



Designing with FPGAs

Beyond Bigger, Faster, Cheaper...

Peter Alfke

Xilinx, Inc.

peter.alfke@xilinx.com

XILINX, Virtex, and Spartan are registered trademarks of Xilinx, Inc.

All other trademarks are the property of their respective owners

© 2000, 2001 Xilinx, Inc. All Rights Reserved



Designing with FPGAs

- ◆ **Why FPGAs ?**
- ◆ **Basic Architecture and New Features**
- ◆ **Designing for High Speed**
- ◆ **Designing for Signal Integrity**
- ◆ **Designing with BlockROMs**
- ◆ **Designing for Low Power**
- ◆ **Designing for Security**
- ◆ **Asynchronous Design Issues**
- ◆ **Tips and Tricks from the Xilinx Archives**
- ◆ **Virtex-II, the newest FPGA Family**
- ◆ **What's Coming Later in 2001?**

Why FPGAs ?

- ◆ **Ideal for customized designs**
 - *Product differentiation in a fast-changing market*
- ◆ **Offer the advantages of high integration**
 - *High complexity, density, reliability*
 - *Low cost, power consumption, small physical size*
- ◆ **Avoid the problems of ASICs**
 - *high NRE cost, long delay in design and testing*
 - *increasingly demanding electrical issues*

***Fast Time-to-Market,
fast response to market changes***

FPGA Advantages

- ◆ **Very fast custom logic**
 - *massively parallel operation*
- ◆ **Faster than microcontrollers and microprocessors**
 - *much faster than DSP engines*
- ◆ **More flexible than dedicated chipsets**
 - *allows unlimited product differentiation*
- ◆ **More affordable and less risky than ASICs**
 - *no NRE, minimum order size, or inventory risk*
- ◆ **Reprogrammable at any time**
 - *in design, in manufacturing, after installation*

Makimoto's Wave

- ◆ 1957 to '67 **Standard** discrete devices (transistors, diodes)
- ◆ 1967 to '77 **Custom** LSI for calculators, radio, TV
- ◆ 1977 to '87 **Standard** microprocessors, custom software
- ◆ 1987 to '97 **Custom** logic in ASICs
- ◆ 1997 to '07 **Standard** Field-Programmable devices

We are in the early part of the FPGA cycle

— Tsugio Makimoto, formerly Hitachi,
Chairman of the Technology Board of Sony Semiconductor Network Co.

User Expectations

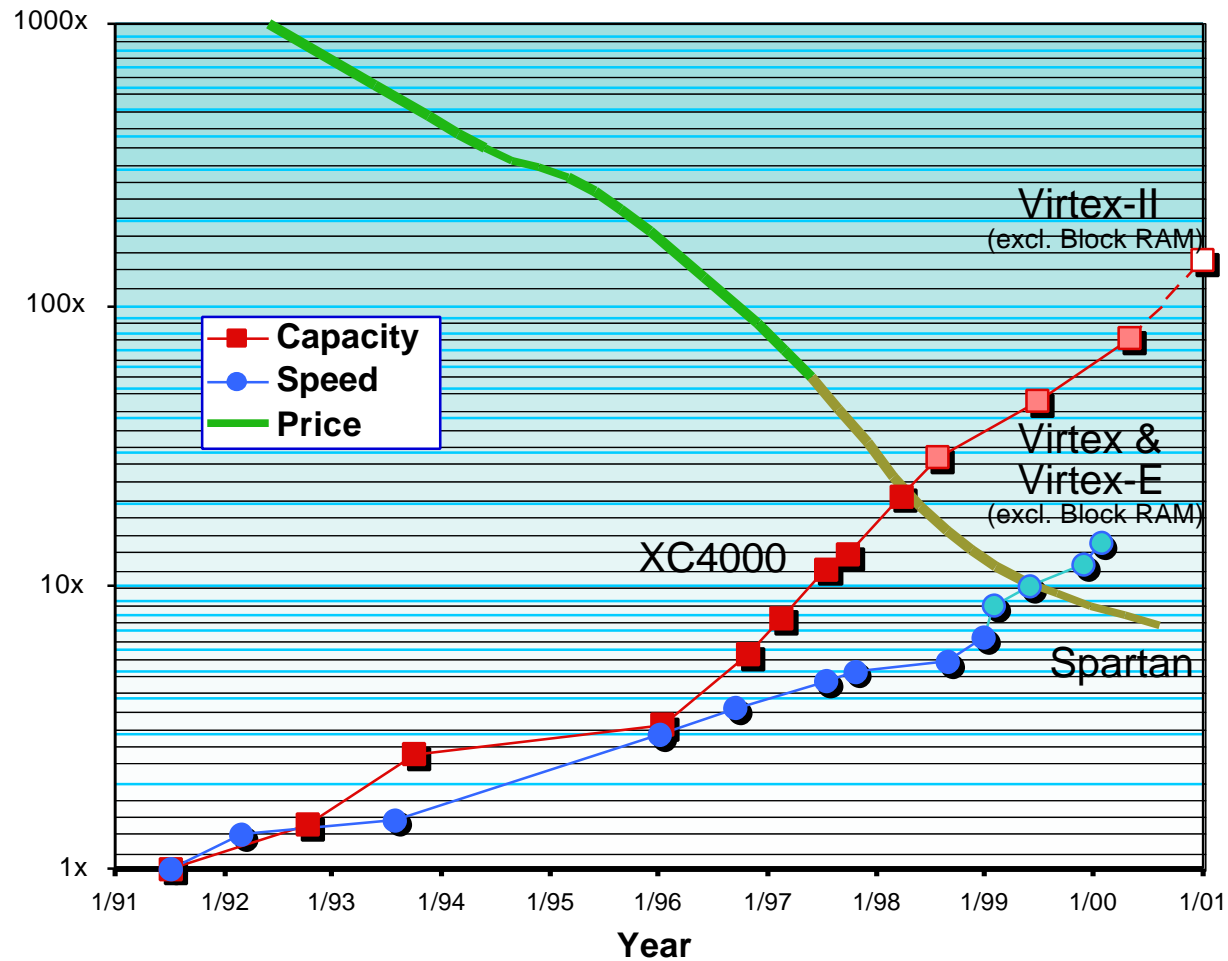
- ◆ **Logic capacity at reasonable cost**
 - *100,000 to a several million gates*
 - *On-chip fast RAM*
- ◆ **Clock speed**
 - *150 MHz and above, global clocks, clock management*
- ◆ **Versatile I/O**
 - *To accommodate a variety of standards*
- ◆ **Design effort and time**
 - *synthesis, fast compile times,*
 - *tested and proven cores*
- ◆ **Power consumption**
 - *must stay within reasonable limits*

Bigger, Faster, Cheaper FPGAs

- ◆ **Millions of gates**
 - *>1 million RAM bits*
- ◆ **>200 MHz system speed,**
 - *800 Mbps I/O*
- ◆ **From 0.3 ¢ to 3¢ per Logic Cell (LUT plus flip-flop)**
 - *Lowest for SpartanXL in high volume and simplest package*
 - *Highest for Virtex-II in low volume*

***“FPGAs have evolved from glue logic
to system platforms”***

A Decade of Progress



Three Pillars of Progress

◆ Technology

- *smaller geometries, more and faster transistors*
- *better defect densities, larger chips, larger wafers, lower cost*

◆ Architecture

- *system features: fast carry, memory, clock management*
- *hierarchical interconnect, controlled-impedance I/O*

◆ Design Methodology

- *powerful and reliable cores, faster compilation*
- *modular, team-based design, internet-based tools*

Basic Architecture and New Features

Beyond Bigger, Faster, Cheaper

On-chip RAM

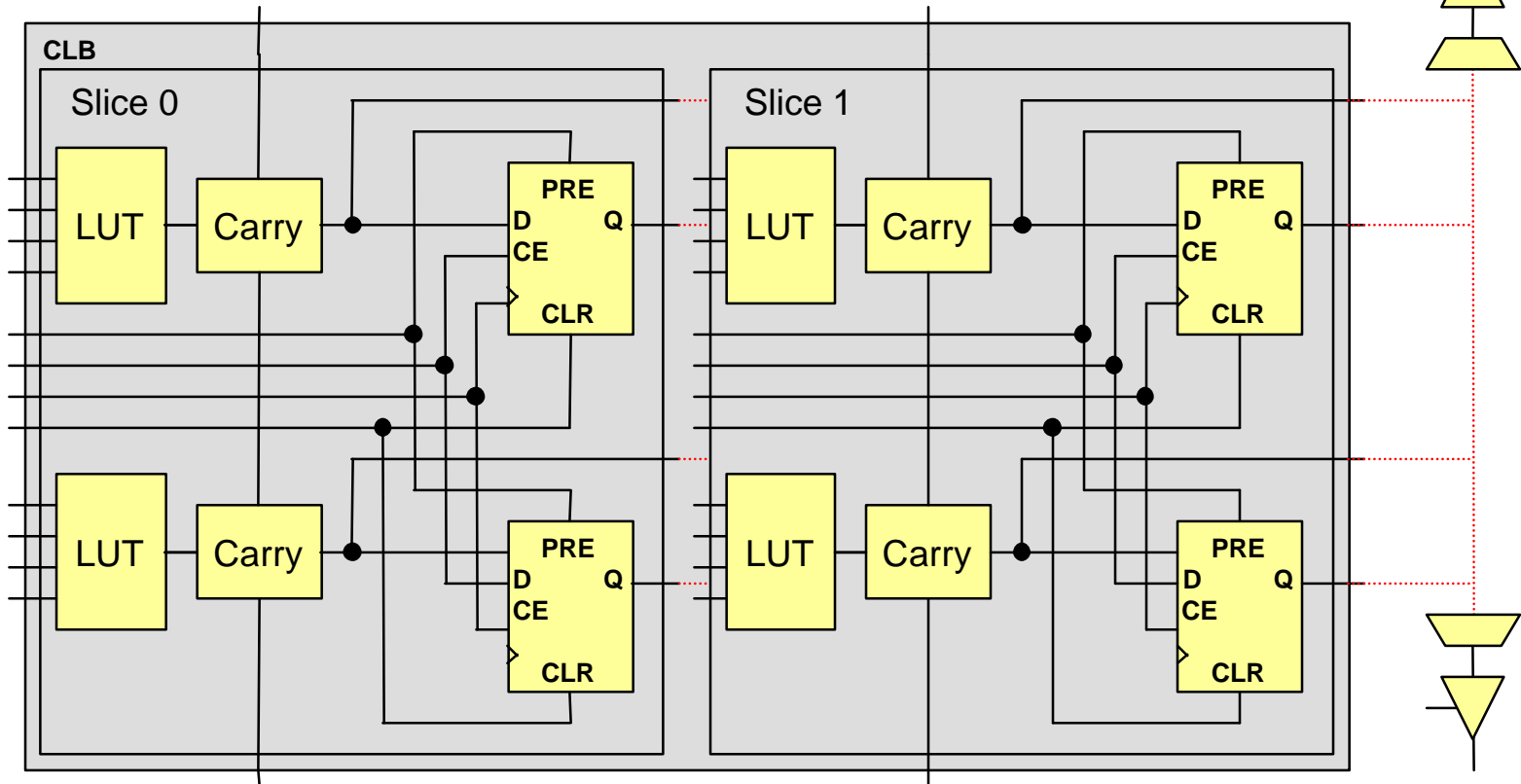
Efficient Arithmetic

Clock Management

Multi-standard I/O

Virtex-II, the next generation

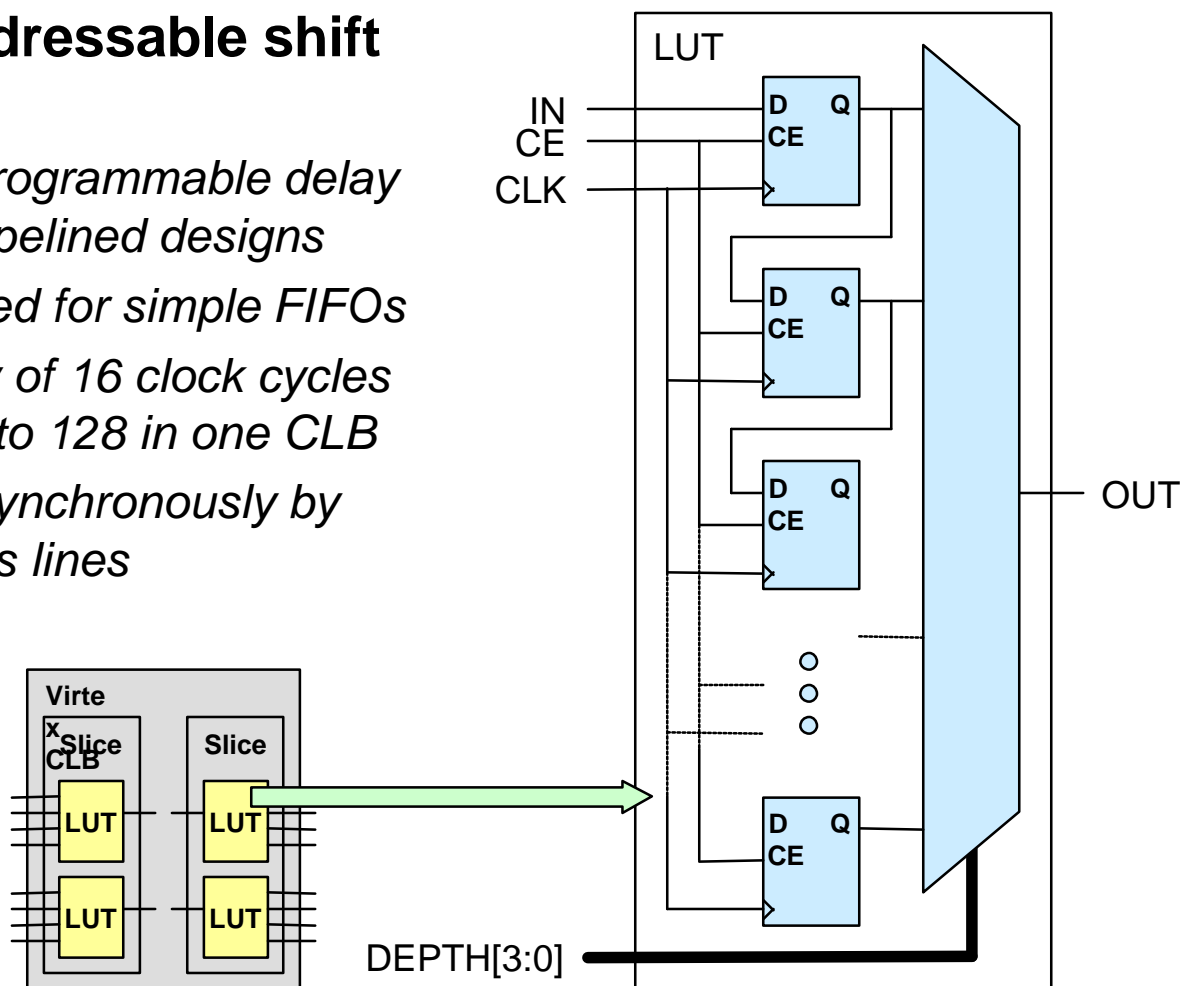
Simplified CLB Structure



- ◆ **Two slices in each Virtex CLB, four slices in each Virtex-II CLB**
 - *Two BUFTs associated with each CLB, accessible by all CLB outputs*
 - *Fast dedicated carry logic runs vertically up*

Shift Register LUT

- ◆ **Dynamically addressable shift register (SRL)**
 - *Ultra-efficient programmable delay for balancing pipelined designs*
 - *Can also be used for simple FIFOs*
 - *Maximum delay of 16 clock cycles in one LUT, up to 128 in one CLB*
 - *Can be read asynchronously by toggling address lines*



SRL16 Applications

- ◆ **1...16-bit shift register in one LUT**
 - *Up to 128 bits in one CLB*
- ◆ **Pipeline compensation (different length per branch)**
- ◆ **FIFO, pseudo-random number generator (LFSR)**
- ◆ **Serial frame synchronizer**
- ◆ **Running average calculator**
- ◆ **Pulse generator and clock divider**
- ◆ **Pattern generator, state machine**
- ◆ **Website:**
 - *<http://support.xilinx.com/support/techxclusives/SRL16-techxclusive2.htm>*

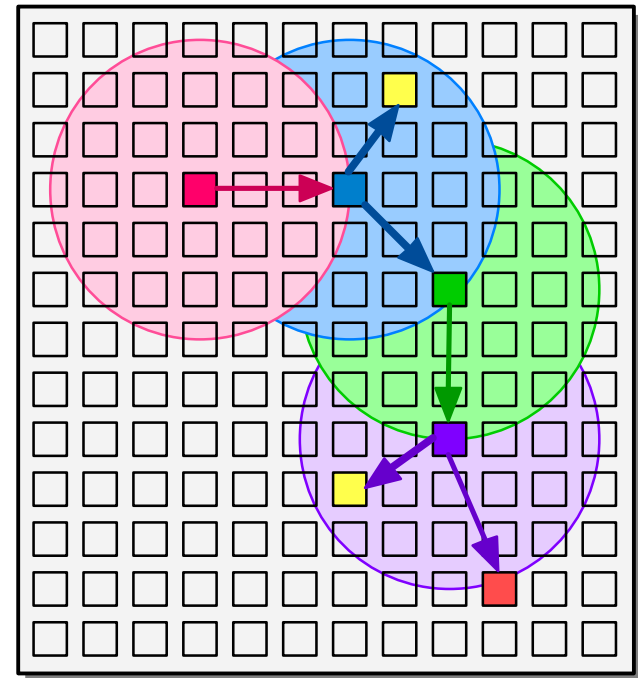
Dedicated Fast Carry

- ◆ **64-bit adders would require 128 levels of logic**
 - *Expensive complex carry schemes would be needed to preserve performance without using Carry Logic*
- ◆ **Virtex minimizes the carry propagation delay**
 - *< 100 ps per bit pair, <50 ps per bit, includes routing*
- ◆ **Fast adders, accumulators, and counters**
 - *24-bit operation at up to 300 MHz in Virtex-II*
 - *64-bit operation at up to 190 MHz in Virtex-II*
- ◆ **Fully synchronous operation**
 - *Same speed for add/subtract, accumulate, or count*

Fast Logic Needs Fast Routing

- ◆ Typical designs need a routing delay of < 1.5 ns
- ◆ Virtex delivers this performance
- ◆ Virtex-II is even faster
- ◆ Delay is independent of direction
 - *Dependably short delays provided by large numbers of short interconnect resources*

Vector-based Interconnect



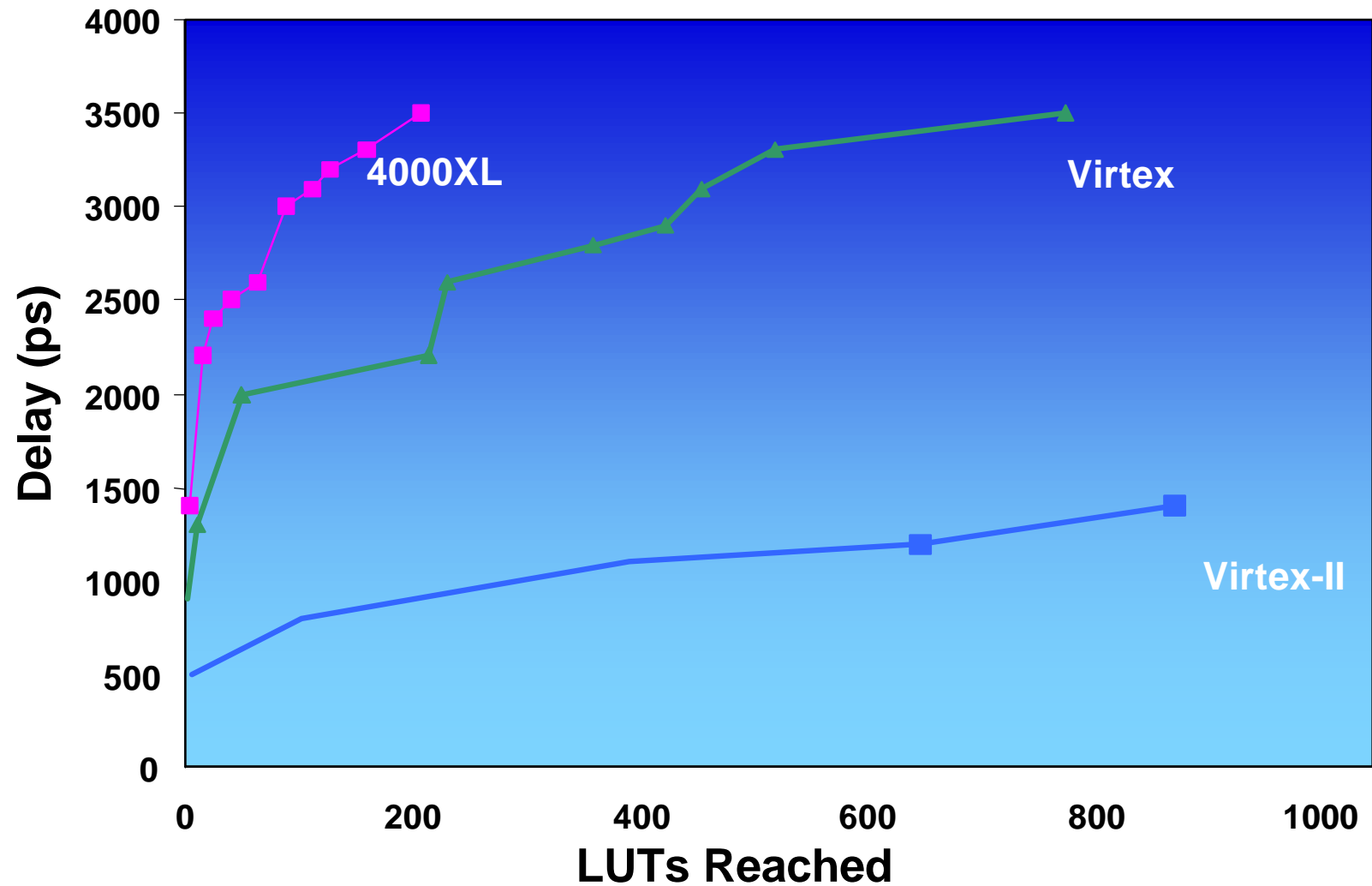
The circles show 1.4-ns routing delay

Virtex-II Provides Fast Routing

- ◆ Each Hex line spans six CLB rows or columns
- ◆ Each Hex interconnect delay < 300 ps
 - *three cascaded Hex lines span 18 columns in any direction*
- ◆ In 1 ns, a CLB output can reach 576 other CLBs
 - *i.e. 4,608 other LUTs*

***The center of an XC2V500,
can reach any logic or RAM input
within less than one nanosecond***

ActiveInterconnect Technology



On-Chip RAM

- ◆ **Up to 120,000 Four-Input Look-Up Tables in Virtex-II**
 - *Each 16-bit ROM, RAM or shift register*
 - 0.5 ns combin. delay, 0.5 ns set-up time, 0.5 ns clock-to-Q
- ◆ **Up to 192 dual-ported synchr. BlockRAMs**
 - *Each 4096 bits in Virtex, 18K bits in Virtex-II*
 - <3 ns access time, >200 MHz operation
- ◆ **Fast interface to external RAM**
 - *Up to 840 Mbp I/O data transfer rate (420 MHz DDR)*

Efficient Arithmetic

- ◆ **Dedicated Carry**

- *For adders, accumulators, counters*
- *<50 ps per bit incremental carry delay*
- *200 MHz operation over 64 bits*

- ◆ **2-s complement multipliers in Virtex-II**

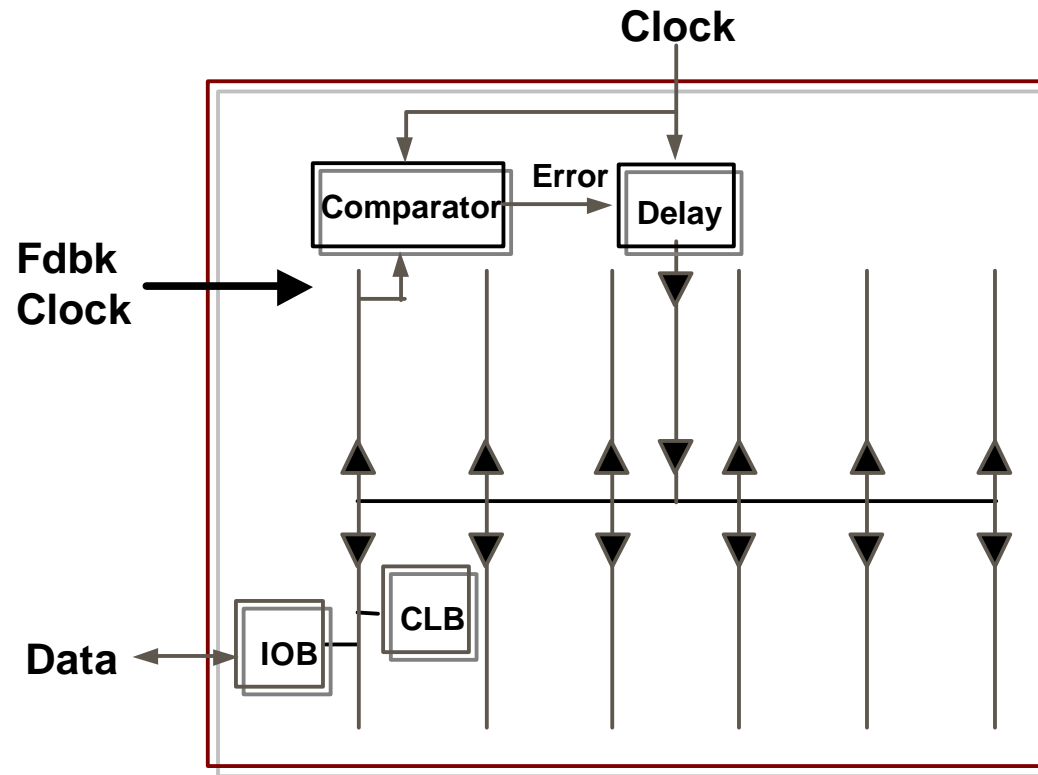
- *18 x 18 bits in <7 ns, 8 x 8 in 4 ns*
 - *Faster pipelined operation will be supported mid 2001*
- *Powerful and efficient for DSP*
- *Up to 192 independent multipliers*
- *Four in the smallest device, XC2V40*

Clock Management with DLL

- ◆ **Eliminates on-chip clock delay**
 - *Can also eliminate on-board clock delay*
- ◆ **Frequency division and multiplication**
- ◆ **Phase-coherent outputs**
- ◆ **Frequency modulation to reduce RFI**

Solves the speed problem of large chips

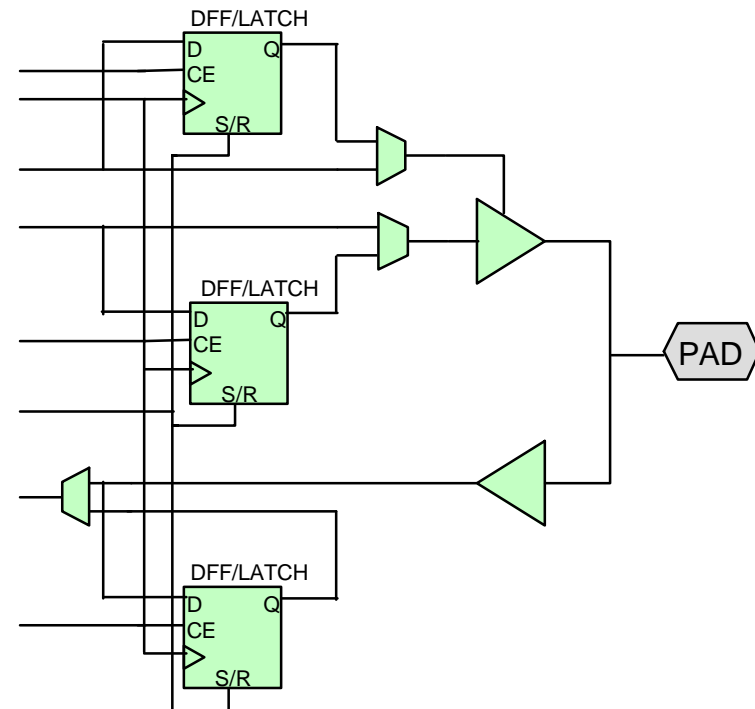
4 to 12 Independent DLLs



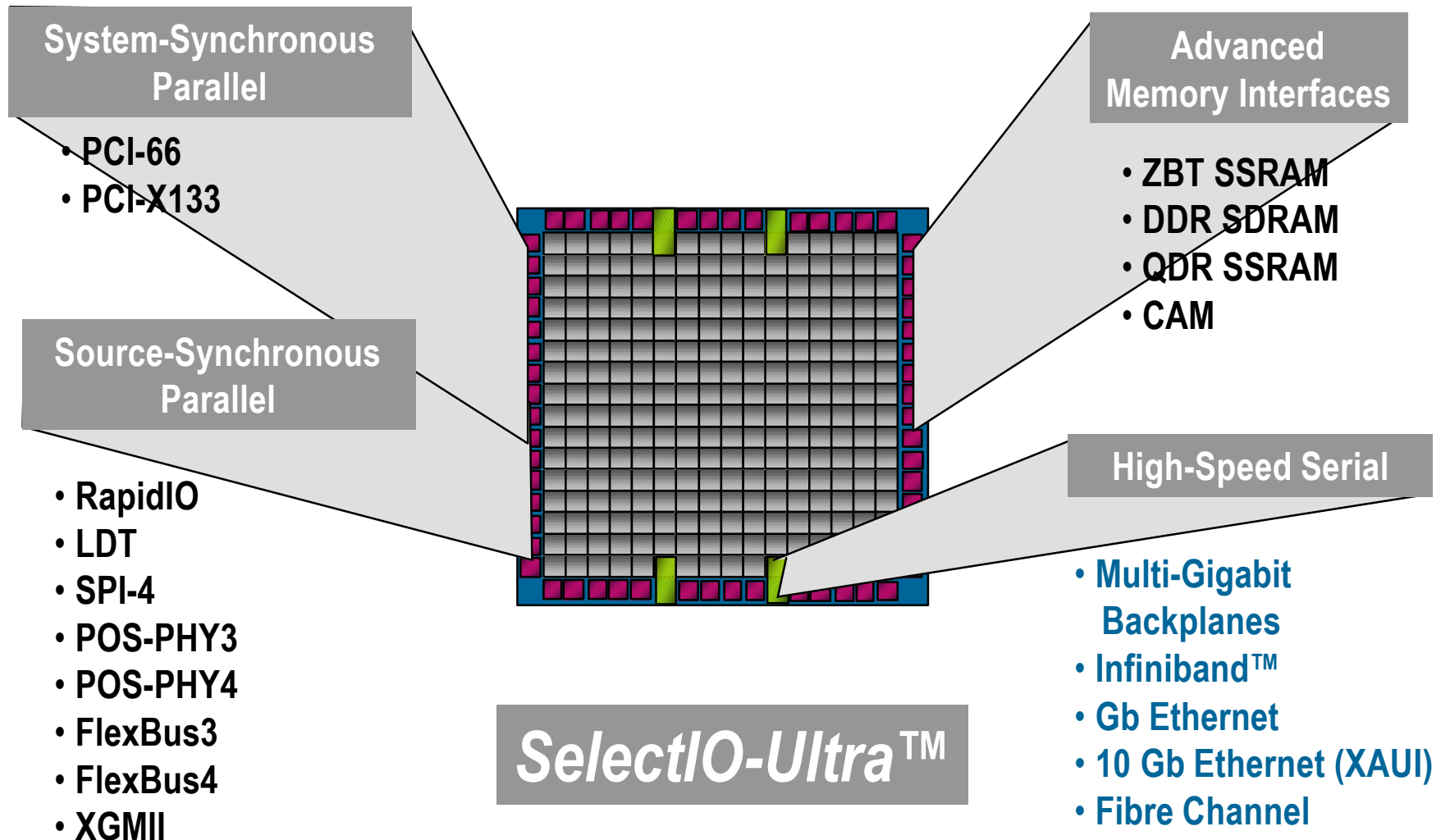
- ◆ **DLLs adjust clock delay to align internal and external clocks**
 - *Digital closed-loop control*
 - *25 to 400-MHz range, 35-picosecond resolution*

Simplified IOB Structure

- ◆ **Fast I/O drivers**
- ◆ **Separate registers for input, output, 3-state control**
 - *Async/Sync set or reset*
 - *Common clock and separate clock enables improve usability*
 - *Configure as FF or latch*
- ◆ **Programmable slew rate and adjustable input delay**
- ◆ **Selectable I/O standards**
 - *Output drive, input threshold*



Virtex-II SystemIO™ Technologies



Multi-Standard I/O

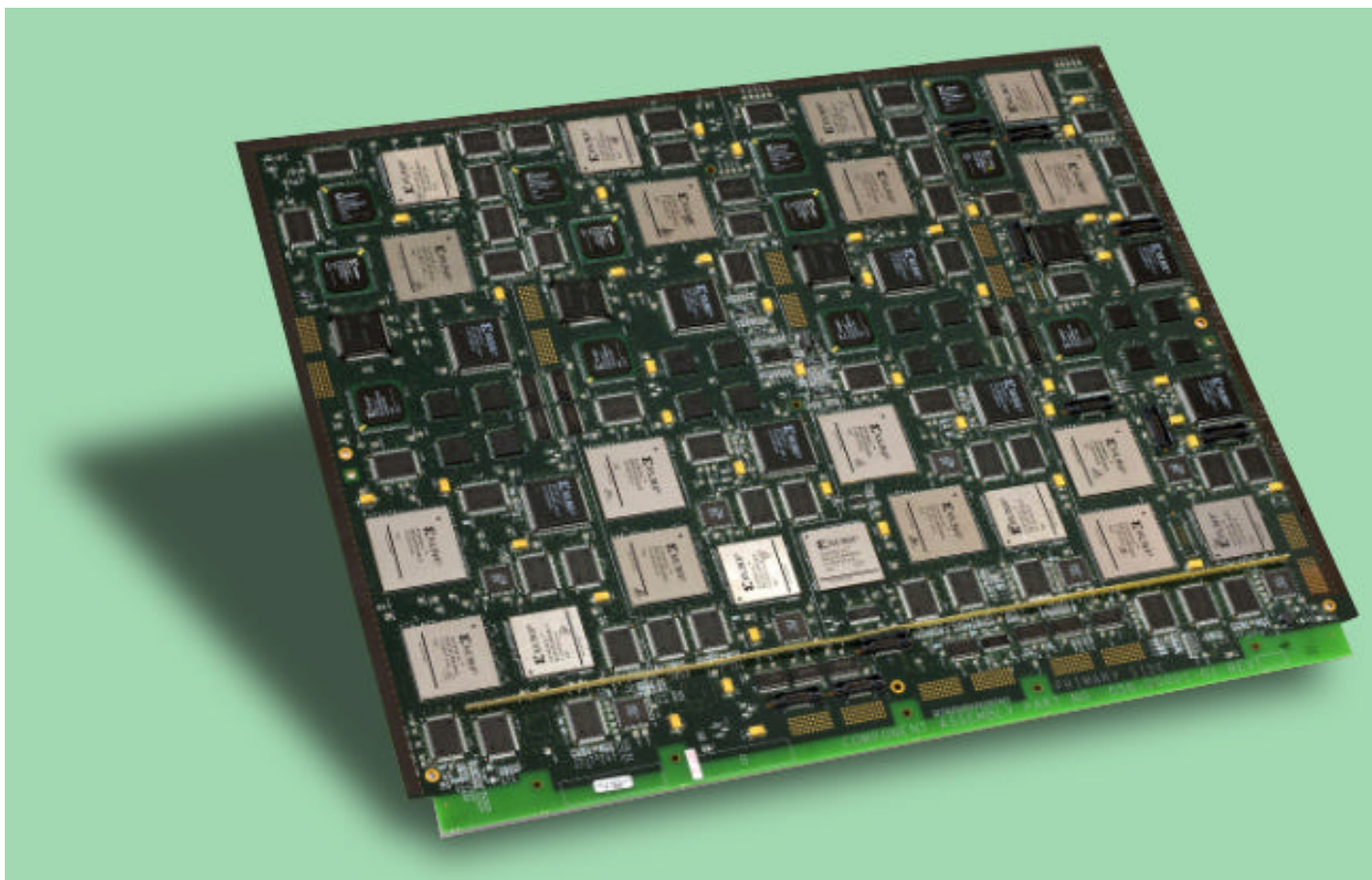
- ◆ **LV-TTL and LV-CMOS**
 - *for logic interfaces*
- ◆ **SSTL and HSTL (3.3, 2.5, 1.5 V)**
 - *for driving terminated lines*
- ◆ **GTL and GTL+**
 - *for driving double-terminated busses*
- ◆ **LVDS and LVPECL**
 - *high-speed differential signals*
- ◆ **Double-Data Rate interfaces**
 - *for ultra-fast data transfer*

Multi-Standard I/O

- ◆ **Essential for system-level FPGAs**
 - *directly interfacing to many different circuits*
- ◆ **Essential for fast interconnects**
 - *requiring different features and trade-offs*
- ◆ **Essential for driving terminated lines**
 - *Demanded by the fast transition times*
- ◆ **On-chip termination simplifies pc-boards**
 - *Eliminates need for external resistor packs*

Optimized interface to any type of logic

Typical 20-Layer PCB: A Very Tough Design Problem



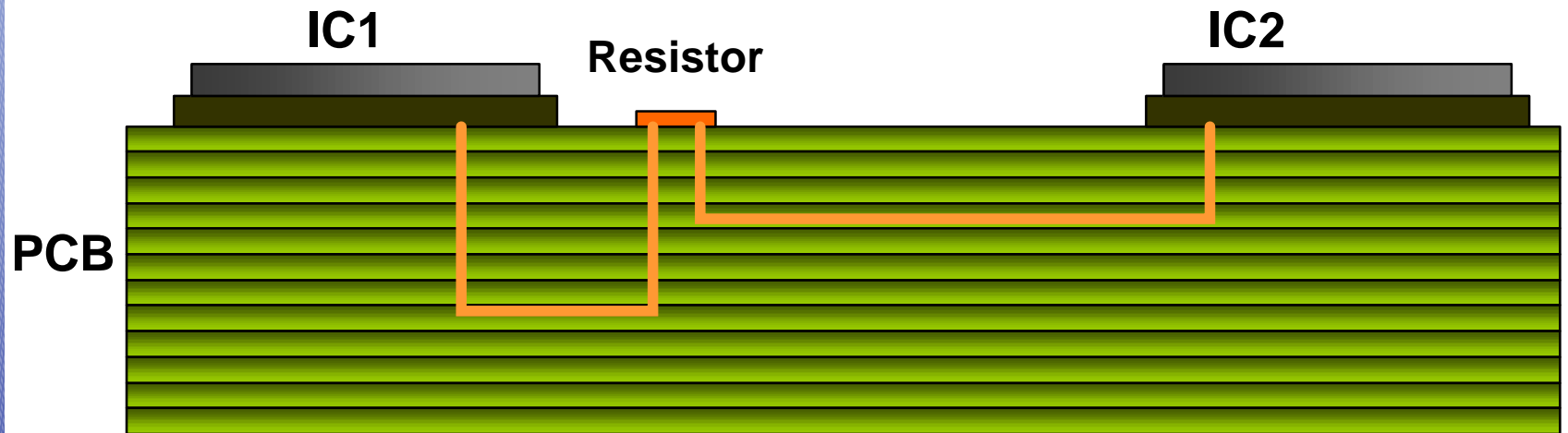
Courtesy, NetCore Systems.

10 March 2001 26

© 2000, 2001 Xilinx, Inc. All Rights Reserved

XILINX®

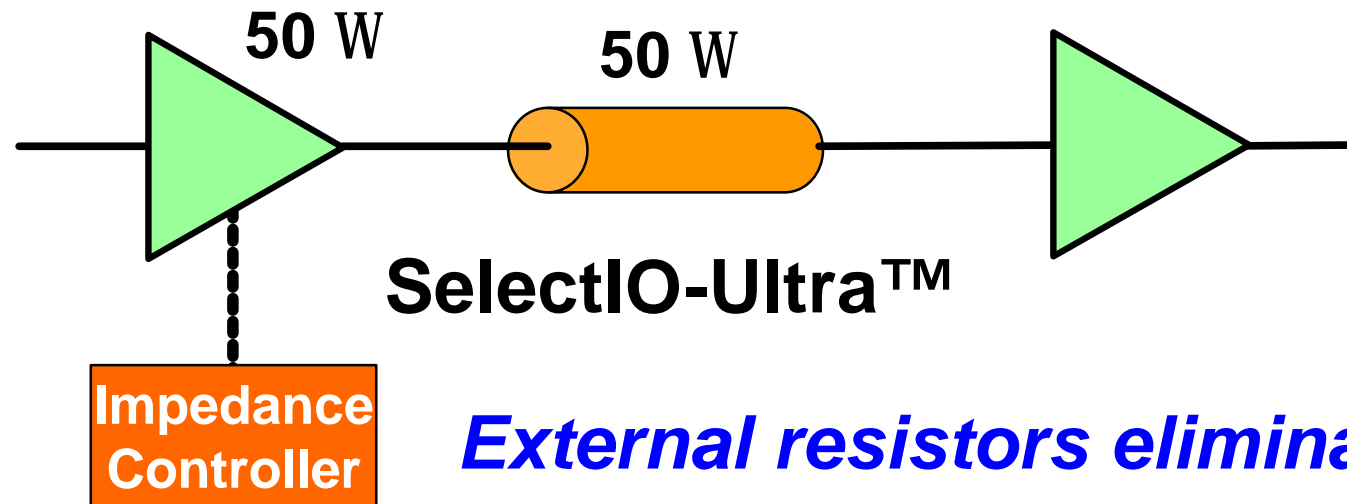
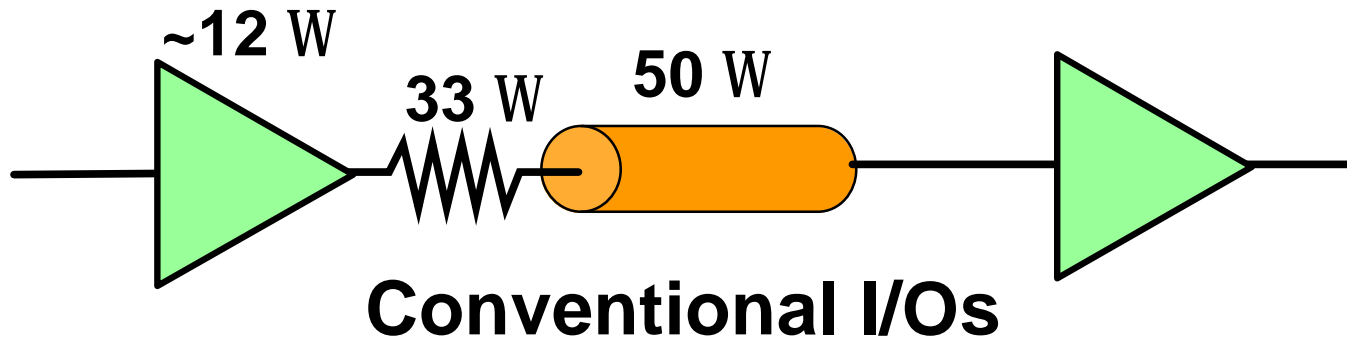
PC-Board Routing Impact



**Multiply this by 1000 pins per chip,
and by the N chips per board!**

8+ weeks for pc-board layout

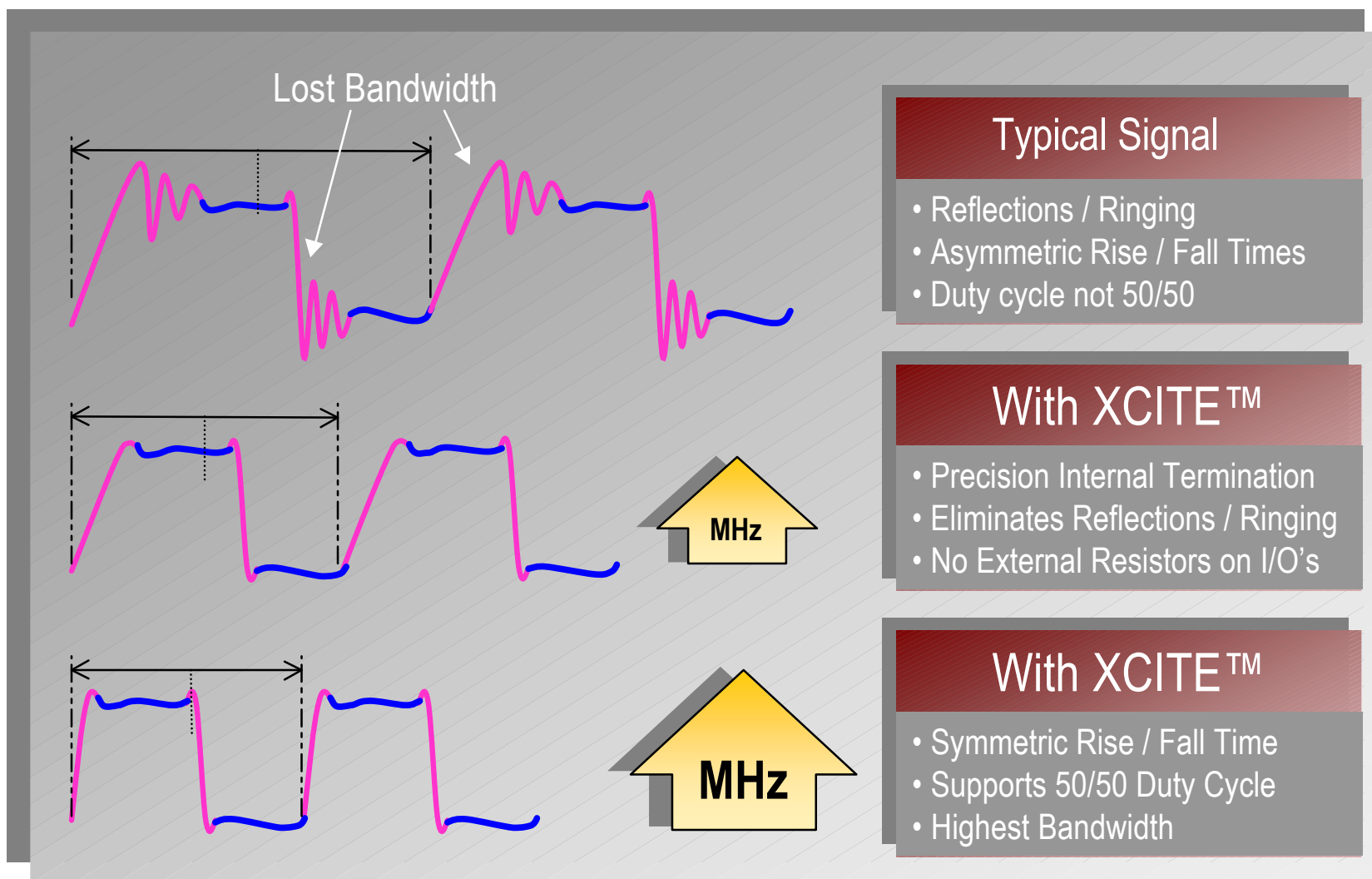
SelectIO-Ultra™



*External resistors eliminated
Impedance maintained by FPGA*

XCITE™

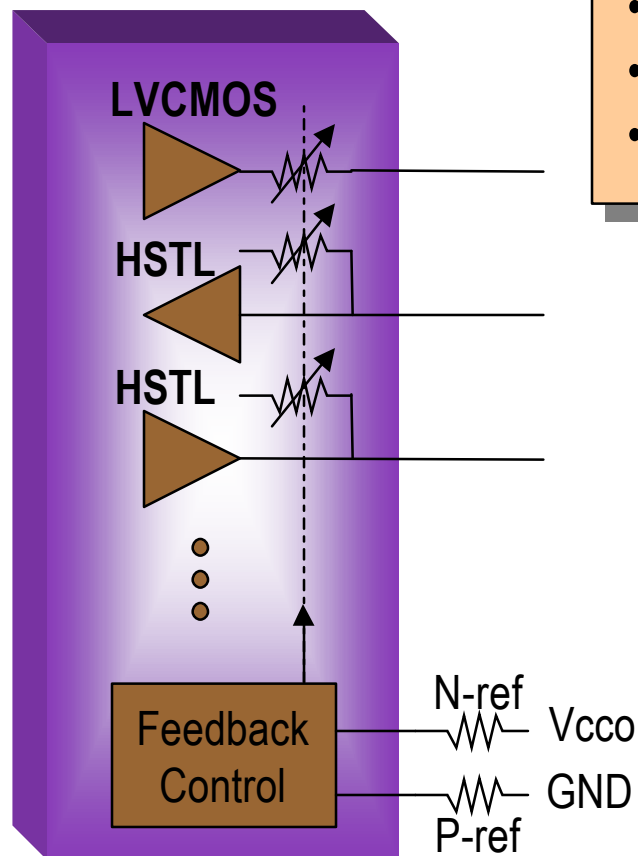
The Evolution of Signal Integrity



XCITE™

Xilinx Controlled Impedance Technology World's 1st Digitally Controlled Impedance Technology

XCITE™ I/O Bank



Eliminate Reflections, Ringing

- Precision internal termination
- 10% tolerance with 1% ref resistors
- Compensated over Voltage, Temp

Eliminate External Resistors

- Built-in impedance control for I & O
- Simplify PCB layout
- Improve system reliability
- Reduce component count

Symmetric Rise/Fall Times

- Separate ref resistors for rise/fall
- Supports 50%/50% clock outputs

Controlled-Impedance Benefits

- ◆ **Better signal integrity,**
 - *higher systems reliability*
- ◆ **Smaller PC-boards,**
 - *easier to layout,*
 - *easier to manufacture*

***XCITE I/O is the only
practical way to interconnect
high pin-count
fine-pitch ball-grid packages***

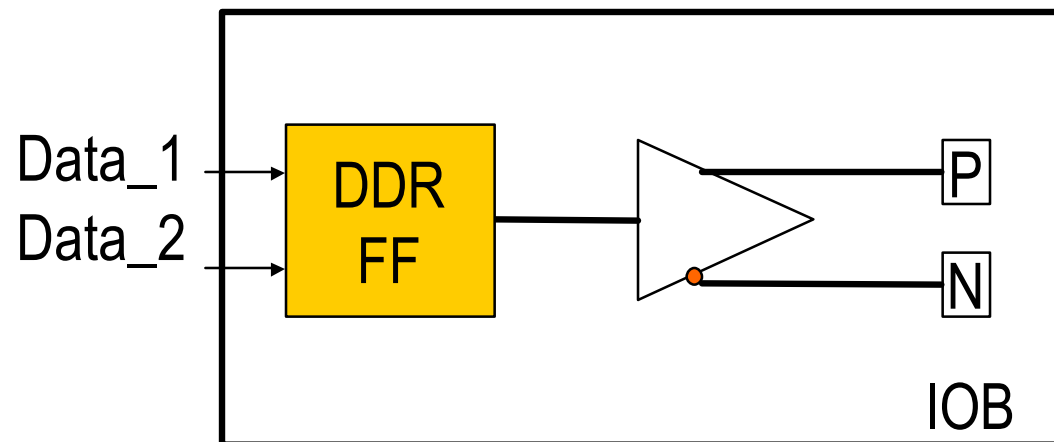
Differential Signaling: 840Mbps

- ◆ **Full LVDS Programmable Solution:**

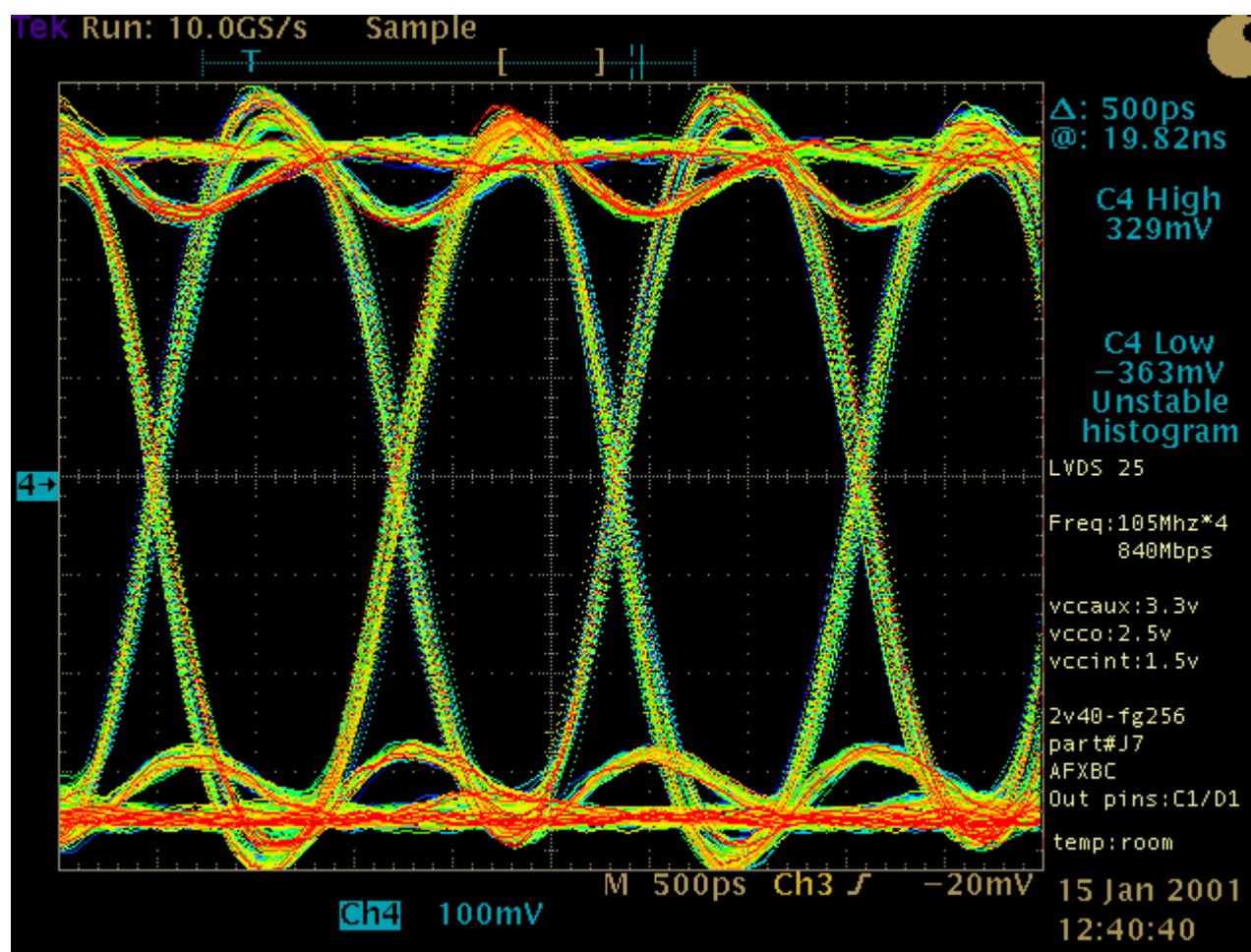
- 2.5 V : 250 mV - 400 mV
- 3.3 V : 250 mV - 400 mV
- Ext. 2.5 V : 350 mV - 750 mV
- Ext. 3.3 V : 350 mV - 750 mV

} Design
Flexibility

- ◆ Integrated current driver

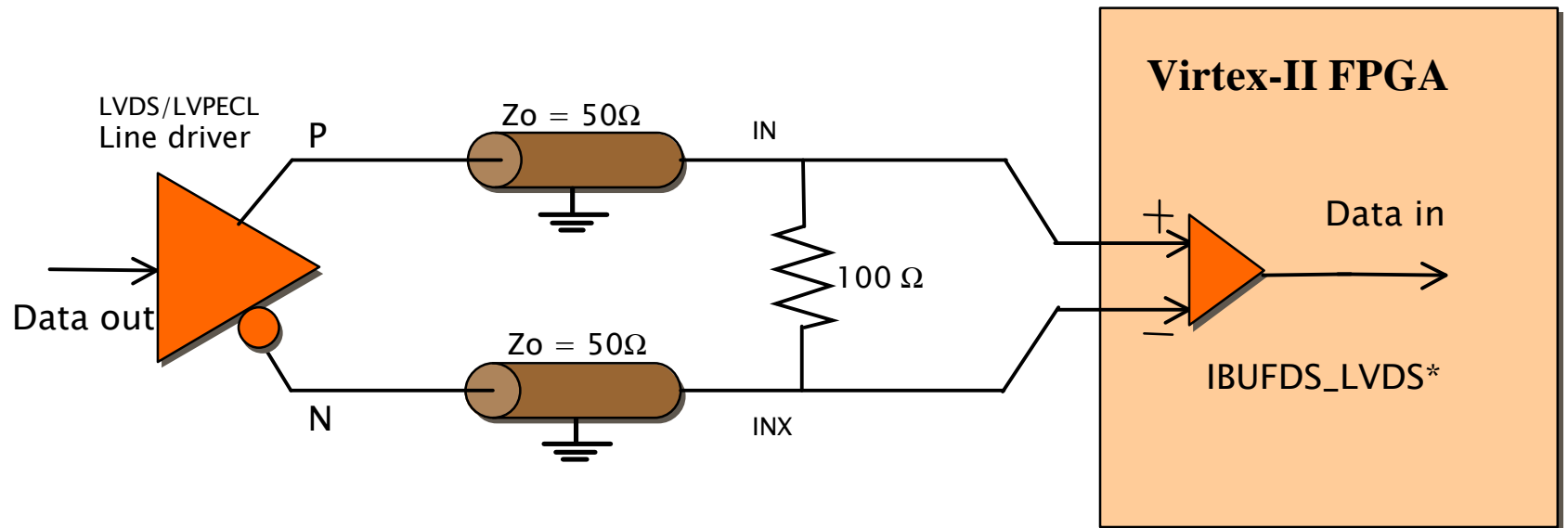


840 Mbps Eye Pattern from Virtex-II LVDS



LVDS Termination

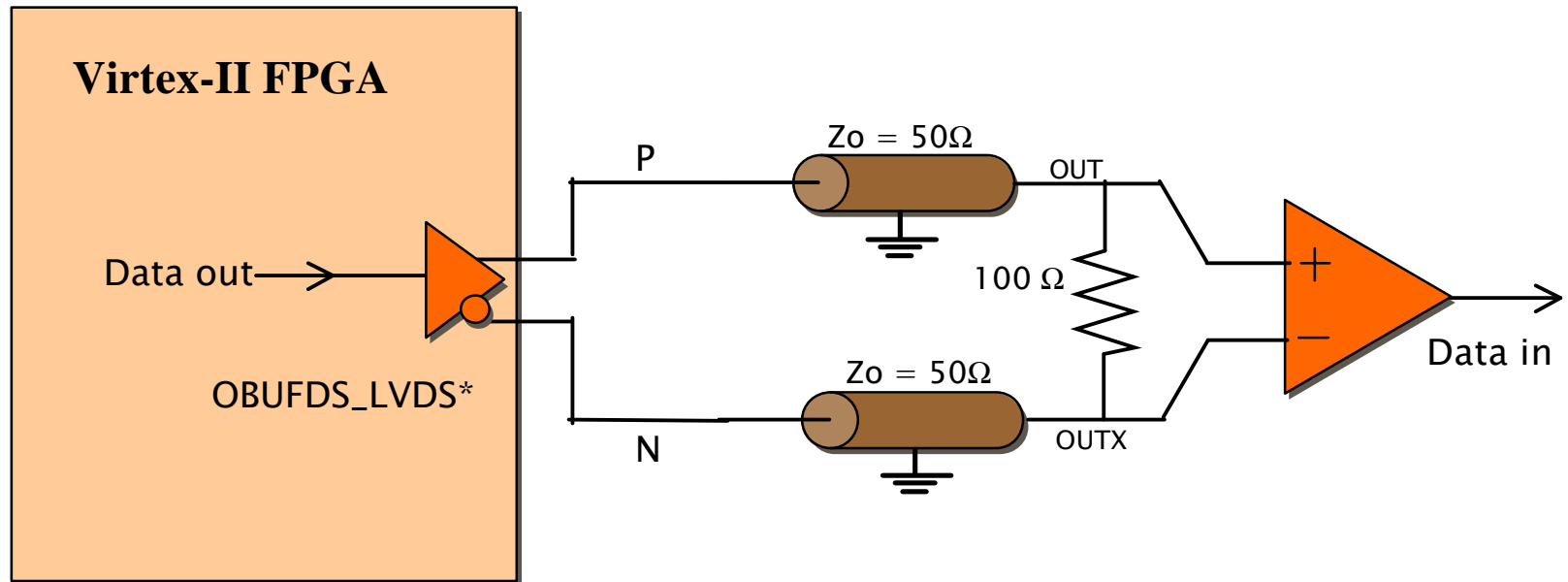
LVDS Receiver: Point-to-point configuration



- ◆ All LVDS receivers require standard termination.

LVDS Termination

LVDS Transmitter: Point-to-point configuration



- ♦ True current-mode driver eliminates the need for external source-termination.
- ♦ The **OBUFDS_LVDS** primitive= True current-mode driver



Designing for High Performance

Performance Parameters I

◆ Parameter Virtex-II-5 (ns)

— **CLB**

— Combinatorial LUT delay	0.41
— Set-up time through LUT	0.65
— Carry delay per bit	0.045
— Clock-to-Q delay	0.40

— **BlockRAM:**

— set-up time (A,D)	0.30
— Clock-to-out	2.89

— **Input**

— Data pin to clock pin set-up	0.78
— Data in delay	0.70

— **Output**

— Data to output pad	2.45
— Clock-to-output pad	3.45

Performance Parameters II

Internal register-to-register

16-bit adder	317 MHz
18 x 18 multiplier	155 MHz
24-bit synchronous counter	305 MHz
64-bit synchronous counter	190 MHz
DLL max output frequency	420 MHz

Virtex-II-5

Package-pin to package pin delays

64-bit decode,	6.8 ns
32 : 1 multiplexer	7.8 ns
One-LUT combinatorial function	4.5 ns

Virtex-II parameters are preliminary and conservative

Designing for High Speed

- ◆ **Understand the architecture, strength and limitations**
 - *LUTs,, LUT-RAMs, SRL16, Carry*
 - *Registered I/O, Output 3-state control flip-flop*
 - *Longlines, 3-state buffers,*
 - *Synchronous dual-ported BlockRAM*
 - *Global clocks with glitch-free enable and input multiplexer*
 - *DLLs, Digital Frequency Synthesizer, Phase control*
 - *Constant-coefficient multipliers in LUTs*
 - *18x18 multipliers in Virtex-II,*

The synthesizer cannot do all your homework

Provide High-Level Floorplanning

- ◆ ***Intelligent pin assignment***
 - prevents routing congestion and poor performance
- ◆ ***Natural structure:***
 - Data flows horizontally, Control flows vertically
 - Vertical adders and counters, carry going upwards
- ◆ ***Pick the best I/O standard, observe banking rules***

Place & route tool should not do all your homework

Design Synchronously, Use Global Clocks

- ◆ **Up to 16 Global Clocks are available**
 - *Very low skew on these clock nets*
- ◆ **DLL eliminates clock distribution delay**
 - *Inside the chip, or even on the pc-board*
- ◆ **Do not gate the clock, use CE instead**
 - *But you may need clock gating for lowest power*
 - *Virtex-II has glitch-free clock gate and clock mux*
- ◆ **Use Carry for adders, counters and comparators**
 - *Superior speed, less logic, forces vertical orientation*
- ◆ **Use predefined cores**
 - *They have been tested and are guaranteed to work at speed*

Use Global Buffers to Reduce Clock Skew

- ◆ **Global buffers are connected to dedicated routing**
 - *Global clock network is balanced to minimize skew*
- ◆ **All Xilinx FPGAs have global buffers**
 - *XC4000 and Spartan have 8*
 - *Virtex and Spartan-II have 4*
 - *Virtex-II has 16 BUFGs with glitch-free input mux*
- ◆ **You can always use a BUFG symbol and the software will choose an appropriate buffer type**
 - *All major synthesis tools can infer global buffers onto clock signals that come from off-chip*

Why Use Timing Constraints?

- ♦ The implementation tools do **NOT** try to find the placement and routing that achieves the fastest speed
 - *they just try to meet your performance expectations*
- ♦ **YOU must communicate your expectations**
 - *through Timing Constraints*
- ♦ **Timing Constraints improve performance**
 - *by placing logic closer together and shortening the routing*

***Timing constraints are the best high-level tool
to achieve guaranteed performance***

More About Timing Constraints

- ◆ **Timing constraints define your performance objectives**
 - *Tight timing constraints increases compile time*
 - *Unrealistic constraints causes the Flow Engine to stop*
 - *Logic Level Timing Report tells whether constraints are realistic*
- ◆ **After implementation,**
 - *review the Post Layout Timing Report to determine if performance objectives were met*
- ◆ **If your constraints were not met,**
 - *use the Timing Analyzer to determine the cause*



Designing for Signal Integrity

Transmission Lines

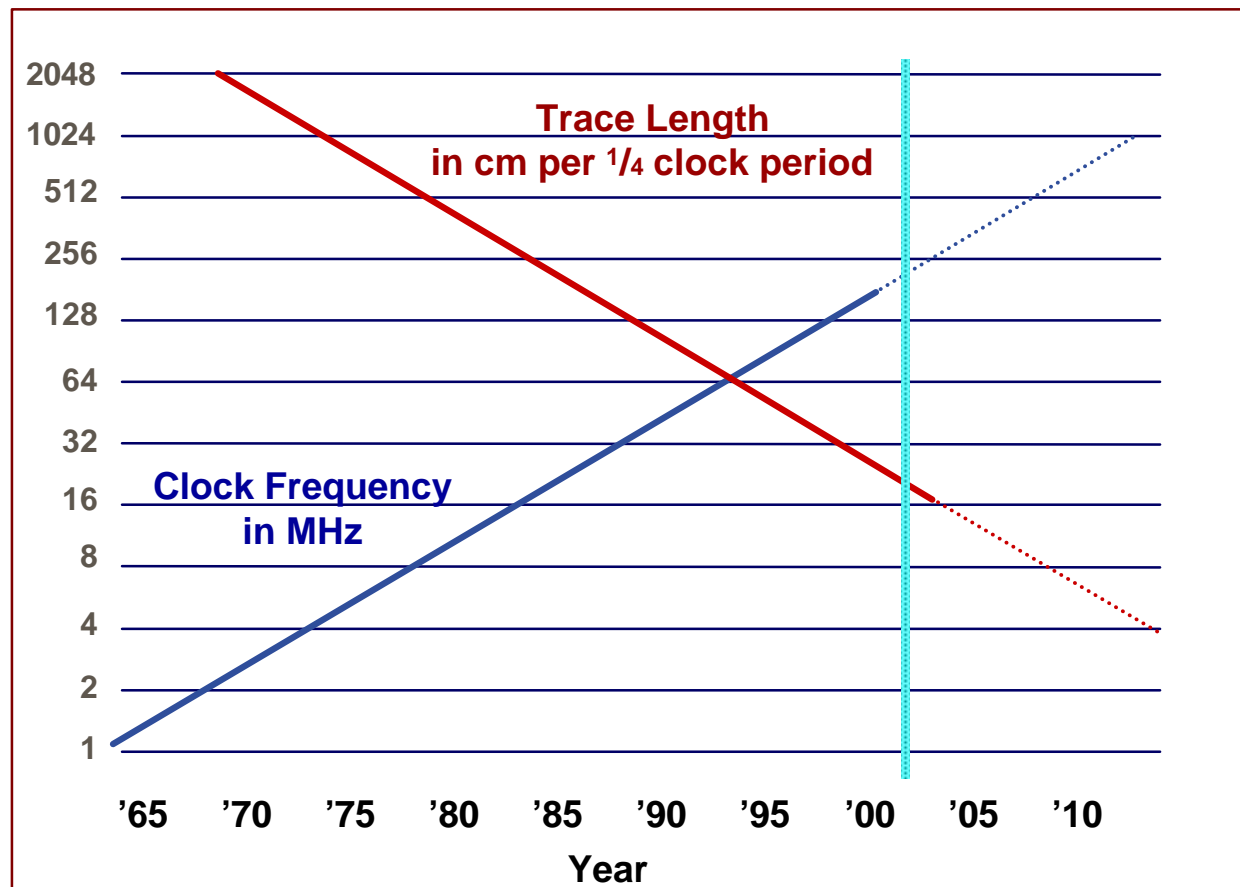
- ◆ Long traces are transmission lines, they can ring
 - *“transmission line” if round trip > transition time*
 - *“lumped-capacitance” if round trip < transition time*
- ◆ Signal delay on a pc-board:
 - *140 to 180 ps per inch (50 to 70 ps per cm)*
- ◆ Avoid reflection by terminating the line
 - *either series termination at the source
or parallel termination at the destination*
- ◆ Longest trace that behaves as a lumped-capacitance:
 - *3 inches max for a 1-ns transition time (7.5 cm)*
 - *6 inches max for a 2-ns transition time (15 cm)*

Evolution

	1965	1980	1995	2010 (?)
Max Clock Rate (MHz)	1	10	100	1000
Min IC Geometry (μ)	-	5	0.5	0.05
Number of IC Metal Layers	1	2	3	10
PC Board Trace Width (μ)	2000	500	100	25
Number of Board Layers	1-2	2-4	4-8	8-16

- ◆ *Every 5 years:
System speed doubles, IC geometry shrinks 50%*
- ◆ *Every 7-8 years: PC-board min trace width shrinks 50%*

Moore Meets Einstein



- ◆ *Speed Doubles Every 5 Years...
...but the speed of light never changes*

Designing for Signal Integrity

- ◆ **Devices need good Vcc bypassing**
 - *Bypass capacitor is the only source of dynamic current*
- ◆ **Output driver needs IBIS models**
 - *http://www.xilinx.com/support/troubleshoot/htm_index/sw_ibis.htm*
- ◆ **User needs understanding of transmission line effects**
 - *Characteristic impedance, reflections, dV/dt*
 - *series termination, parallel termination,*
- ◆ **Model the pc-board with HyperLynx**
 - *Multi-Layer with undisturbed ground/power planes*
 - *Controlled-impedance signal lines (50 to 75 Ohms)*
- ◆ **Website:**
 - *<http://www.xilinx.com/support/techxclusives/CircuitBoard-techX6.htm>*

Signal Integrity Tools

- ◆ **IBIS models**

- http://www.xilinx.com/support/troubleshoot/htm_index/sw_ibis.htm

- ◆ **HyperLynx**

- ◆ **Fast oscilloscope and fast probes**

- *Beware of slow scopes measuring **1 ns** rise time:*

- *A **1 GHz** scope with a 1 GHz probe displays **1.2 ns** rise time*

- *A **250 MHz** scope and probe displays: **3.0 ns** rise time*

- ◆ **Measure eye patterns**

- *Use LFSR to generate pseudo-random sequence*

- ◆ **Spectrum analyzer**

- *Measure the effect of decoupling capacitors, etc.*

- ◆ **Website:**

- <http://www.xilinx.com/support/techxclusives/signals-techX5.htm>

Power Supply Decoupling

- ◆ **CMOS current is dynamic**
 - *I_{cc} current spike on every active clock edge*
- ◆ **Peak current can be 5x the average current**
 - *Instantaneous current peaks can only be supplied by decoupling capacitors*
- ◆ **Use one 0.1 uF ceramic chip capacitor per Vcc pin**
 - *Low L and R are more important than high C*
 - *Double up for lower L and R if necessary*
 - *Use direct vias to the supply planes, extremely close to the power-supply pins*
 - *On-chip plus package capacitance is ~0.01μF*

Tricks of the Trade

- ◆ **Reduce the output strength**
 - *LVTTL and LVCMOS offer 2, 4, 6, 8, 12, 16, and 24 mA*
- ◆ **Use SLOW attribute where available**
 - *Increases transition time*
 - *especially when driving transmission lines*
- ◆ **Explore different I/O standards**
 - *Different supply voltages, input thresholds*
 - *Unidirectional, bidirectional, bus-oriented, differential*
- ◆ **Reduce fan-out and load capacitance**
- ◆ **Add virtual ground to alleviate SSO problems**
 - *Ground output pin inside and outside, give it max strength*

Testing for Performance and Reliability

- ◆ **Manipulate circuit speed for testing purposes:**
 - *Hot and low V_{cc} = slow operation*
 - *Cold and high V_{cc} = fast operation*
- ◆ **If it fails hot: insufficient speed**
 - *Use a faster speed grade*
 - *Modify the design, add pipelining*
- ◆ **If it fails cold: signal integrity and hold time issues**
 - *Look for clock reflections*
 - *Look for excessive internal clock delays*
 - *Look for decoding spikes driving clocks*
 - *Look for “dirty asynchronous tricks”*

Model and Measure

- ◆ **Model device, package, pc-board**
 - *Avoids pc-board re-spin*
- ◆ **Measure performance and noise margin**
 - *Avoids field disasters*
- ◆ **Do not panic:**
 - *It's only 1 and 0, High and Low that count*
 - *Noise immunity takes care of the rest*
- ◆ **References:**
 - *Classes: see www.hyperlynx.com, then go to TRAINING*
 - *Book: Johnson & Graham High-Speed Digital Design*
- ◆ **Website:**
 - *www.xilinx.com/support/techxclusives/techX-home.htm*



Designing with BlockROMs

Designing with BlockRAMs

- ◆ **Dual-ported synchronous BlockRAMs**
 - *Synchronous read and write*
- ◆ **Two Ports share nothing but the common data array**
 - *Individual address, data, clock, read/write, CE*
- ◆ **Each port can be configured individually**
 - *Parallel-serial (or S-P) converter “for free”*
- ◆ **4K bits per BlockRAM in Virtex**
 - *4K, 2K, 1K or 512 deep (256 x 16 with ports combined)*
- ◆ **18K bits per BlockRAM in Virtex-II**
 - *16K, 8K, 4K, 2K, 1K or 512 deep (256 x 72 combined)*
- ◆ **Max 180 BlockRAMs in Virtex, max 192 in Virtex-II**

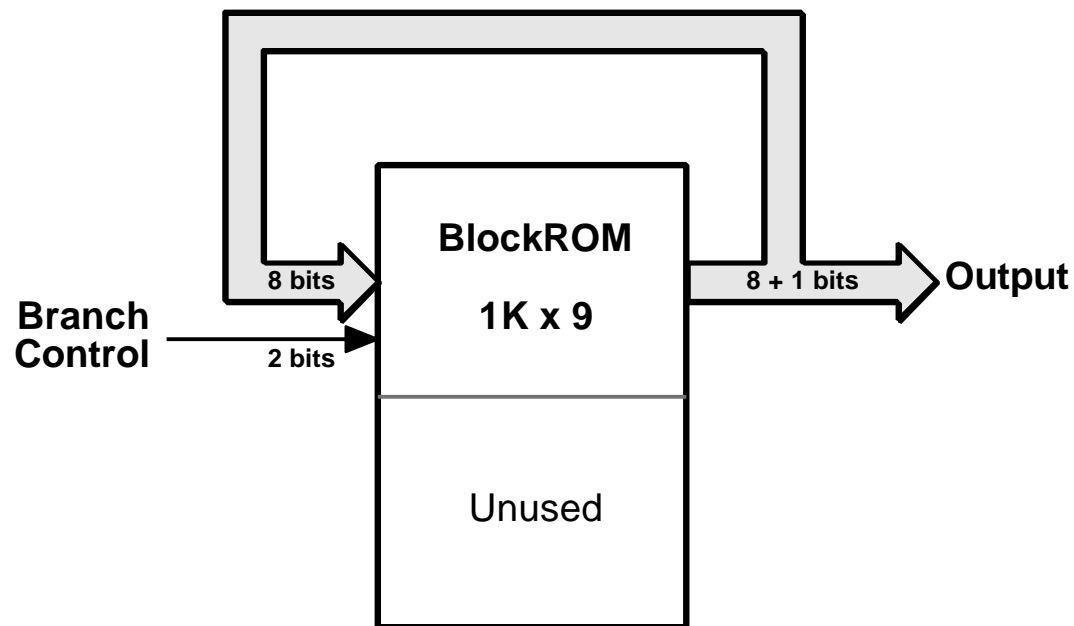
BlockROM State Machines

- ◆ **BlockRAM can be initialized as BlockROM**
- ◆ **Virtex has 4086 bits per BlockROM**
- ◆ **Counters**
 - *Two 8-bit Gray counters*
 - with additional binary outputs
 - *Two 1-digit decimal counters*
 - with 7-segment read-out
 - *16-bit up/down binary counter*
 - *4-digit BCD up/down counter*
- ◆ **Finite State Machines**
 - *Two 4-input 32-state state machines*

BlockROM State Machines

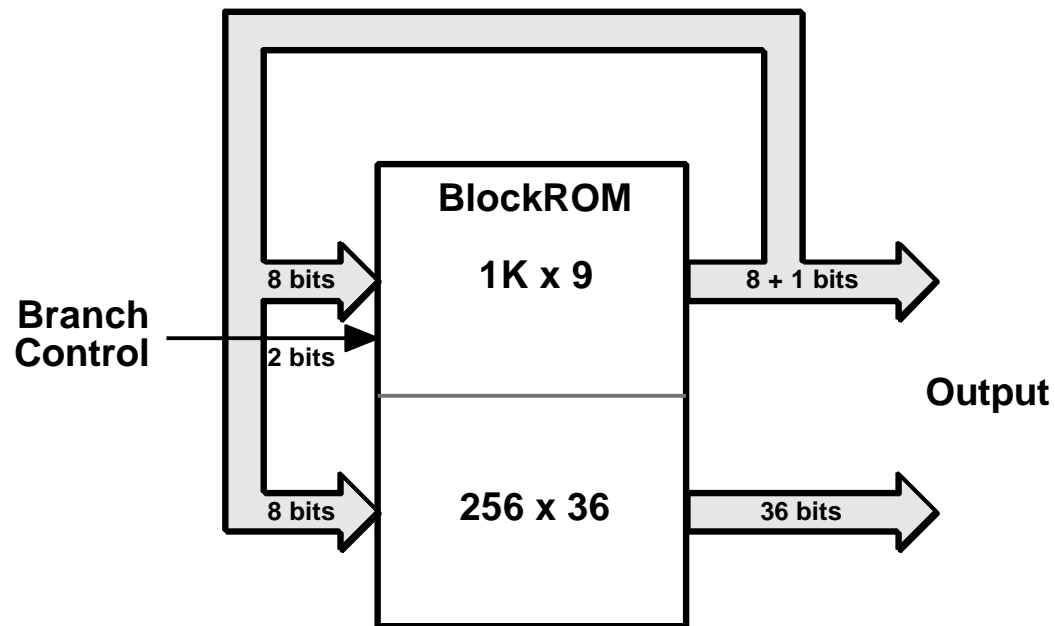
- ◆ **Virtex-II has 18K bits per BlockROM**
- ◆ **Counters**
 - *One 20-bit Gray counter*
 - *One 6-digit decimal counter*
 - using one additional CLB
 - *One 20-bit binary counter*
- ◆ **Finite State Machines (FSMs)**
 - *Two 5-input 64-state state machines*
 - *Two 4-input 128-state state machines*

Fast FSM in 1/2 BlockROM...



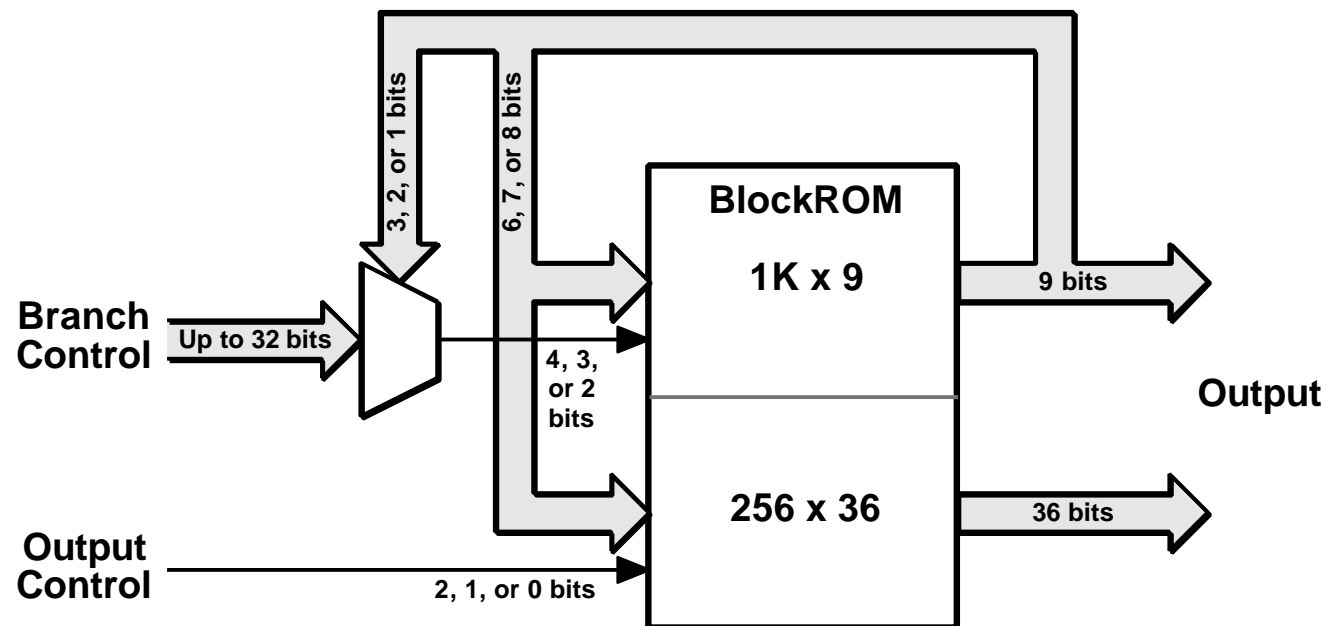
- ◆ **256 states, 4-way branch, 150 MHz operation**
 - or 128 states, 8-way branch, same speed
 - or 64 states, 16-way branch, same speed

...plus 36 Additional Outputs



- ◆ **36 additional parallel outputs**
— *from the other half of the BlockRAM*

...and Many Control Inputs



64, 128, or 256 states with multi-branch capability
36 freely assigned + 8 encoded outputs
optional multiplexed control inputs

All in one BlockRAM plus two CLBs

BlockROM Code Converters

- ◆ Virtex has 4096 Bits per BlockROM
- ◆ Simultaneous Sine and Cosine table
- ◆ Two 9-bit binary to 3-digit BCD converters
 - *Binary and BCD have identical LSB*
- ◆ Two telecom 8-bit μ -law or A-law to linear converters
- ◆ 3-digit BCD to 9-bit binary converter
- ◆ Wallace-tree adder
 - *22, 44, or 48 inputs in several BlockROMs*

BlockROM Code Converters

- ◆ **Virtex-II has 18K bits per BlockROM**
- ◆ **High-resolution simultaneous Sine and Cosine table**
- ◆ **Two 11-bit binary to 4-digit BCD converters**
 - *Binary and BCD have identical LSB*
- ◆ **Two telecom 8-bit μ -law / A-law to linear converters**
- ◆ **Two 3-digit BCD to 10-bit binary converter**
 - *BCD and binary have identical LSB*
- ◆ **Wide-input Wallace-tree adder in multiple BlockROMs**



Designing for Low Power

Designing for Low Power Consumption

- ◆ To extend battery life
- ◆ To reduce chip temperature and cooling requirements
 - $T_{jmax} = 125 \text{ degr.C}$ (150 degree.C in ceramic)
 - Delays increase 0.35% / degr.C
above the guaranteed 85 degr.C junction temperature
- ◆ Use the free Xilinx Power Estimator
 - <http://www.xilinx.com/cgi-bin/powerweb.pl>

Power is proportional to CV^2f
Minimize all three !

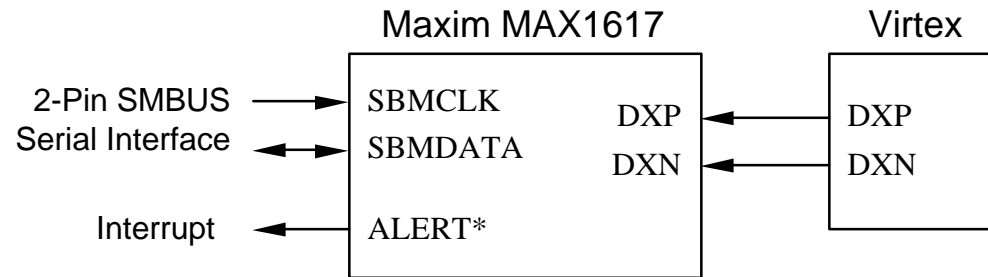
Designing for Low Power

- ◆ **Clock Power + I/O Power + Logic Power**
- ◆ **Clock Power**
 - Minimize # of high-speed clock nets
 - Use DLLs for phase-aligned sub-clocks
 - CE does not reduce clock power
- ◆ **I/O power**
 - Avoid wasted current in input buffers
 - Use fast, full-swing input signals
 - Use output registers to avoid output glitches

Low Logic Power

- ◆ **Control Vcc tightly**
 - *Power is proportional to V_{cc}^2*
- ◆ **Minimize logic transitions and glitches**
- ◆ **Optimize counters:**
 - *Gray and Johnson are best*
 - *Binary counters double the power*
 - *Linear Feedback Shift Register are even worse*
- ◆ **Minimize internal node capacitance**
 - *Use aggressive timespecs*
 - *Design for the highest speed possible, **even if not needed***
 - *This assures lowest interconnect capacitance and provides the lowest power at the lower clock frequency*

Thermal Solution



◆ Remote Die Sensor

- *Specially designed to be used with the maxim MAX1617*
- *Simple 2-pin interface with no calibration required*
- *Provides two channels*
 - *FPGA die temp reported from -40 to +125 degr.C (+/- 3 degr.C)*
- *Programmable over-temperature & under-temp. alarms*
- *Originally intended for the Pentium II*

Precise thermal management is now easy



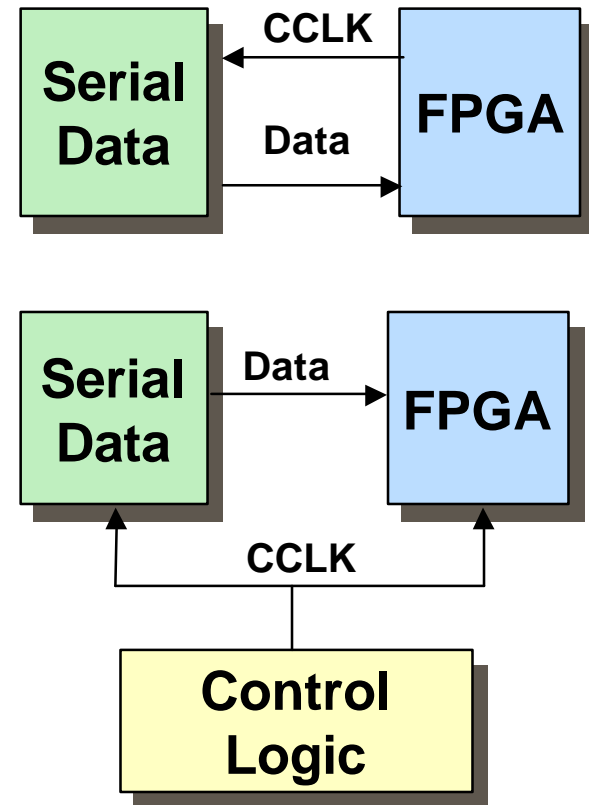
Designing for Security

Designing for Security

- ◆ **Configuration bitstream can be intercepted**
 - *But not interpreted or reverse-engineered*
 - *Some users are concerned about IP theft*
- ◆ **Virtex -II offers security through encryption**
 - *Triple-DES with 3 x 56 bits*
 - *Triple-DES has never been cracked*

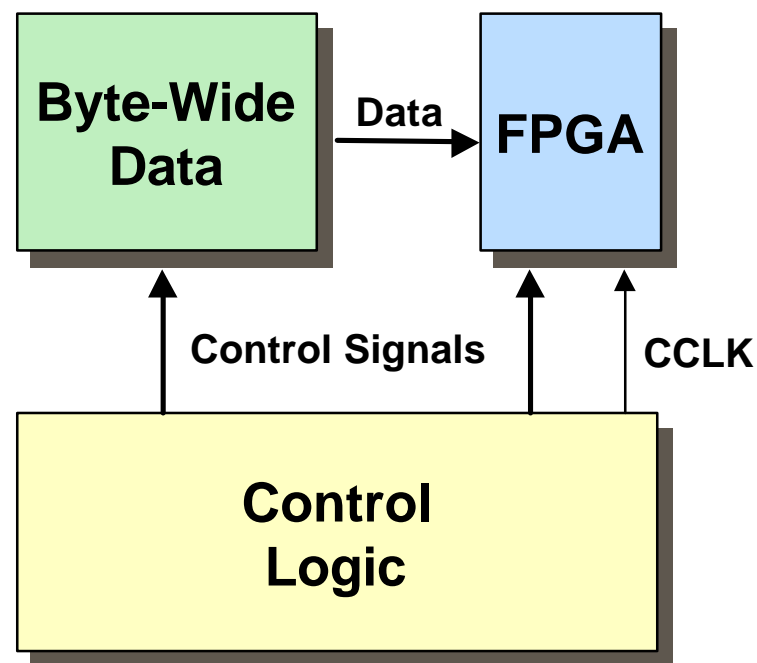
Configuration Modes: Serial Modes

- ◆ **Data is loaded one bit per CCLK**
- ◆ **Master serial**
 - *FPGA drives configuration clock (CCLK)*
 - *FPGA provides all control logic*
 - *Note that CCLK is also an input !*
- ◆ **Slave serial**
 - *External control logic generates CCLK*
 - Microprocessor
 - Xilinx download cable
 - Another FPGA



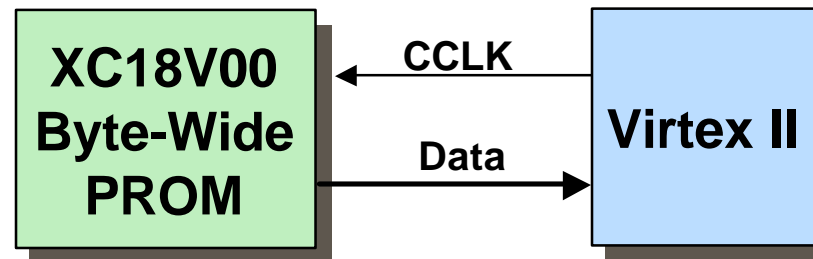
Configuration Modes: Byte-Wide SelectMAP Mode

- ◆ **Slave SelectMAP**
 - *CCLK is driven by external logic*
 - *Data is loaded one byte per CCLK*



Configuration Modes: Master SelectMAP Mode

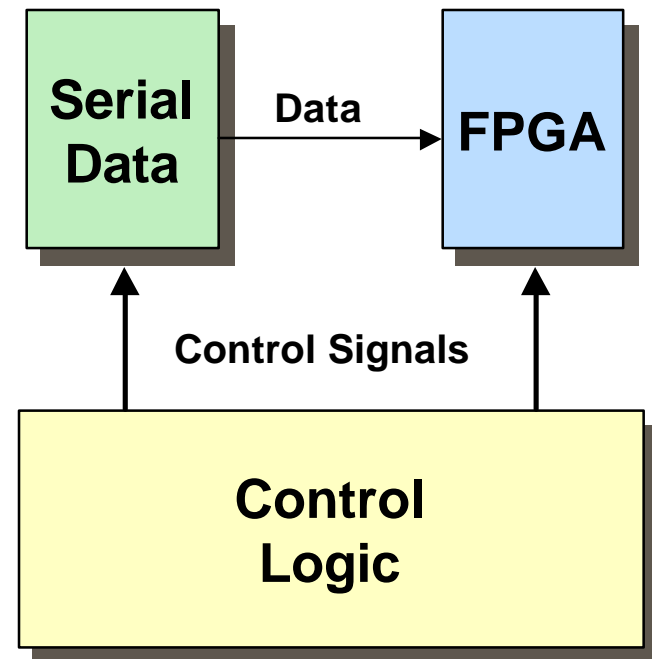
- ◆ **Master SelectMAP**
 - *CCLK is driven by the Virtex II FPGA*
 - *Data is loaded one byte per CCLK*



*New to Virtex II
by popular demand...*

Configuration Modes: Boundary Scan Mode

- ◆ External control logic required
- ◆ Control and data drive the boundary scan pins (TDI, TMS, TCK)
- ◆ Data is loaded bit-serially one bit per TCK



DesignSecurity

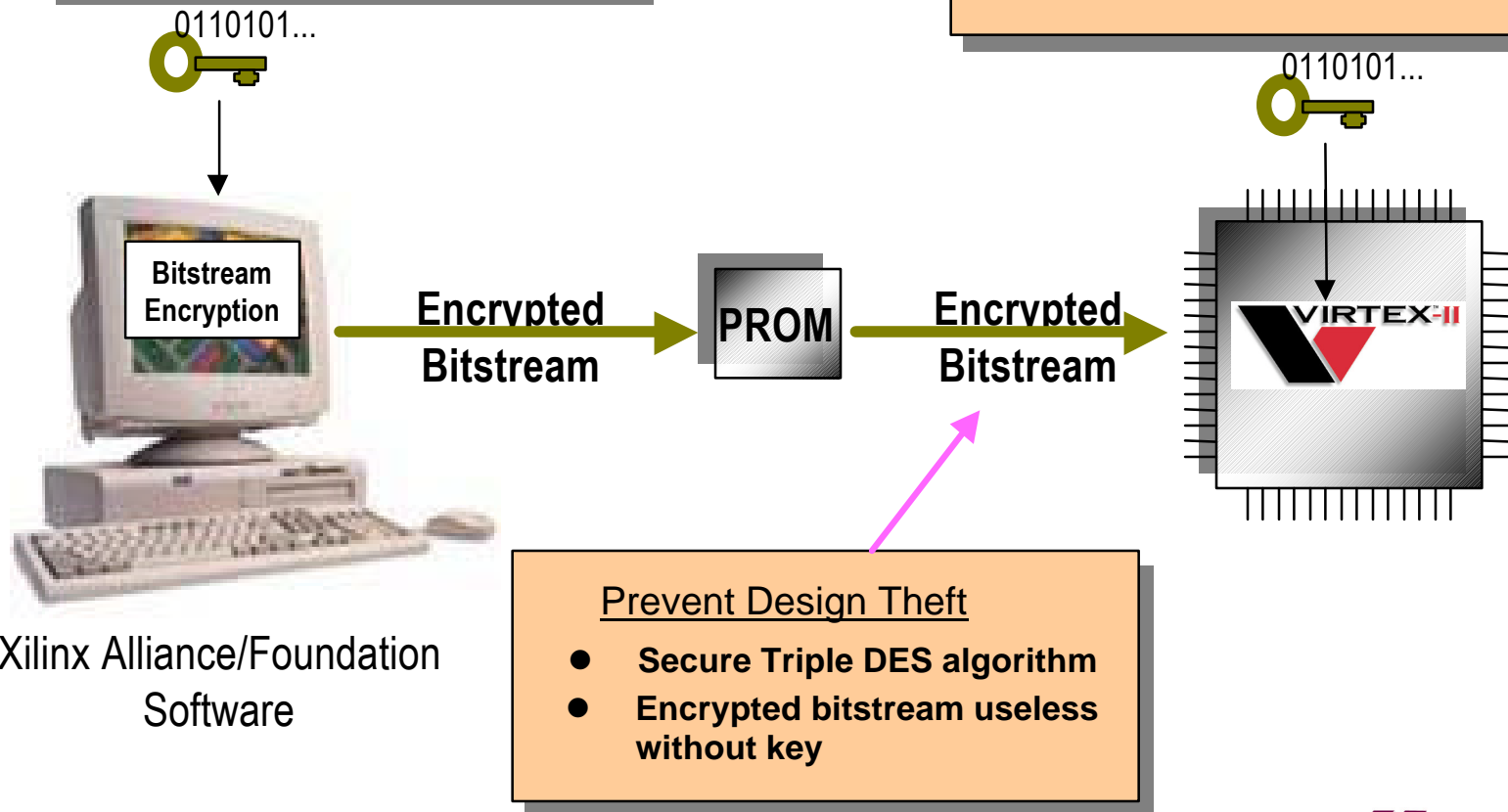
Bitstream Encryption

Easy To Use

- Simple encryption interface
- User-supplied secret key

New Capability

- Secure IRL Updates
- Supports whole-chip IP business model





Asynchronous Issues

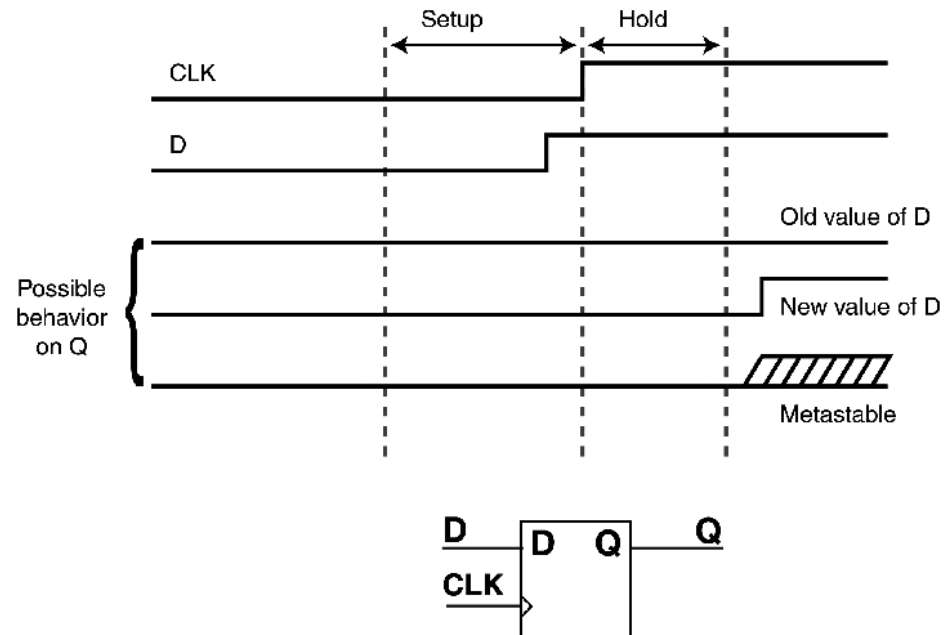
Understanding Asynchronous Design Issues

- ◆ **Most systems operate synchronously inside**
 - *But asynchronous inputs are a fact of life*
- ◆ **Occasionally, an asynchronous input will cause a flip-flop to go metastable**
 - *This is a rare, but unavoidable, probabilistic event*
- ◆ **Solution:**
 - *Faster flip-flops recover faster*
 - *Double-synchronization reduces probability*

Awareness and understanding are crucial

Setup and Hold Time Violations

- ◆ Violations occur when the flip-flop input changes too close to a clock edge
- ◆ Three possible results:
 - Flip-flop clocks in old data value
 - Flip-flop clocks in new data value
 - Flip-flop output becomes **metastable**



Metastability is a rare, random event

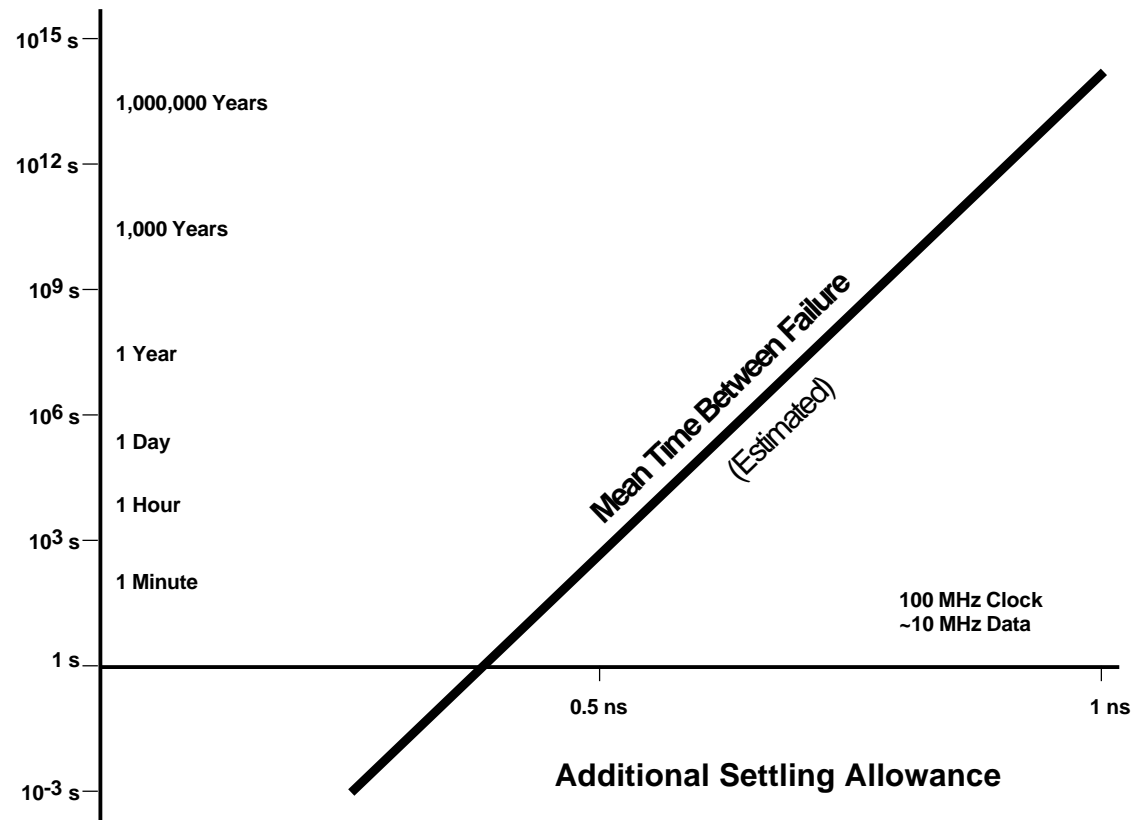
Metastability

- ◆ **Caused by asynchronous data input**
 - *Violates set-up time requirement*
 - *Usually gets synchronized in the flip-flop without problem*
- ◆ **But if data changes within a tiny set-up time window**
 - *Then the flip-flop can go metastable*
 - *Resulting in unpredictable delay to reach stable 1 or 0*
- ◆ **The 0 vs. 1 uncertainty is irrelevant**
 - *The slightest timing change would give a correct 1 or 0*
- ◆ **The unpredictable delay is the problem**
 - *It can violate set-up times in the system, causing erratic operation or even crashes*

Mean Time Between Failure

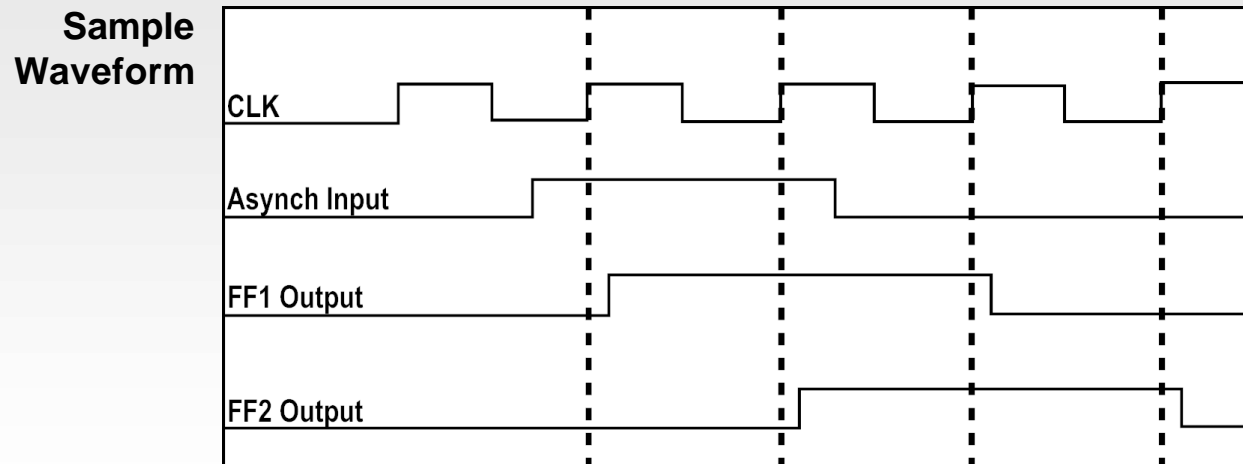
- ◆ **Measure MTBF = f (extra delay)**
 - *Assume a given clock and data rate*
- ◆ **MTBF is exponential function of delta t**
 - *Slope determined by gain-bandwidth product*
- ◆ **Modern CMOS resolves extremely fast**
 - *But modern system have little time slack*
- ◆ **The problem is as unavoidable as death and taxes**
 - *but probability can be reduced by design*

Metastability Data

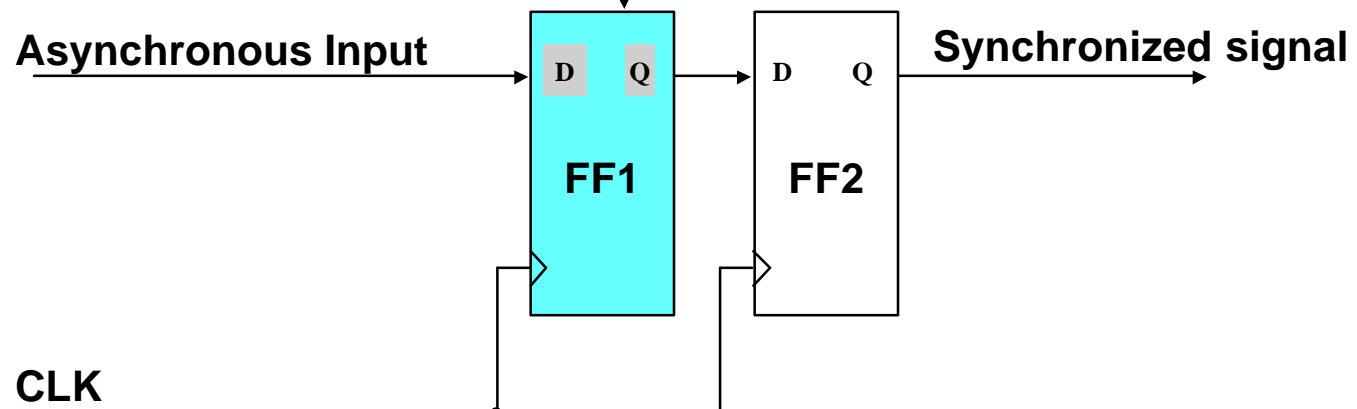


- ◆ Website (will be updated in March 2001):
— <http://www.xilinx.com/xapp/xapp094.pdf>

Synchronization Circuit



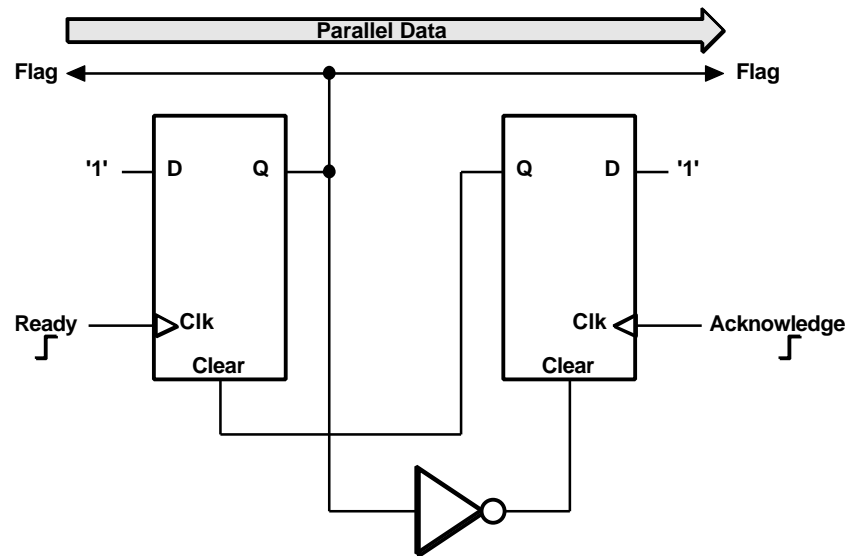
Guards against metastability



Moving Data Across Asynchronous Clock Boundaries

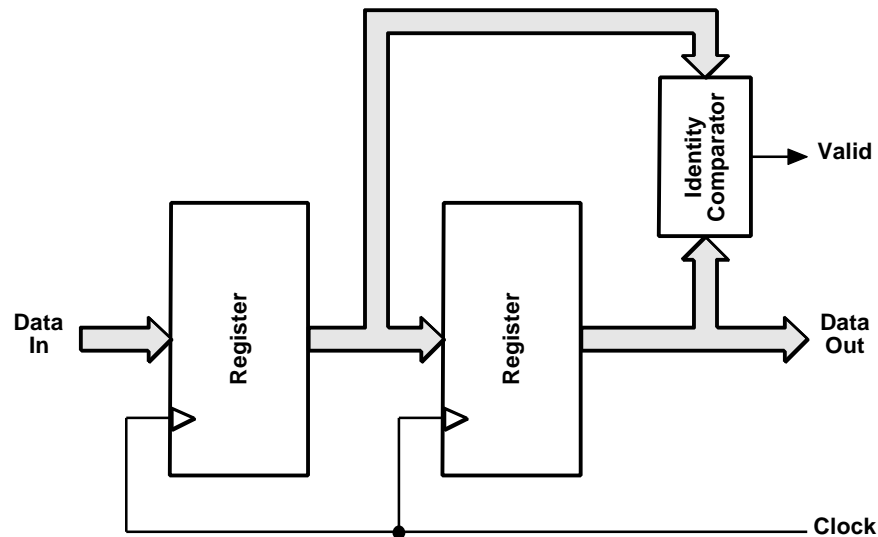
- ◆ Worst-case timing happens, sooner or later
- ◆ *Murphy never sleeps !*
- ◆ Never use parallel flip-flops to synchronize an asynchronous input signal
 - *Always synchronize at a single point*
- ◆ Don't try to synchronize parallel data
 - *Use the methods described on the following slides*
 - *The problem is data corruption, not metastability*
- ◆ Use cascaded stages to combat metastability
- ◆ Website:
 - <http://www.isdmag.com/editorial/2000/design0003.html>

Moving Parallel Data with Asynchronous Handshake



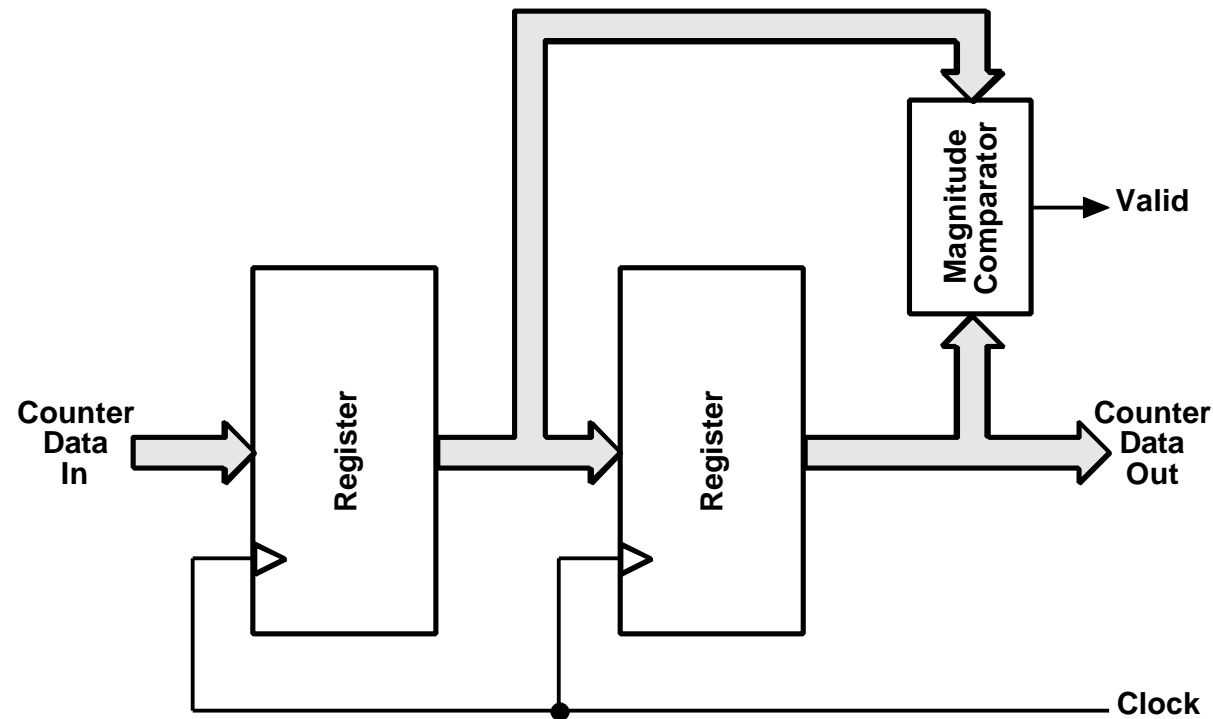
- ◆ **Transmitter: Data available raises Ready, sets Flag**
 - *Receiver scans F, accepts parallel data, raises Acknowledge*
- ◆ **Acknowledge sets flip-flop, which resets Flag**
 - *Benign race condition between flip-flops*
- ◆ **Both sides must observe and obey the Flag**

Moving Parallel Data without Handshake



- ◆ If Rx is much faster than Tx:
- ◆ Double-buffer the Data and compare
 - If both buffers are identical: good data
 - If both are not identical: wait
- ◆ Identity detector can also be transition detector

Transfer Counter Value without Handshake



- ◆ Comparator detects “reasonable” difference
- ◆ Rejects absurd differences only

Moving Data at Full Speed

- ◆ **200 MHz asynchronous FIFO in Virtex-II**
 - *16K deep, n bits wide*
 - *to*
 - *512 deep, $36n$ bits wide*
- ◆ **Uses n BlockRAMs for data storage**
- ◆ **Only eight to eleven CLBs for control**

See new app note in March 2001

Moving Data at Full Speed

- ◆ **200 MHz asynchronous FIFO in Virtex**
 - *4K deep, n bits wide*
 - *to*
 - *512 deep, $8n$ bits wide*
- ◆ **Uses n BlockRAMs for data storage**
- ◆ **Only 12 to 16 CLBs for control**

See new app note in March 2001

Asynchronous FIFOs

- ◆ Parameters: width, depth, clock frequency
- ◆ Data path = dual-ported BlockRAM
- ◆ Control = 2 addresses + Full, Empty
- ◆ Synchronous control is very simple:
 - *Two counters + trivial state machines*
- ◆ Asynchronous control is very tricky
 - *Asynchronous addresses must control FULL and EMPTY*

Many (most?) FIFOs are asynchronous

Full and Empty Control

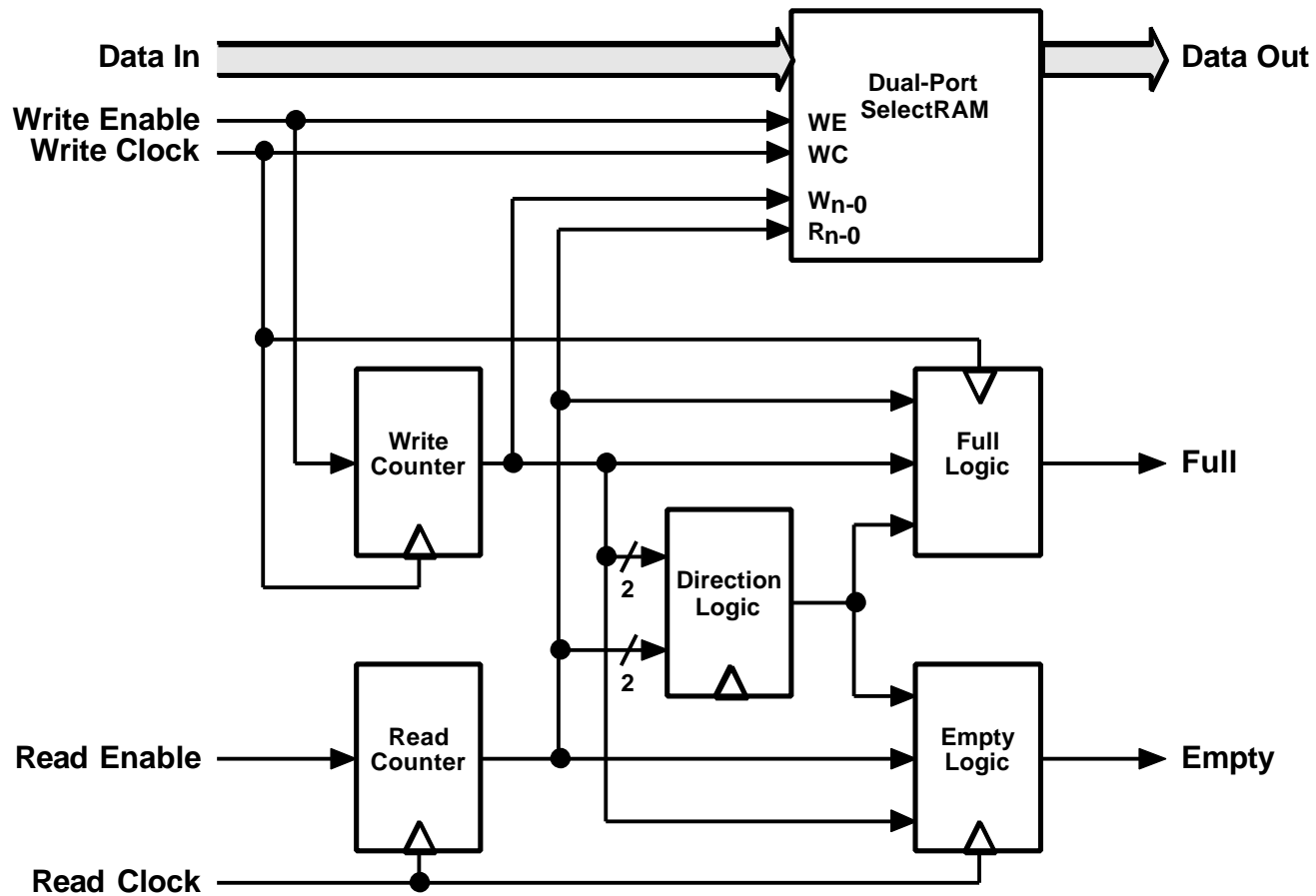
Identity-compare write and read addresses

— *identical addresses mean either Full or Empty*

Two problems:

- ◆ Comparing two asynchronously changing binary addresses will cause glitches
- ◆ Distinguish between Full and Empty
 - *both are indicated by address identity*

FIFO Block Diagram



Gray-Coded Addresses

- ◆ **Only one bit per address changes any time**
 - *no glitches from the identity comparator*
- ◆ **Implementation:**
 - *Build binary counter*
 - *Generate XOR of two adjacent D-inputs*
 - *Feed these XORs to a register = Gray code*
 - *MSB binary = MSB Gray*
- ◆ **Advantage:**
 - *Very fast and easily expandable, binary as a bonus*
 - *Takes advantage of the fast carry structure*

No pipeline delay, but twice the binary counter cost

Separate Full from Empty

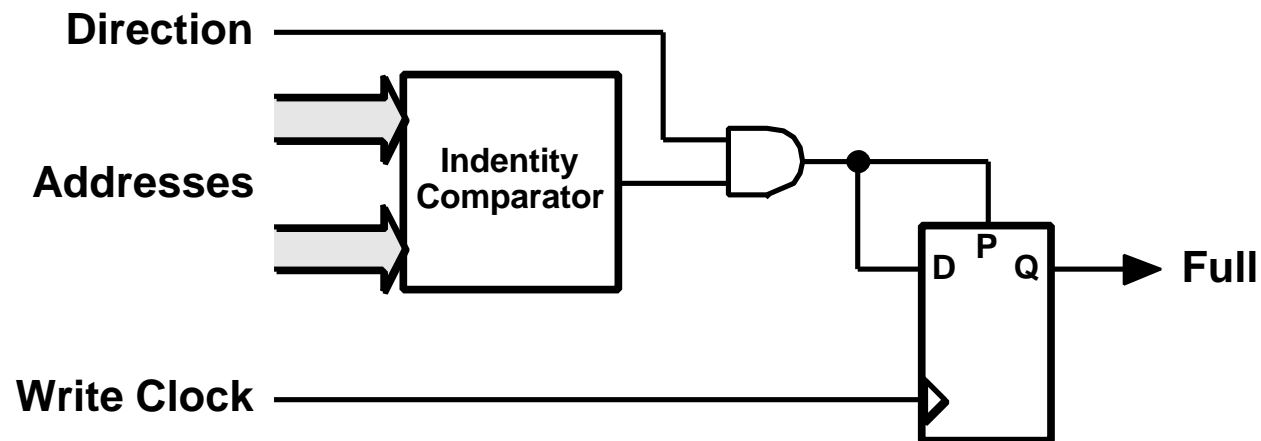
- ◆ **Divide address space into 4 quadrants, defined by the counter MSBs**
 - *This works in binary as well as in Gray*
- ◆ **Monitor the quadrant relationship of the write and read address counters**
- ◆ **Set a flag to distinguish between potentially going Full or Empty**
 - *include this in the address identity comparator*

Synchronize to the Proper Clock

- ◆ **FULL must be synchronous to write clock**
 - *Read is not concerned with fullness*
- ◆ **EMPTY must be synchronous to read clock**
- ◆ **Leading edges are naturally synchronous:**
 - *Full is the result of a write clock*
 - *Empty is the result of a read clock*

Trailing edges are caused by the other clock

Synchronizing the Trailing Edges



- ◆ **Combinatorial FULL is the result of a write.**
 - *Use it to asynchronously preset a flip-flop.*
 - *Use it also as D-input, clocked by the write clock.*

This synchronizes both edges to the write clock.

Do the Same with EMPTY

- ◆ **EMPTY can share the identity decoder**
 - *Then individually gated by Direction*
- ◆ **You can also put the binary outputs to good use:**
 - *they can provide “dipstick” indication:*
 - *Subtract, but beware of glitches.*

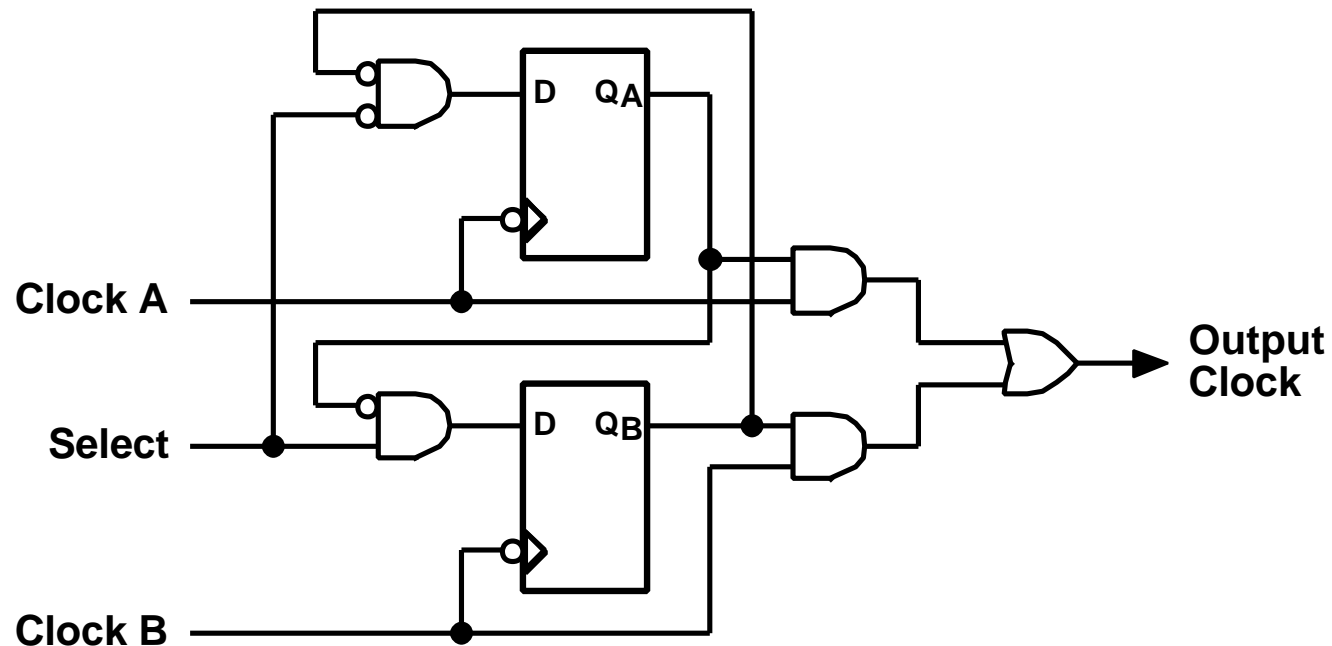
Asynchronous FIFO in Virtex

- ◆ **180 MHz asynchronous operation**
 - *4K deep, 1n bits wide*
 - *2048 deep, 2n bits wide*
 - *1024 deep, 4n bits wide*
 - *512 deep, 8n bits wide*
- ◆ **Uses n BlockRAMs plus 16 to 20 CLBs**
 - *BlockRAMs for data storage*
 - *CLBs for address counters, direction detection, EMPTY and FULL detection across asynchronous boundary*

Asynchronous FIFO in Virtex-II

- ◆ **200 MHz asynchronous operation**
 - 16K deep, n bits wide
 - 8K deep, $2n$ bits wide
 - 4K deep, $4n$ bits wide
 - 2048 deep, $9n$ bits wide
 - 1024 deep, $18n$ bits wide
 - 512 deep, $36n$ bits wide
- ◆ **Uses n BlockRAMs plus 8 to 11 CLBs**
 - BlockRAMs for data storage
 - CLBs for address counters, direction detection, EMPTY and FULL detection across asynchronous boundary

Asynchronous Clock MUXing

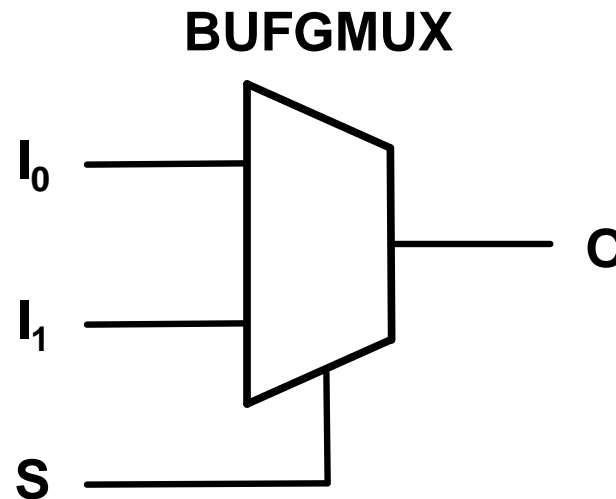


- ◆ This circuit waits for the present clock to go Low
 - *Output then stays low until the new clock is Low*

Guaranteed to switch glitch-free, no runt pulses

— http://www.xilinx.com/xcell/xl24/xl24_20.pdf

Virtex-II Clock Multiplexer



- ◆ **Each global clock buffer is a mux**
 - *can switch between 2 clock sources*
 - *configured for rising or falling edge*
- ◆ **Can also do clock gating (enable)**

Dangerous stuff, but these circuits do it safely

Conclusions

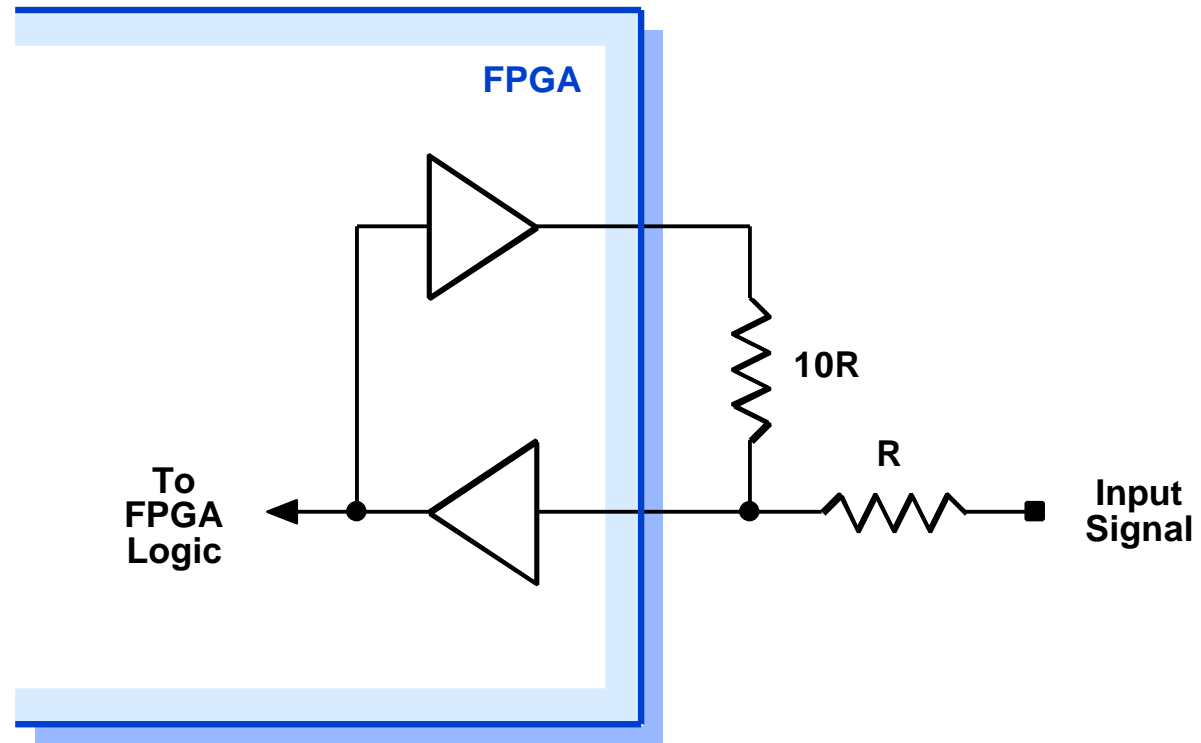
- ◆ **Asynchronous data transfer is dangerous**
 - *but not if you understand the issues and know how to design around them*
- ◆ **Clock gating is unhealthy**
 - *but not if you use smart circuits*
- ◆ **Metastability can hurt very badly**
 - *but only if inside a very tight timing budget*

Modern CMOS resolves very fast (within a few ns)



Tips and Tricks from the Xilinx Archives

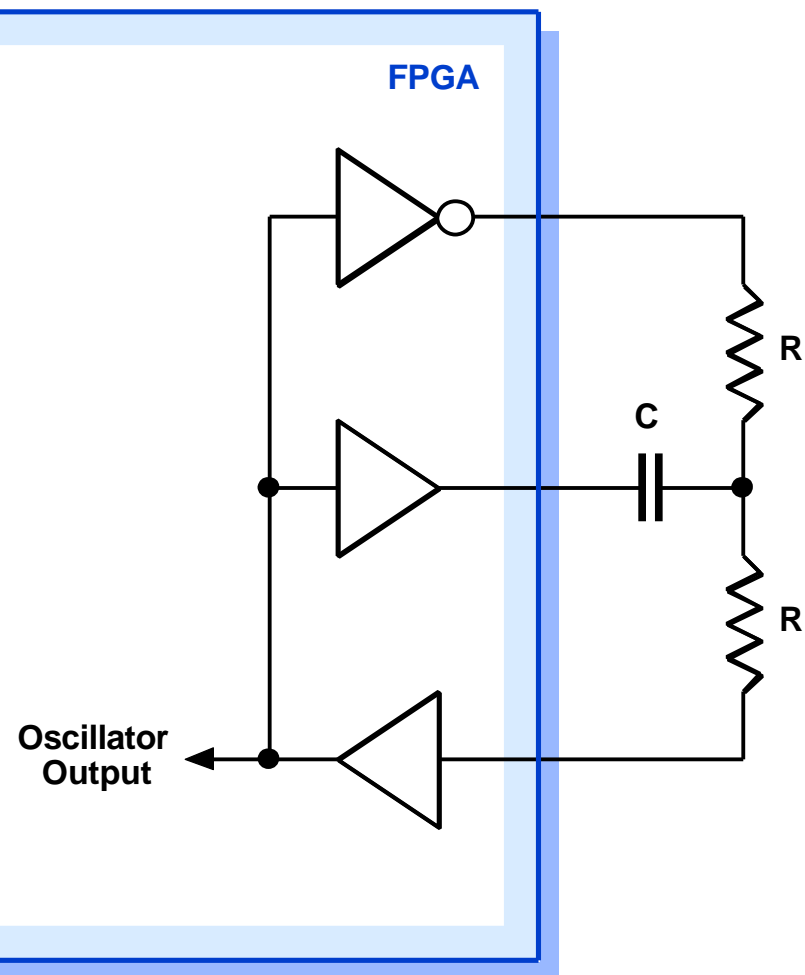
Schmitt Trigger



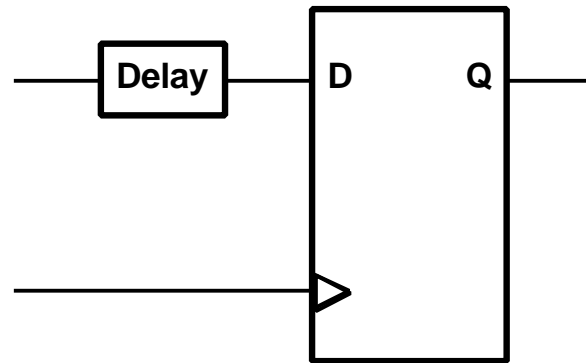
- ◆ Hysteresis = 10% of V_{cc}

RC Oscillator

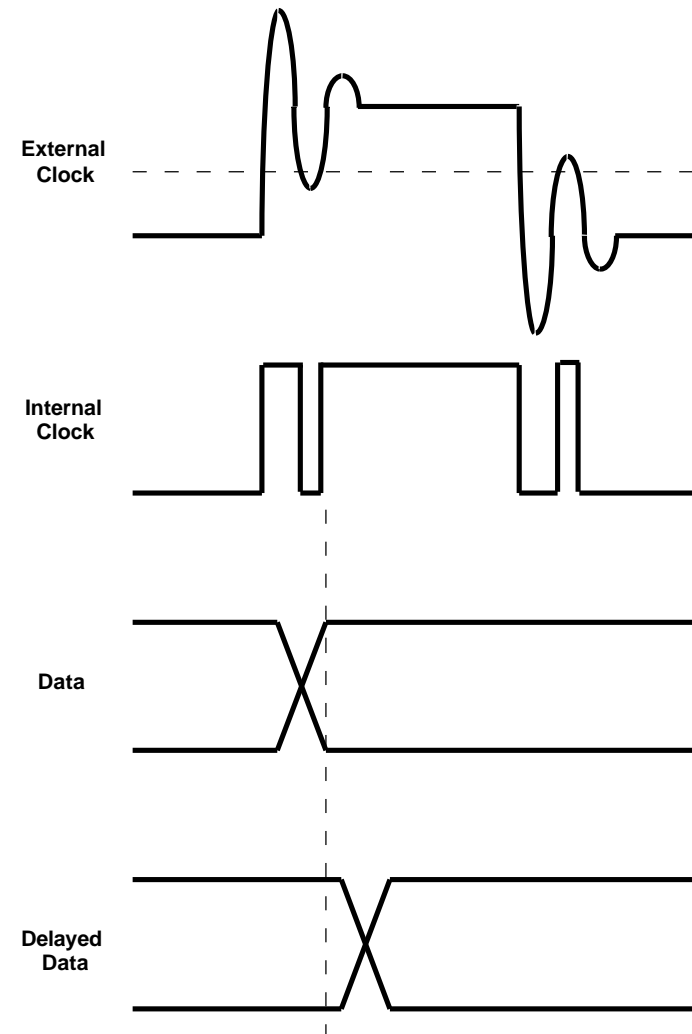
- ◆ **Wide frequency range, Hz to MHz**
 - *100 Ohm to 100 kilohm*
 - *100 pF to 1 uF*
- ◆ **Reliable start-up is absolutely guaranteed**
- ◆ **Oscillator can be started and stopped internally**



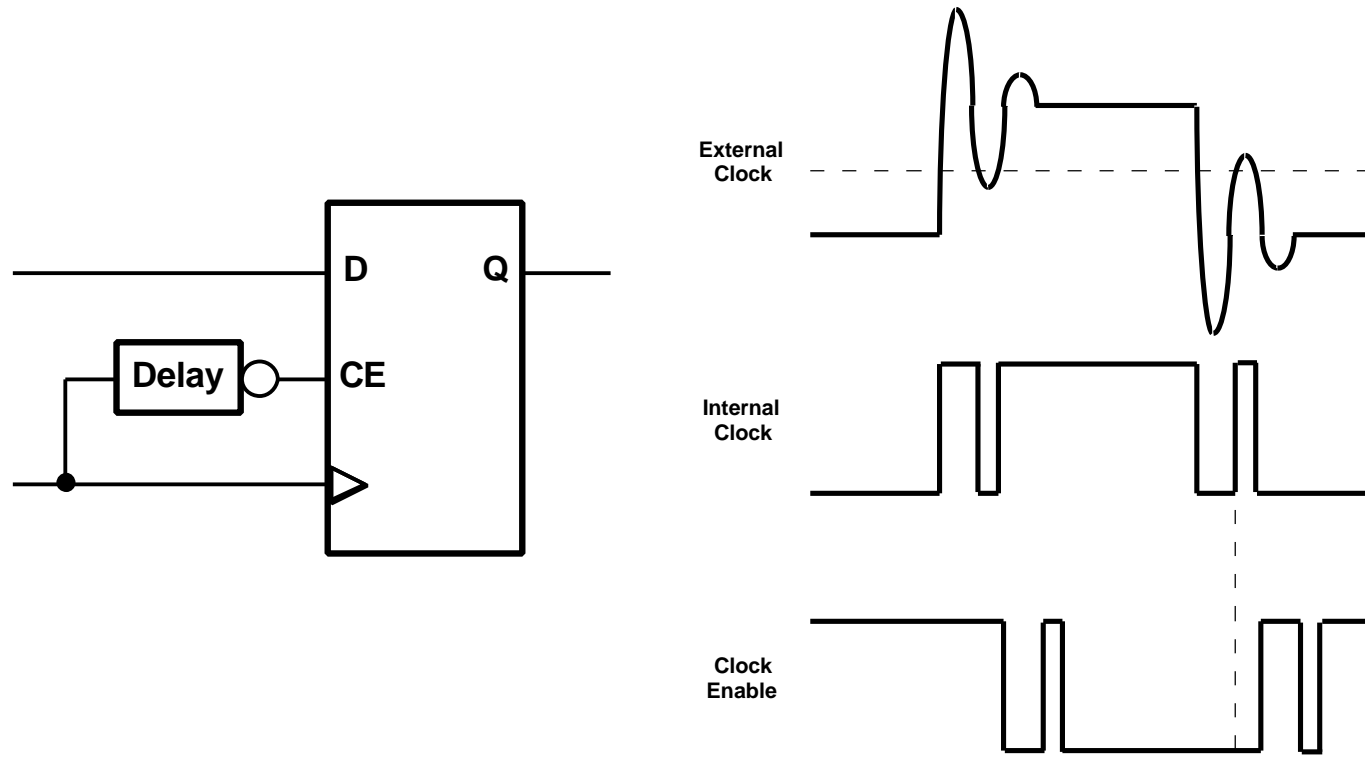
Coping with Clock Reflections



- ◆ **Problem: Double pulse on the active edge**
- ◆ **Solution: Delay D, to prevent the flip-flop from toggling soon again**

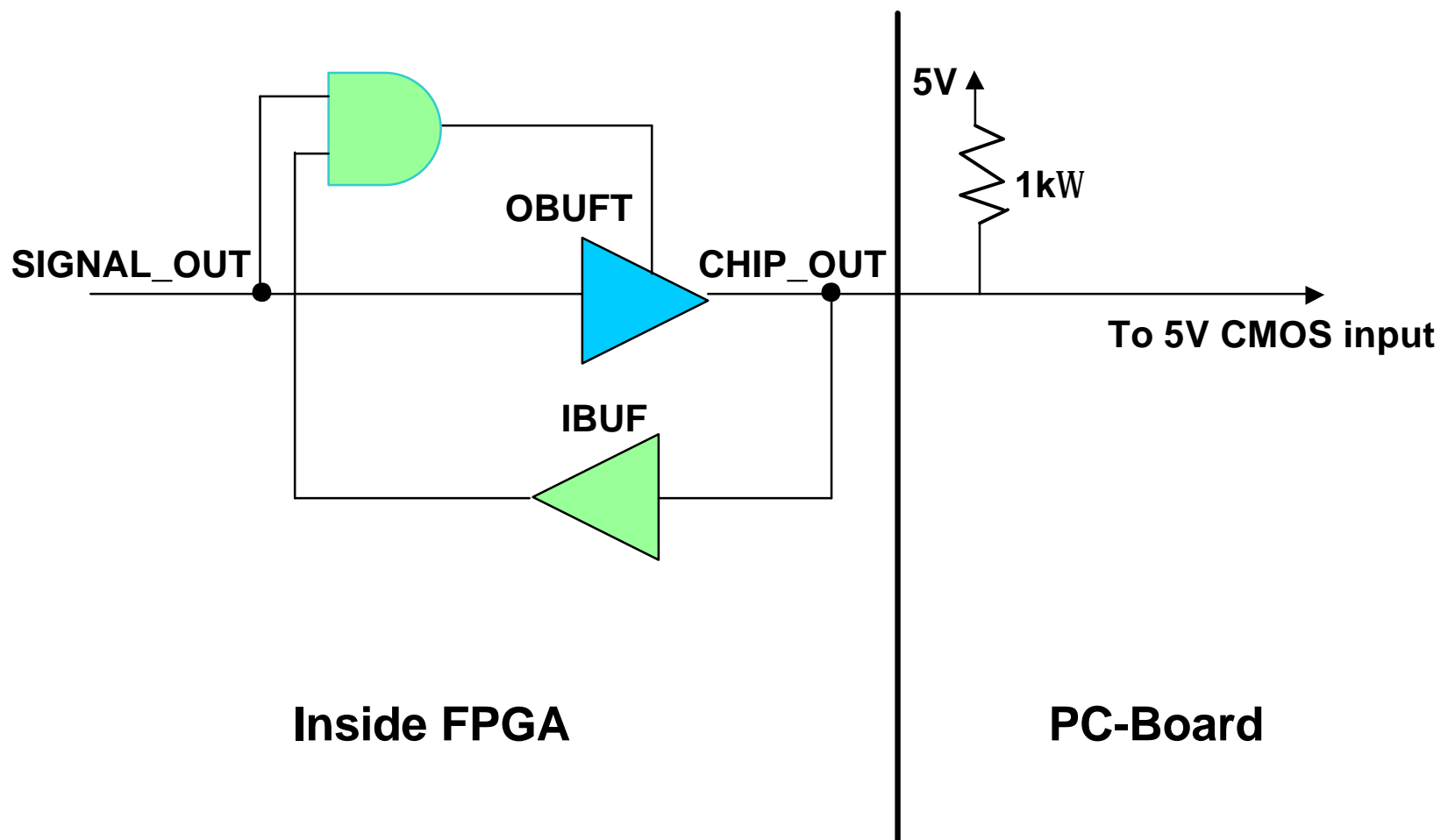


Coping with Clock Reflections



- ◆ **Problem: Double pulse on the inactive edge**
- ◆ **Solution: Disable flip-flop, by using the clock level**
— http://www.xilinx.com/xcell/xl34/xl34_54.pdf

5V-Tolerant 3.3V Output Driving 5V CMOS-Level Input



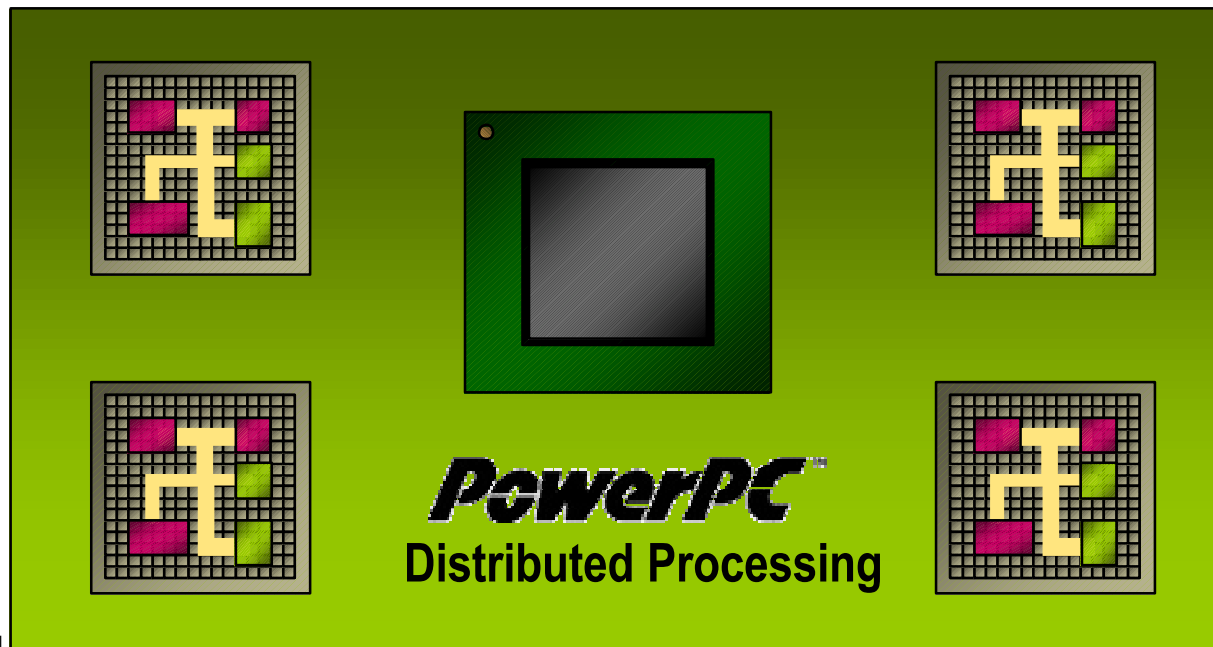
Virtex-II, the Next Generation

- ◆ **0.13 μ , 8-layer metal CMOS process**
 - *Cu power distribution and interconnect*
- ◆ **Up to 10 million system gates**
 - *>100,000 LUTs and flip-flops*
 - *>1000 BlockRAMs and multipliers*
 - *>200 MHz clock rate, multi-Gbps serial I/O*
- ◆ **On-chip PowerPC with cache**
- ◆ **3 Gigabit Serial I/O**



***PowerPC - the Leading Embedded CPU Architecture in
Telecom & Networking Infrastructure***

***Virtex-II Platform FPGA Adds Distributed PowerPC Processing
with Application-Specific Hardware Acceleration***



Putting it all together: The Virtex-II Series Platform FPGA.

World's Fastest Logic & Routing

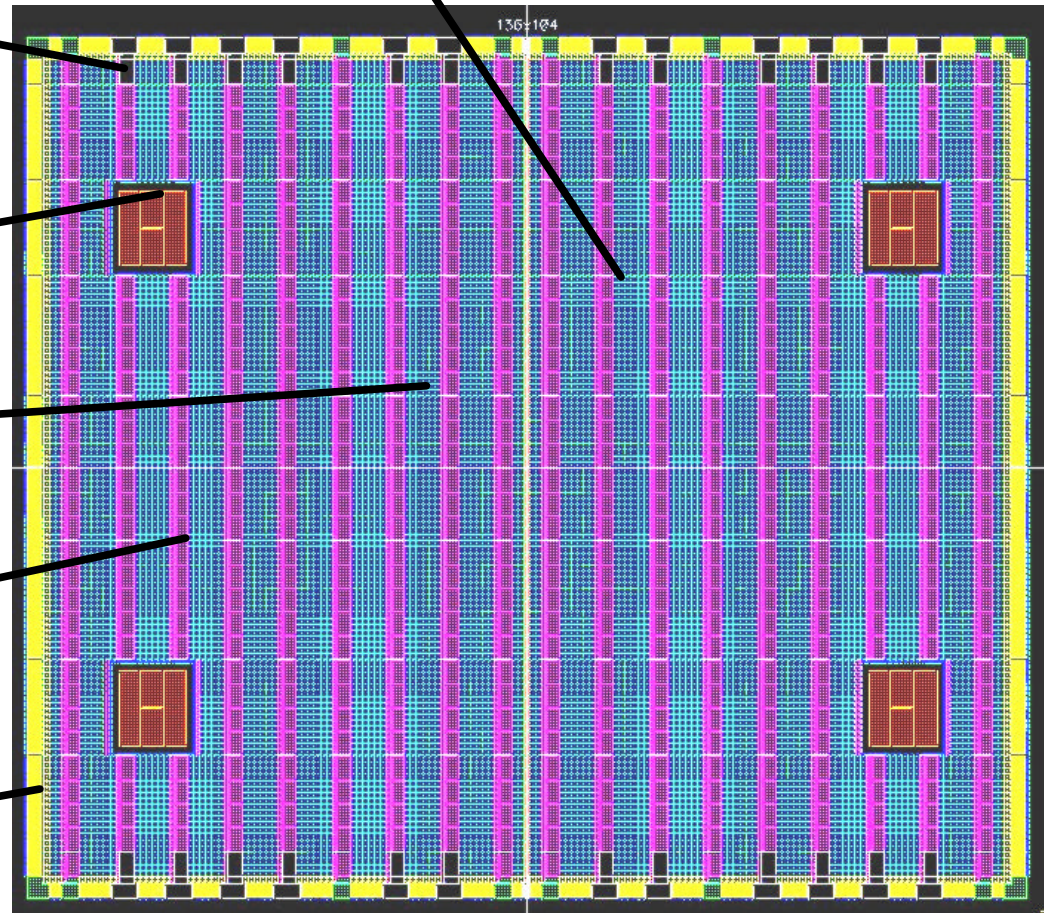
3.125Gb Serial

**IBM PowerPC®
RISC CPU**

XtremeDSP™

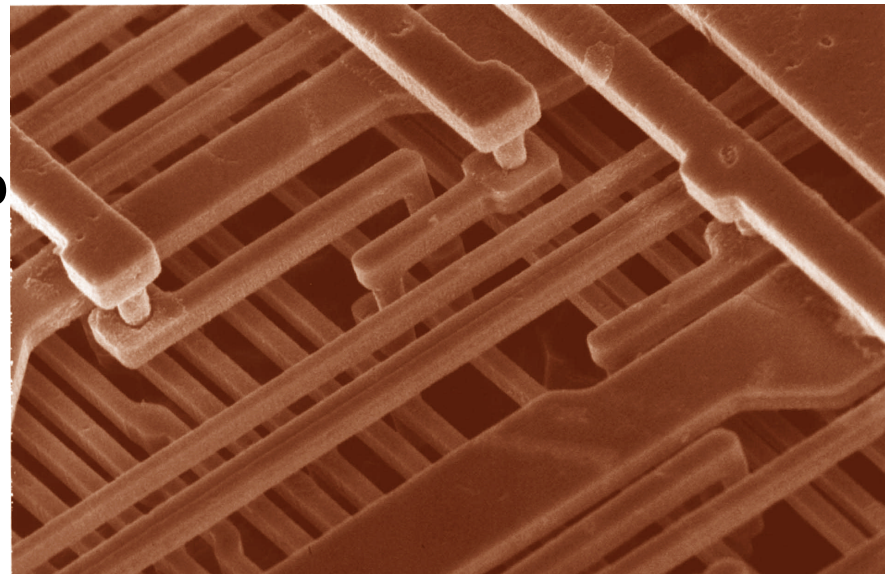
**Synchronous
Dual-Port RAM**

**SelectIO Ultra™
& SystemIO™**

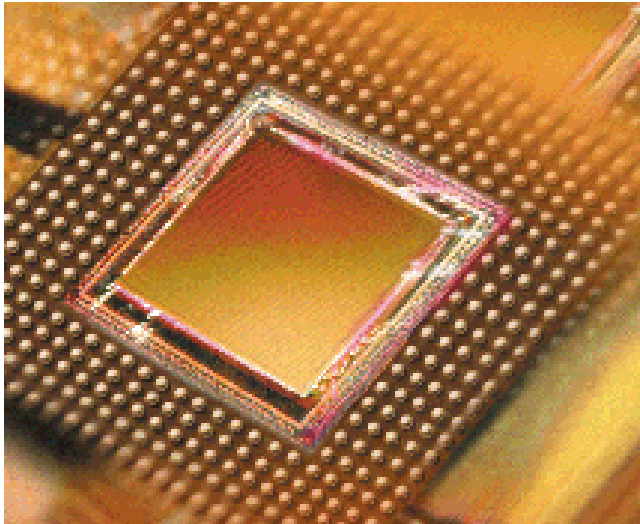


Advanced 0.13 micron CMOS

- ◆ **World Logic Partnership**
 - *IBM, Xilinx, UMC, Infineon*
- ◆ **Ultra-high speed 92 nm transistor technology**
- ◆ **8-layer Cu combined with low-K dielectric**
- ◆ **12" volume-production on three continents**



FPGAs circa 2005



- ◆ 50 Million system gates
- ◆ 2 Billion transistors on one chip
- ◆ 70-nm process technology
- ◆ 10-layer Cu technology
- ◆ Hard and soft IP blocks
- ◆ 1 GHz embedded processor
- ◆ Mixed-signal IP
- ◆ 10-Gigabps I/O channels

Beyond Bigger, Faster, Cheaper ...

On-chip RAM

Efficient Arithmetic

Intelligent Clock Management

Multi-standard I/O, Built-In Termination

*FPGAs have evolved from glue logic
to cost-effective system platforms*

Additional Information on the Following Pages:

- ◆ List of good URLs
- ◆ Virtex-II architecture
- ◆ Clock distribution nets
- ◆ Clock management

Clock de-skew

Frequency Synthesizer

Phase shifter

Clock gating

Clock multiplexing

List of Good URLs

- ◆ **www.xilinx.com**
- ◆ **www.xilinx.com/support/sitemap.htm**
 - www.xilinx.com/products/virtex/handbook/index.htm
 - www.xilinx.com/support/techxclusives/techX-home.htm
 - www.xilinx.com/support/troubleshoot/psolvers.htm

General FPGA-oriented Websites:

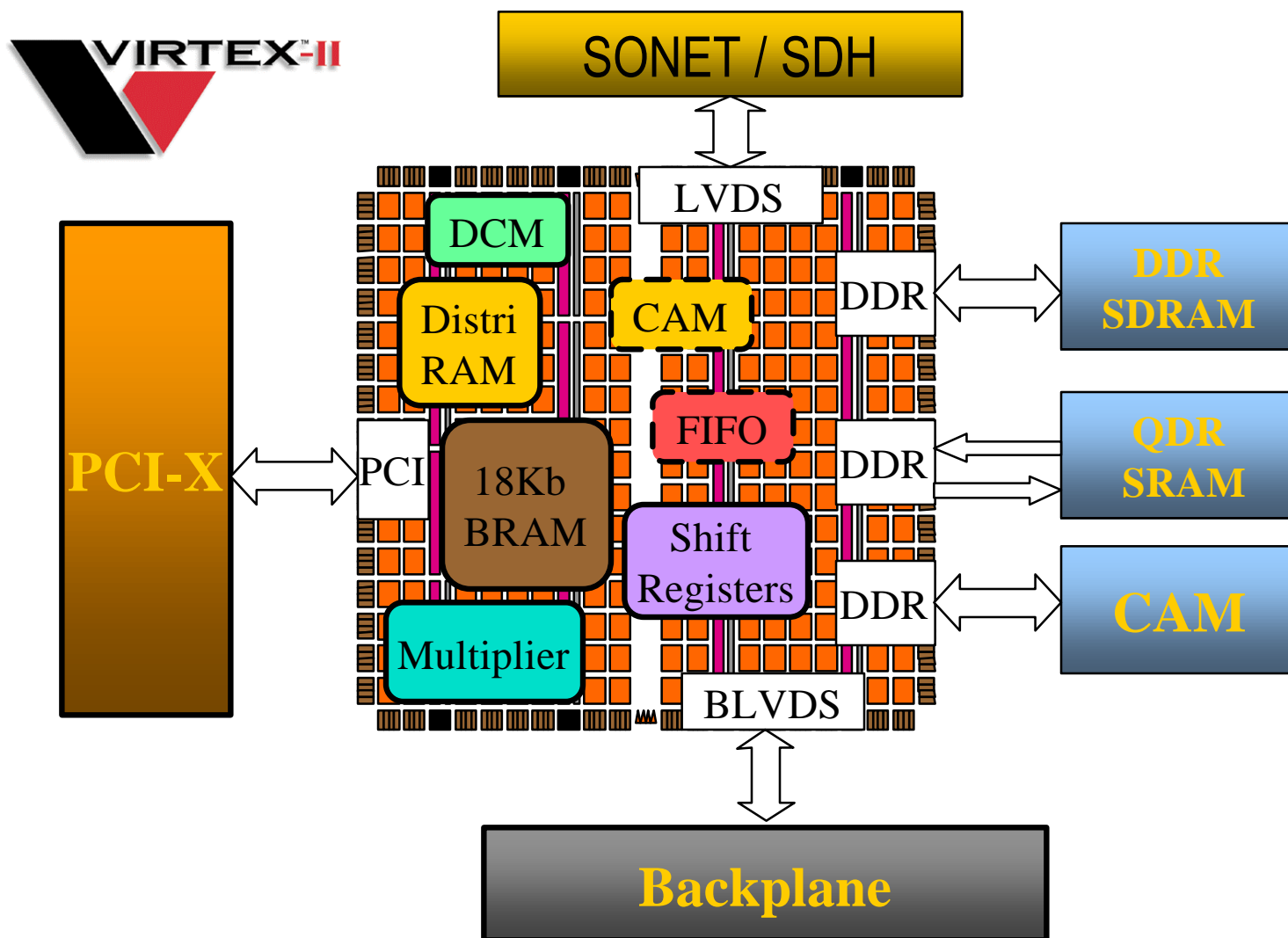
- www.fpga-faq.com
- www.optimagic.com

Newsgroup: comp.arch.fpga

All datasheets: www.datasheetlocator.com

Search Engine (*personal preference*): www.google.com

Virtex-II Architecture

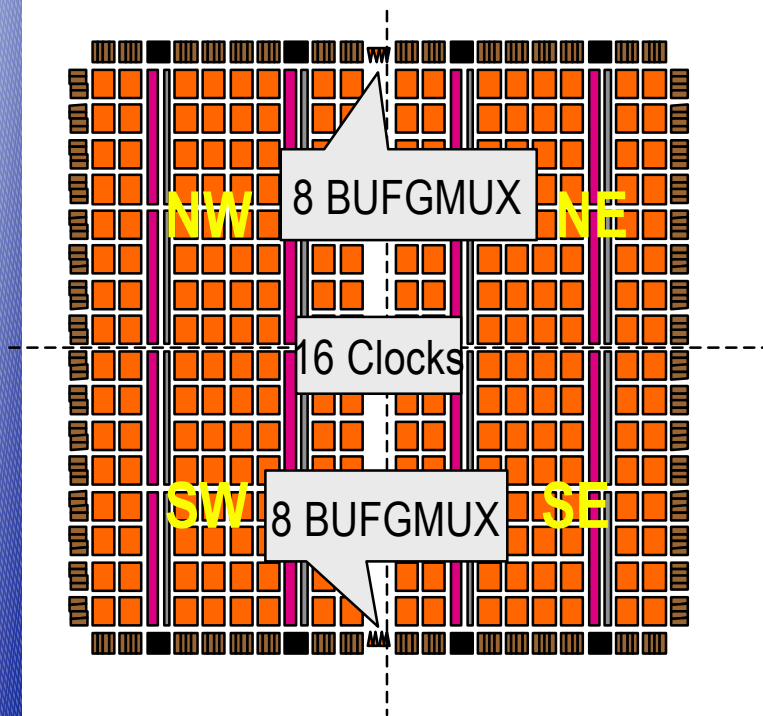


Enhanced Clock Distribution

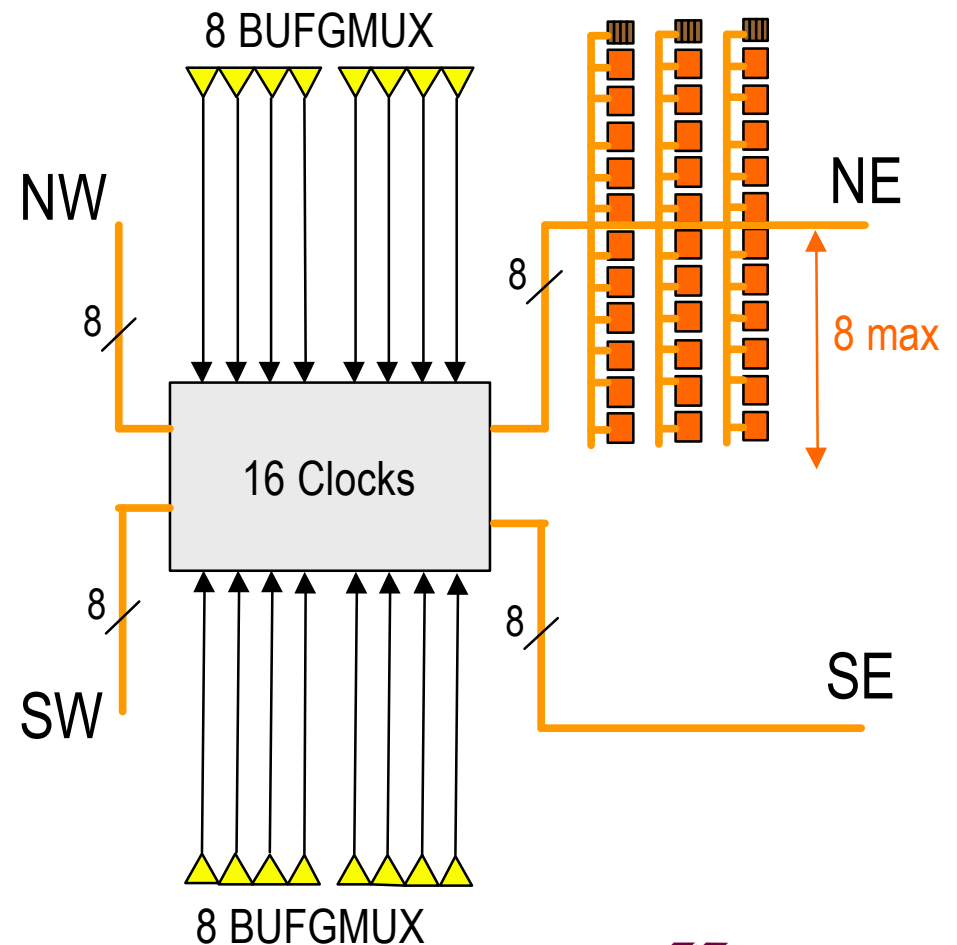
- ◆ **16 Global Clock Multiplexers**

- Eight on the top
- Eight on the bottom
- Switch “glitch free”
between two clock inputs

- ◆ **Eight clocks selectable for each quadrant**

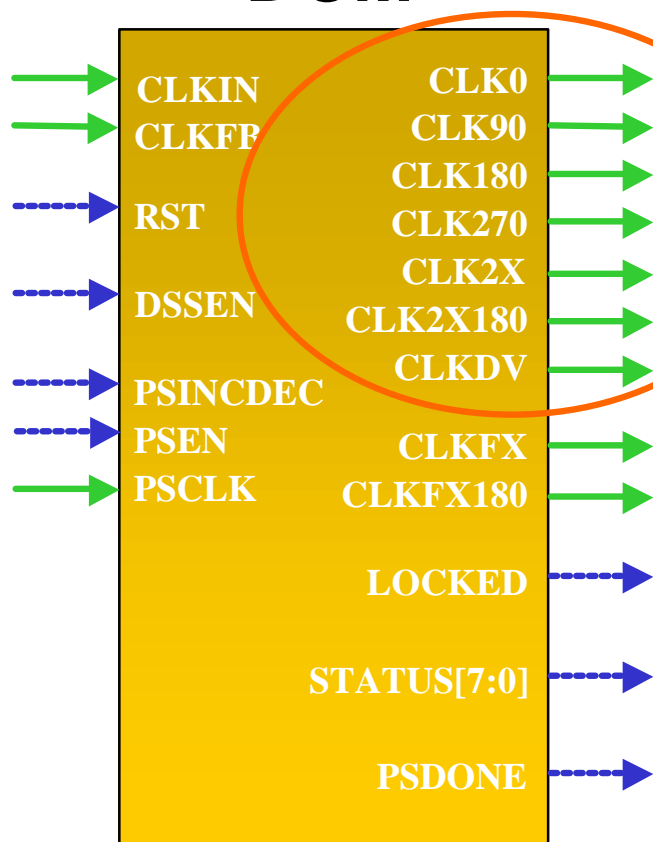


Unused branches are disabled
(Power Saving)



Digital Clock Manager: DLL

DCM



→ clock signal
→ control signal

◆ Digital Clock Manager

- Clock phase de-skew
- Temperature compensation

◆ Delay-Locked Loop

- Duty Cycle correction

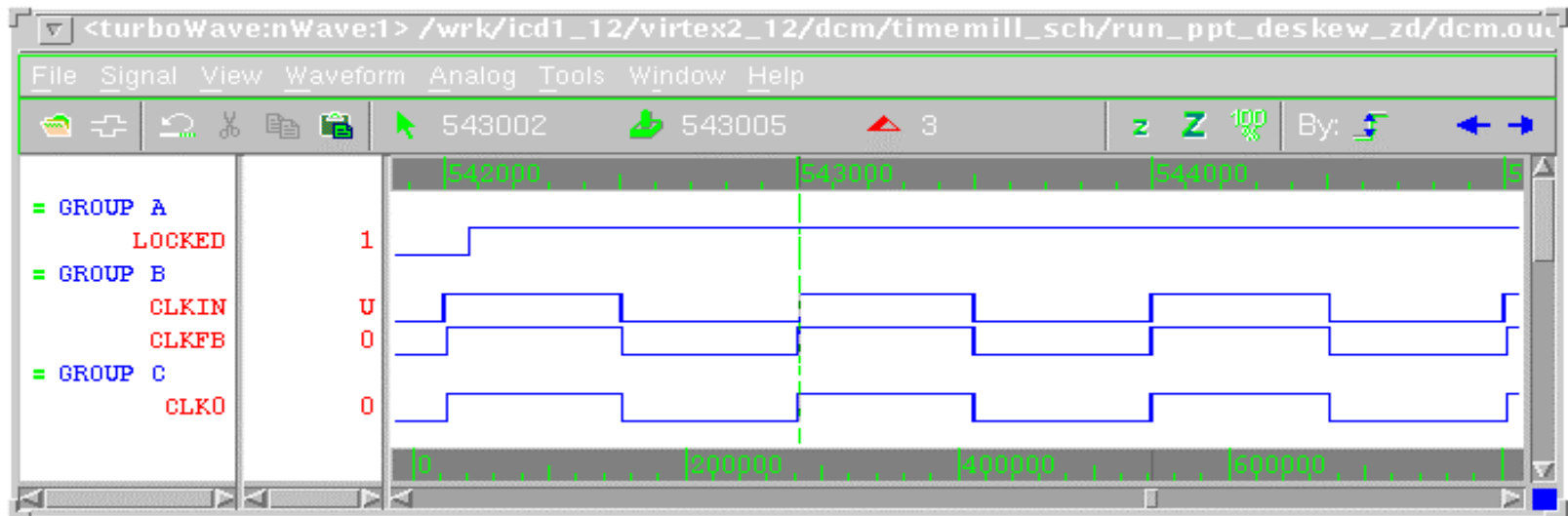
— Attributes:

- DUTY_CYCLE_CORRECTION
- DLL_FREQUENCY_MODE
- CLKDV_DIVIDE = 1.5 to 16.0
- STARTUP_WAIT
- CLK_FEEDBACK =
CLK0 or CLK2X

◆ Up to 4 clock outputs per DCM

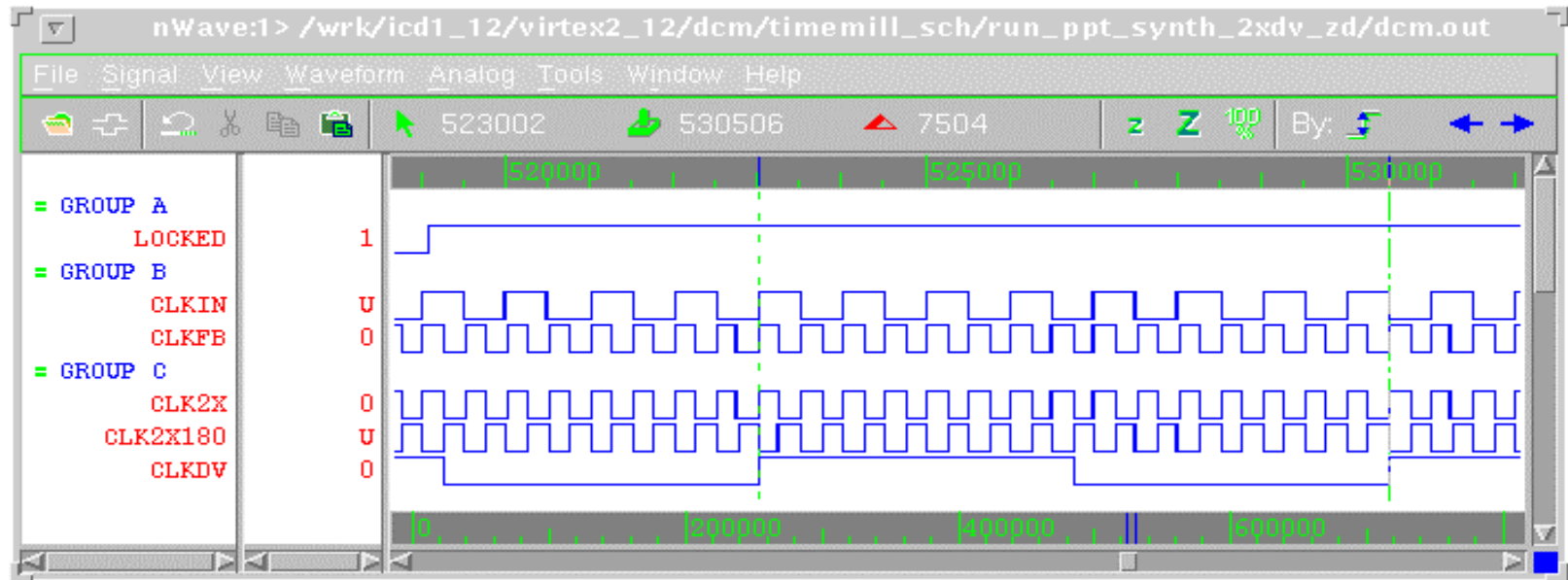
Clock De-skew

CLKIN = 100MHz



- ◆ Remove clock delay between input clock and flip-flop clock pins
- ◆ Create de-skewed board level clocks

Basic Frequency Synthesis

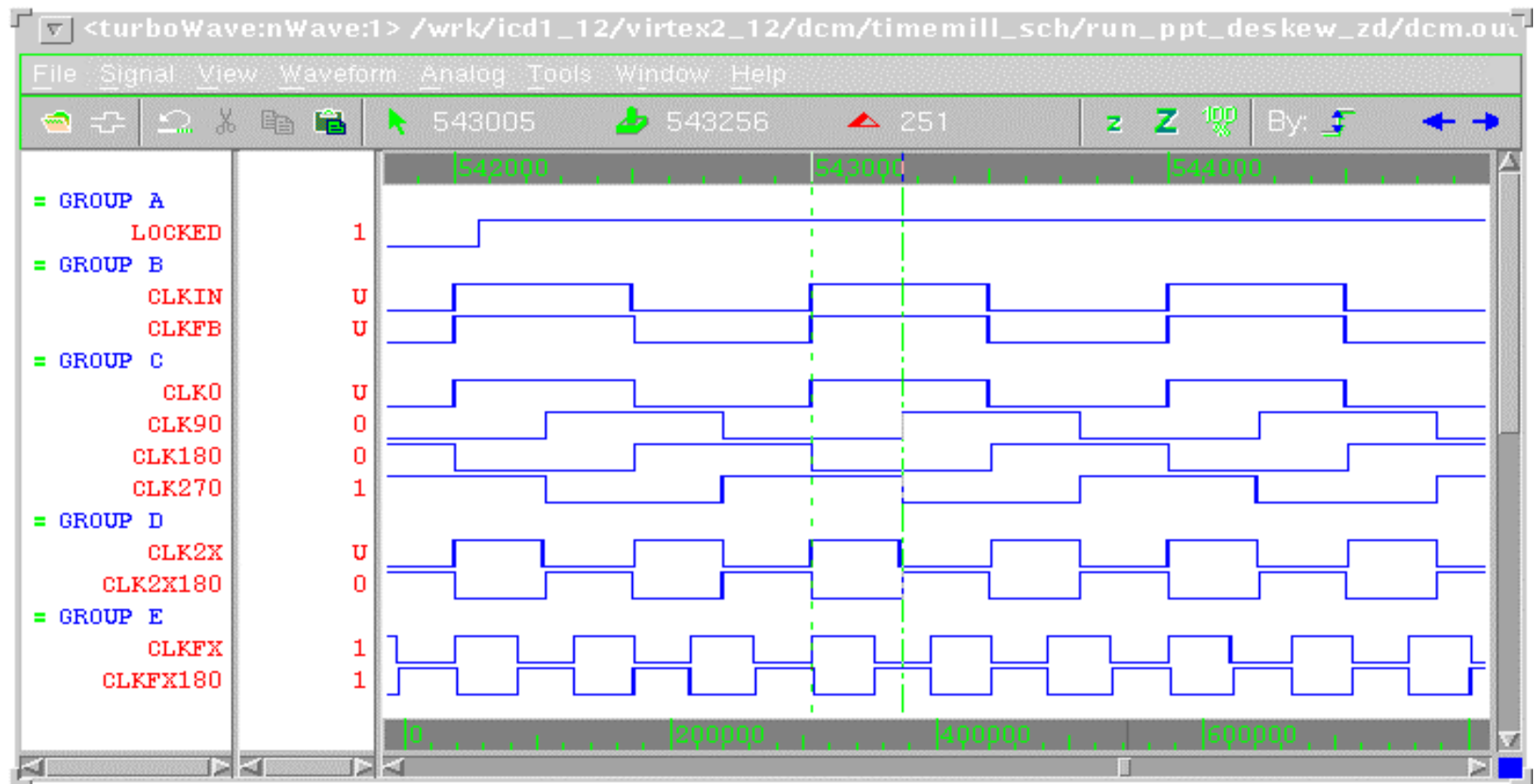


CLKIN = 100MHz

CLK2X = 200MHz

CLKDV = 13.3MHz (D=7.5)

Coarse Phase Shifting

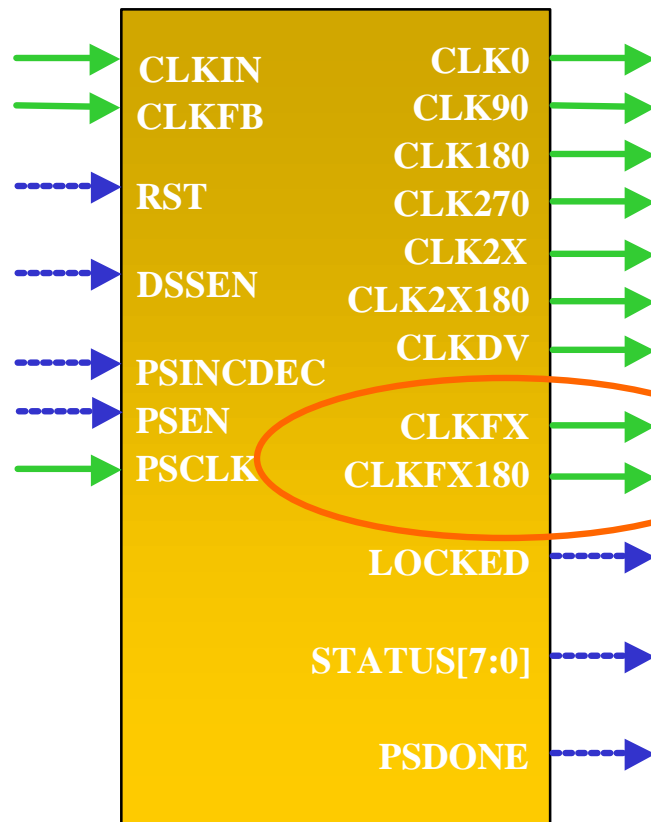


Coarse phase shifting = 0, 90, 180, 270 degrees

CLK0 = 100MHz, CLK2x = 200MHz, CLKFX = 300MHz

Advanced Frequency Synthesis

DCM



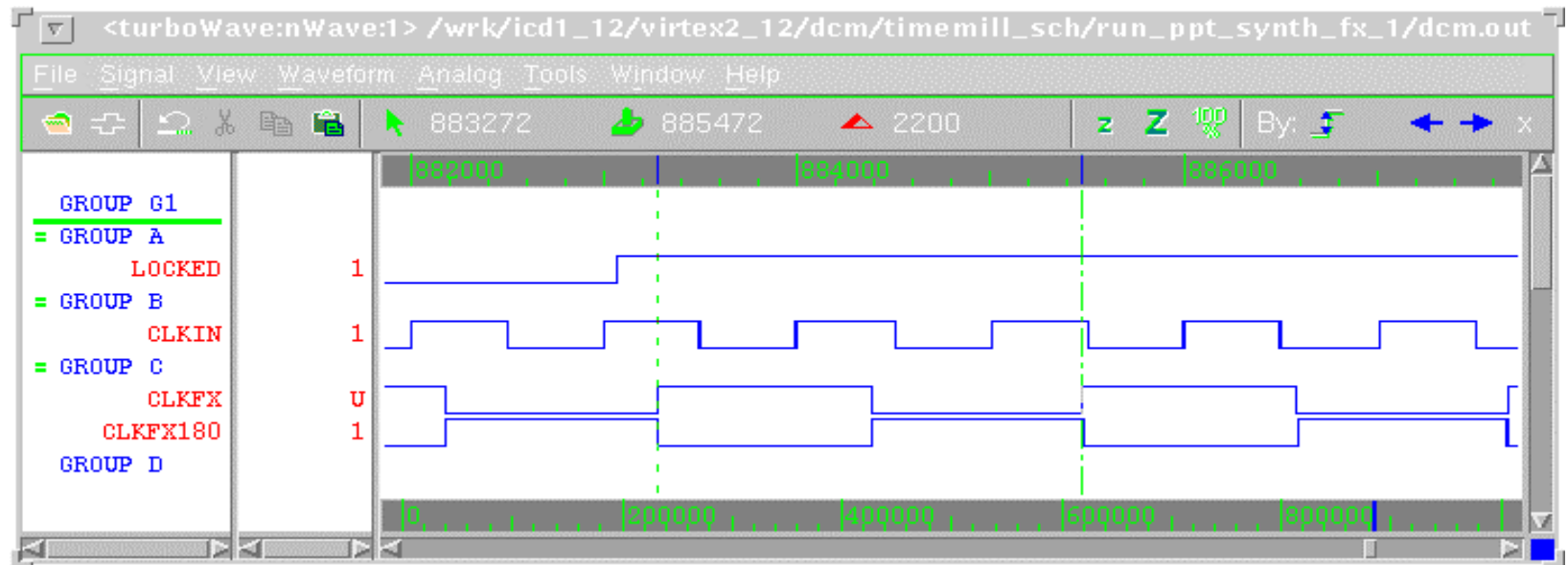
◆ Frequency Synthesis

- *CLKFX is any M / D product of the CLKIN frequency*
- *$M = 1$ to 4096, $D = 1$ to 4096*
- *Default: $M=4$, $D=1$ (4X CLKIN)*
- *Always nominal 50/50 duty-cycle*
- *Attributes:*
 - CLKFX_MULTIPLE (integer)
 - CLKFX_DIVIDE (integer)
 - DFS_FREQUENCY_MOD

After LOCKED:

$$\text{Freq}_{\text{CLKFX}} = (M/D) \times \text{Freq}_{\text{CLKIN}}$$

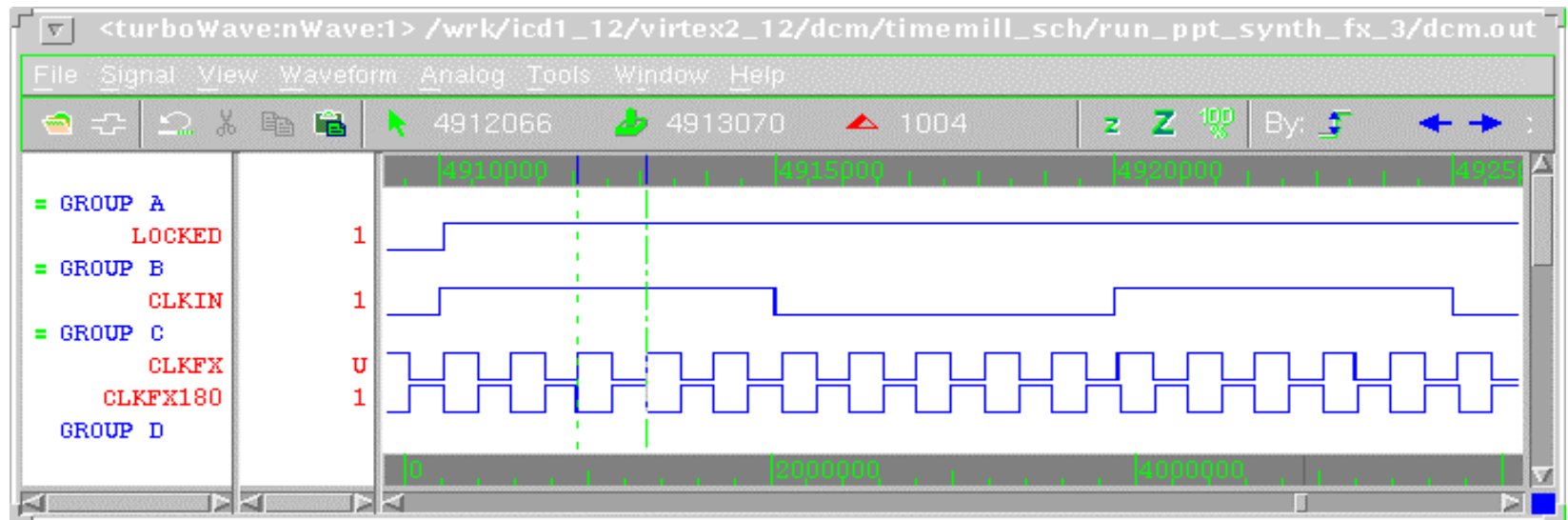
Advanced Frequency Synthesis



CLKIN = 100MHz

CLKFX = 45.45MHz ($M/D = 5/11$)

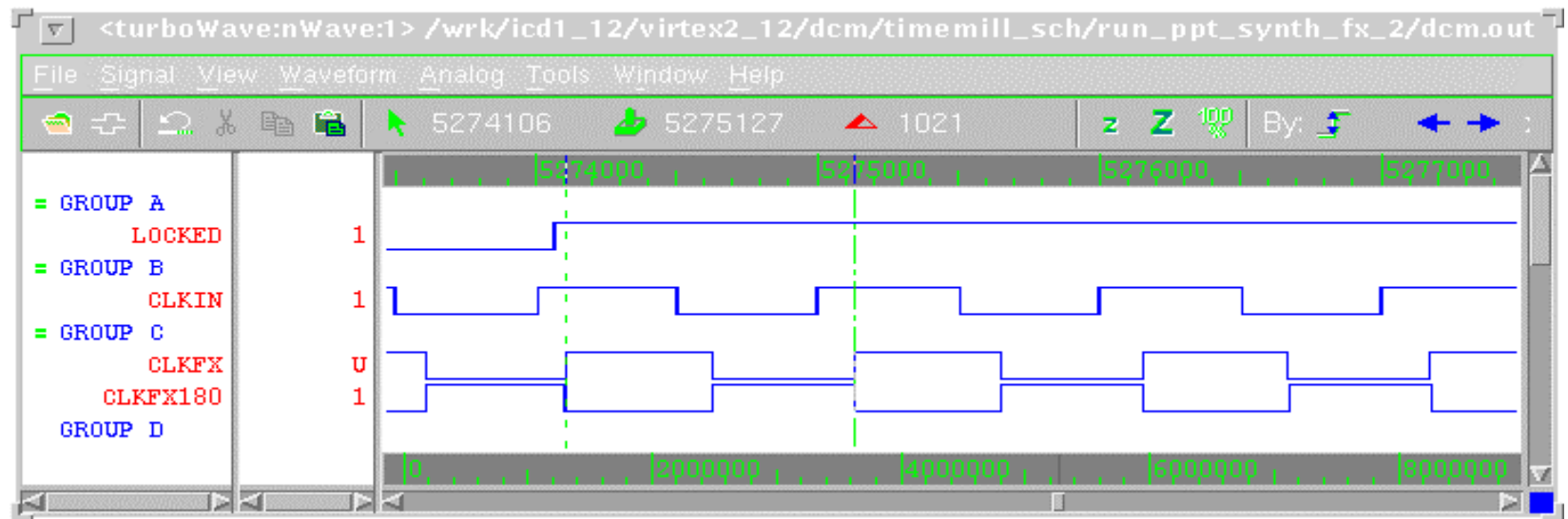
Advanced Frequency Synthesis



CLKIN = 10MHz

CLKFX = 100MHz (M/D = 10/1)

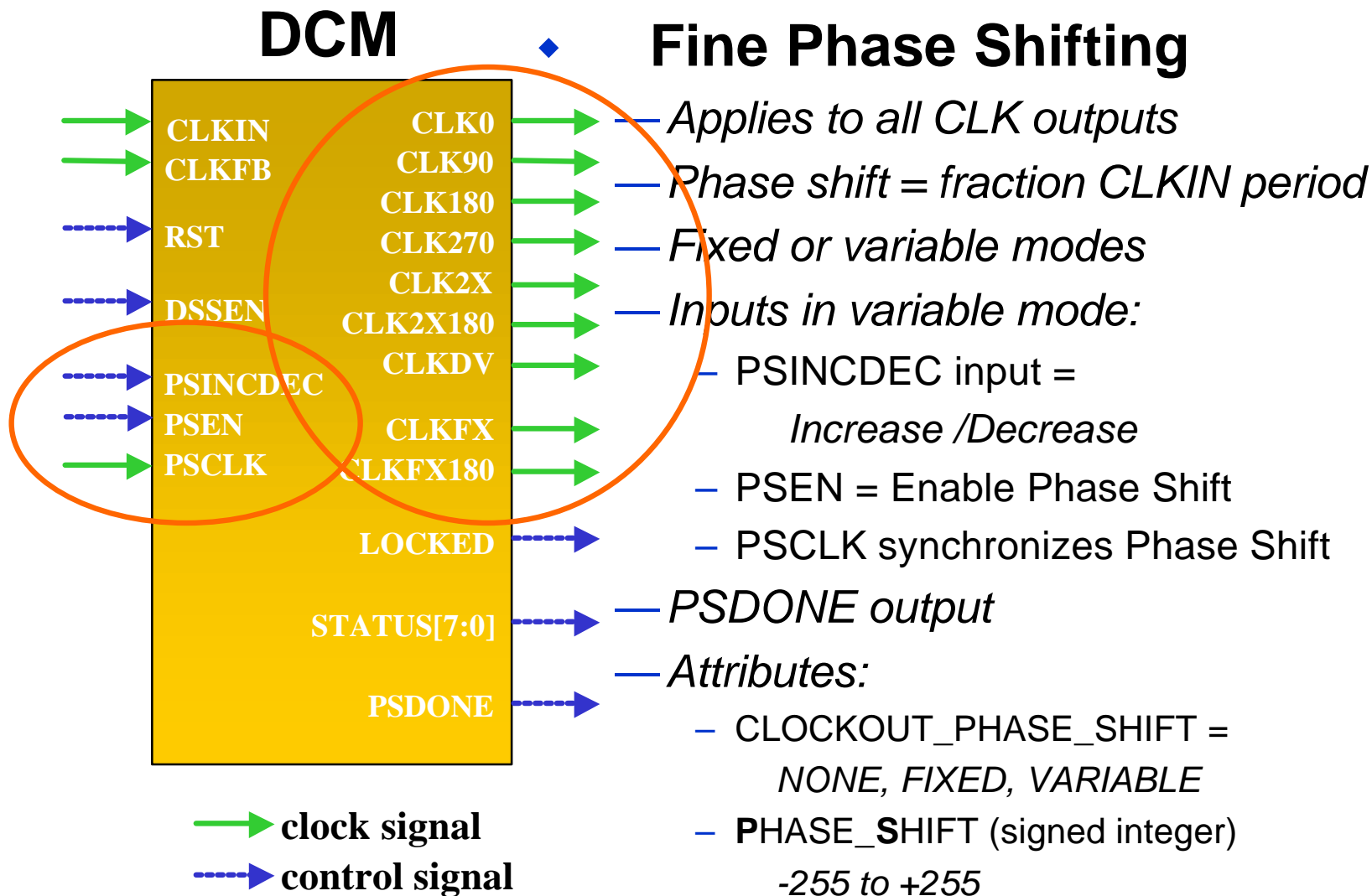
Advanced Frequency Synthesis



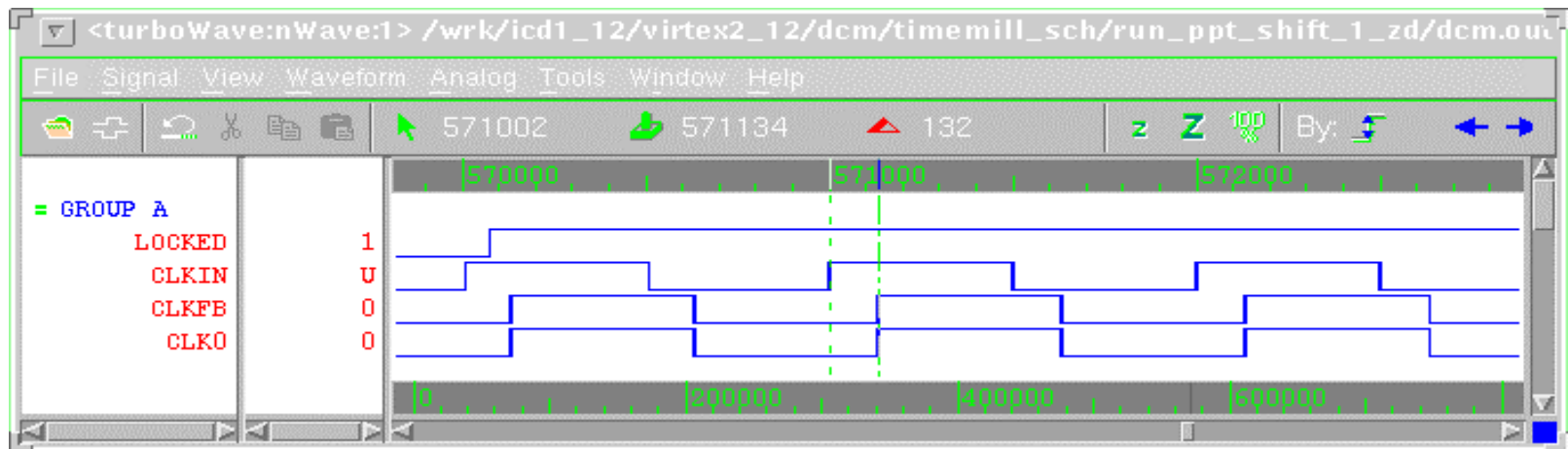
CLKIN = 100MHz

CLKFX = 98.1MHz ($M/D = 101/103$)

High Resolution Phase Shifting



High-Resolution Phase Shifting

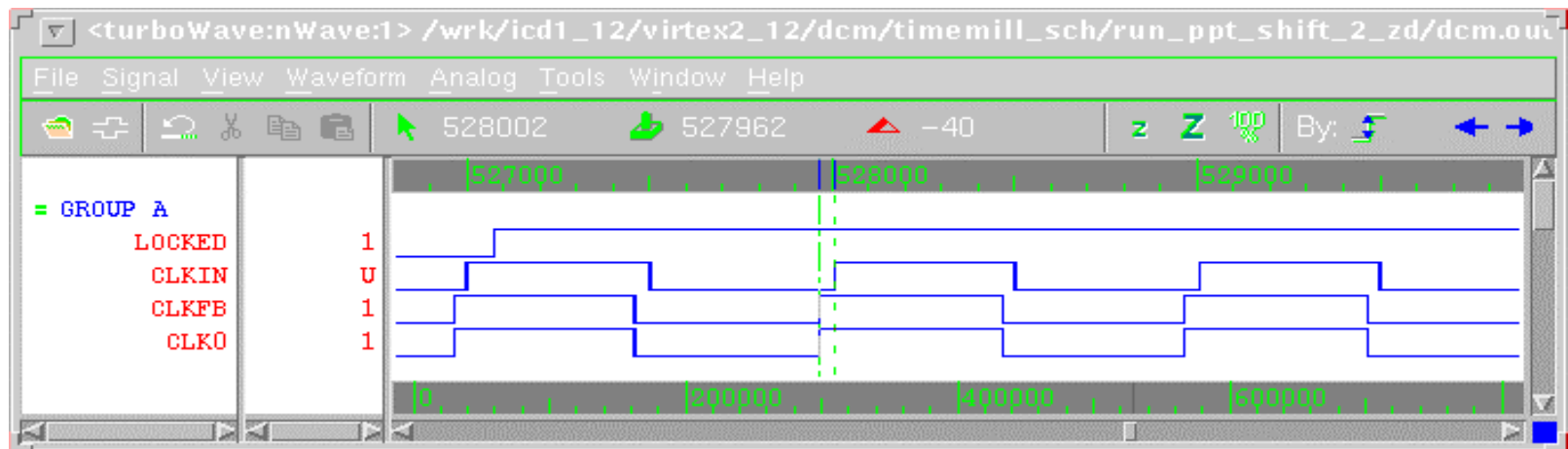


Desired phase shift = +1.3ns

CLKIN = 100MHz (10ns)

PS = (1.3ns/10ns) x 256 = 33

High-Resolution Phase Shifting

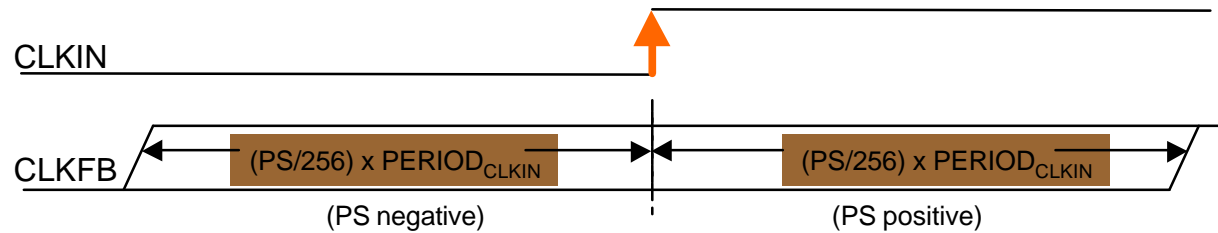


Desired phase shift = -15 degrees

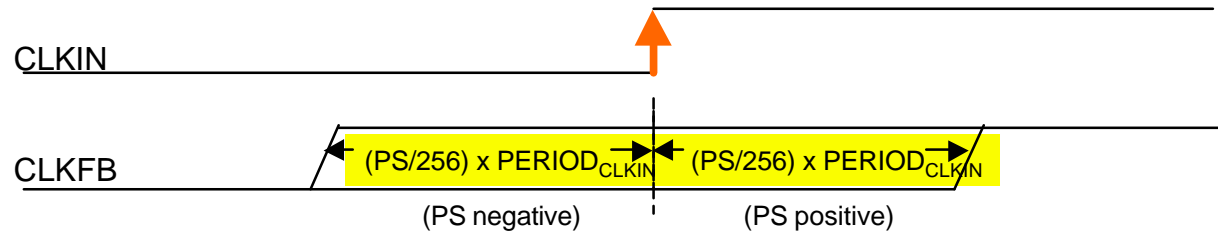
$$PS = (-15^\circ/360^\circ) \times 256 = -11$$

Phase Shift Effects

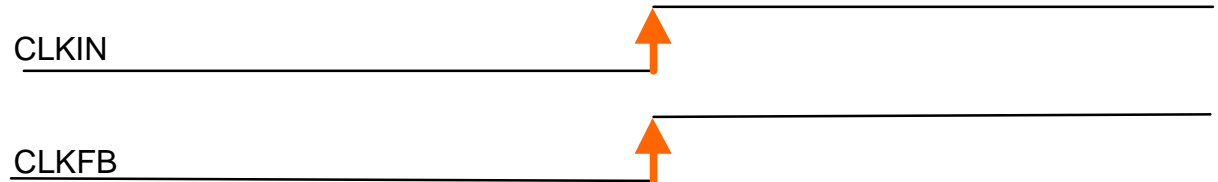
CLKOUT_PHASE_SHIFT
= FIXED



CLKOUT_PHASE_SHIFT
= VARIABLE



CLKOUT_PHASE_SHIFT
= NONE



Global Clocks: BUFGMUX

◆ Three Modes:

— Clock Buffer

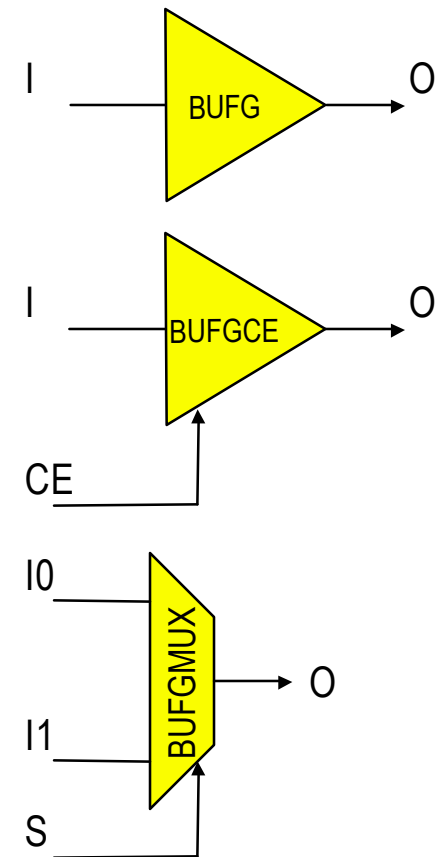
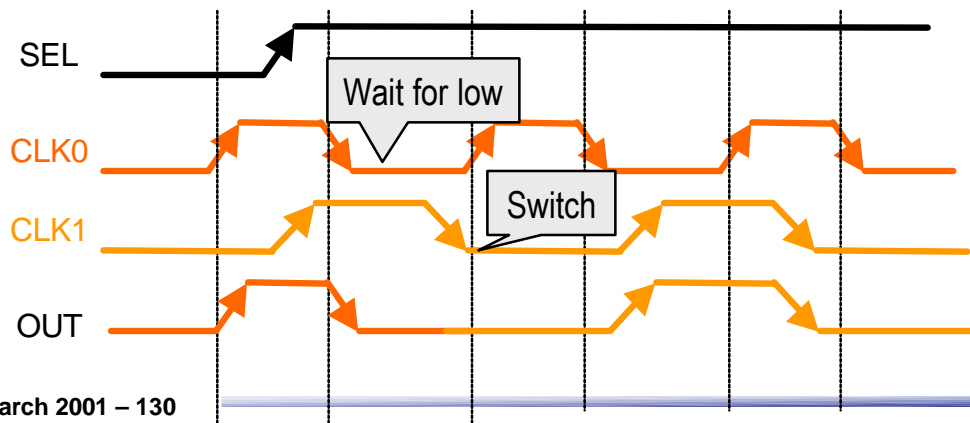
- Low skew clock distribution
- BUFG primitive

— Clock Enable

- Stop the clock High or Low
- BUFGCE (stop Low)

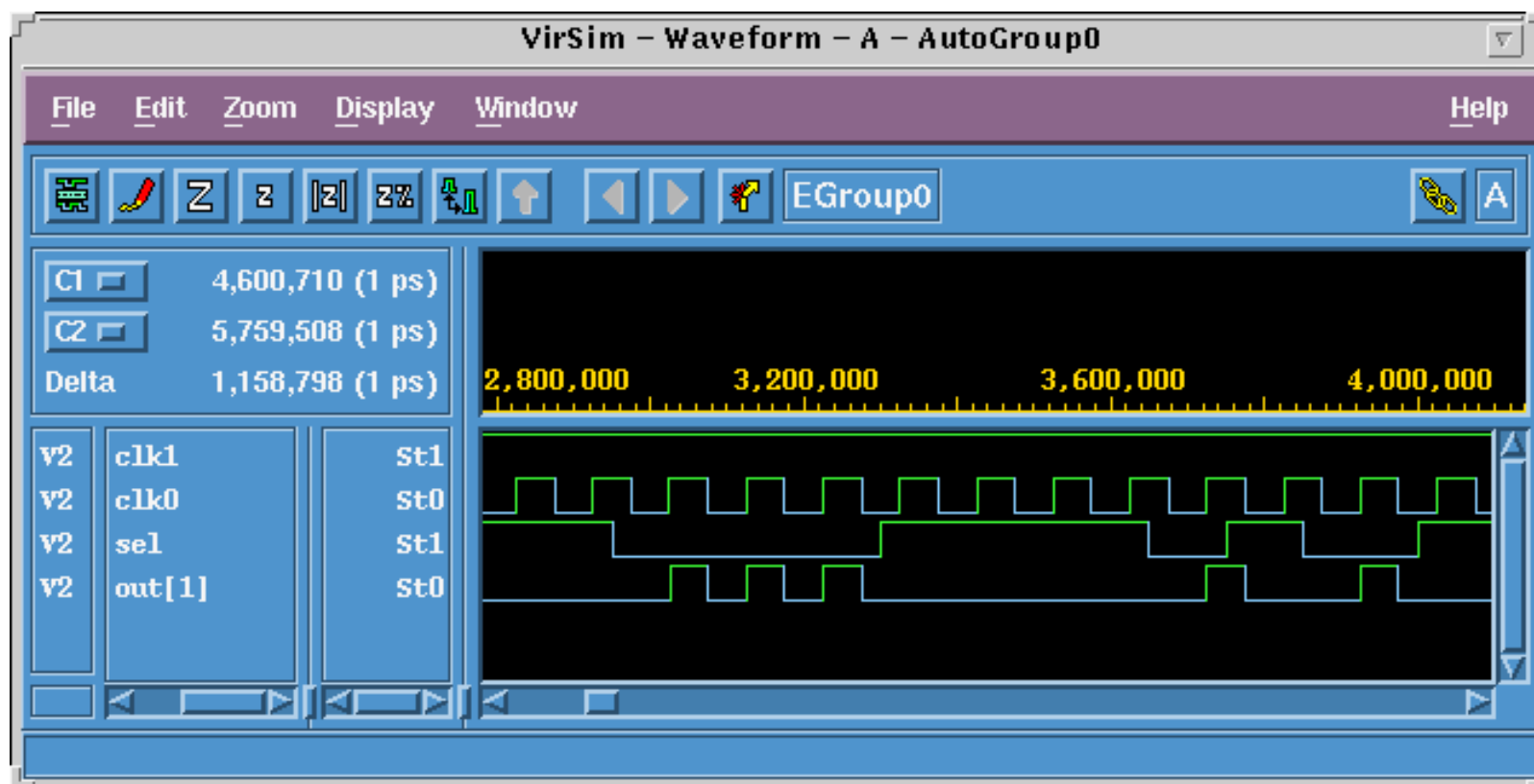
— Clock Multiplexer “Glitch-free”

- Switch between unrelated clocks
- BUFGMUX



No pulse width is ever shorter than 1/2 of the period

Device Type	Percentage
Smartphone	95%
Tablet	85%
Smartwatch	75%
Smart TV	65%
Smart Home Device	55%
Smart Car	45%



Switching from clk0 to clk1

