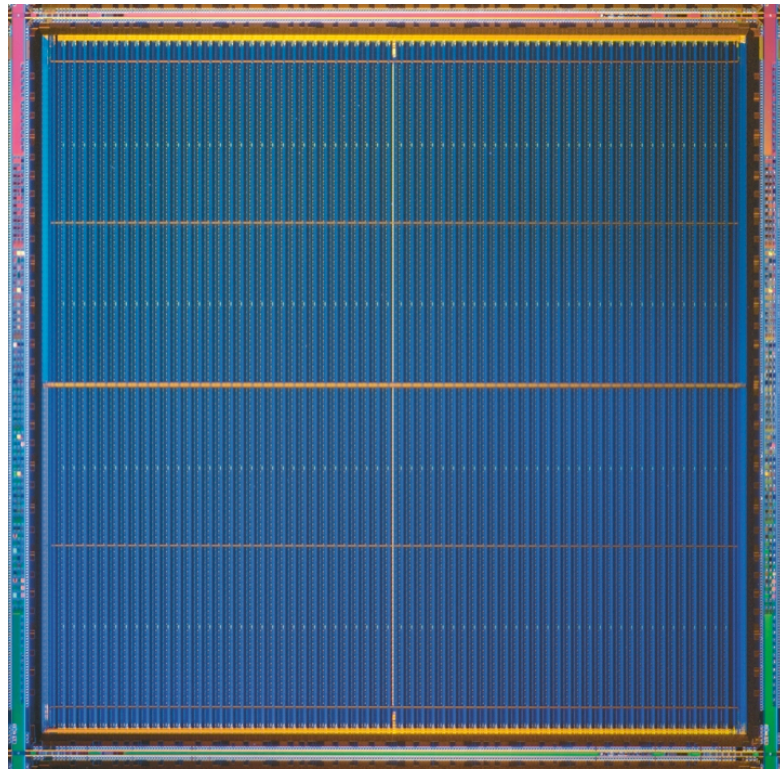
Clocks have dedicated wires (low skew)



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

Die photo:
Xilinx
Virtex

Gold wires
are the
clock tree.

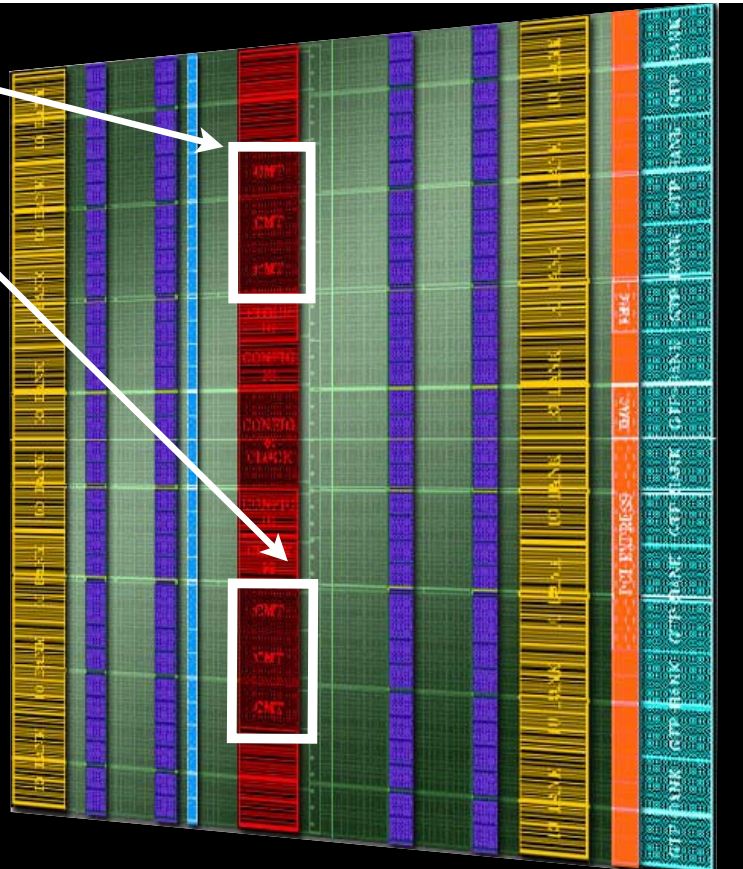


LX110T:

12
Digital Clock
Managers (DCM)

6 Phase Locked
Loops (PLL)

20 Clock I/O Pads



Summary - what you need to know

1. **Performance** is directly related to clock frequency. Usually higher clock frequency results in higher performance (more ops/sec).
2. **Max clock frequency** is determined by the worst case path (the "critical path").
3. To first order the **delay of a path** is the sum of the delays of the parts in series (FF output: clk-to-Q, total combinational logic delay, FF input: setup time), plus some extra for worst case clock skew ("uncertainty").

Summary (cont.)

4. **CAD algorithms** exist for analyzing circuit timing. The algorithms are used as part of logic synthesis to help the tool achieve a timing target. Also, standalone tools exist to analyze circuit timing at various abstraction levels: post synthesis gate (or LUT) netlist, placed and routed, physical layout.
 - a. **Static analysis** traces paths through the circuit estimating the delay. Limitations are that real delay is data dependent (due to gate level behavior and because of logic "masking"). Therefore static tools need to be pessimistic.
 - b. **Simulators** often include timing information so that actual delays can be reported based on real data patterns. Of course, accuracy is limited by the accuracy of the models used.

Summary (cont.)

5. The "knobs" that you can turn as a designer to affect circuit timing:
- a. IC process. Finer geometry IC processes allow circuits to run faster because for a given voltage smaller transistors conduct more current.
 - b. IC layout. Layouts can be compressed to minimize wiring capacitance. Transistor "folding" can reduce transistor capacitances for a given performance. (Although modern processes have stringent constraints on layout styles)
 - c. Transistor-level circuits. Transistors can be sized to minimize delay. In special cases (RAM arrays for instance) special precharge circuits and "sense amplifiers" can speed signal transmission.

Summary (cont.)

5. The "knobs" that you can turn as a designer to affect circuit timing (cont):
- a. Combinational logic circuit level. Logic factoring (2-level versus multi-level). Reduction tree balancing. Alternative arithmetic structures (adder example) These are some combination of the logic synthesis tools and designer.
 - b. Synchronous digital system level. Retiming and pipelining can shorten the critical path.
 - c. As an FPGA designer: part choice, turn knobs on tools to trade performance for area, optimize Verilog RTL (micro-architecture, sub-circuit architecture)