Designing with FPGAs

Beyond Bigger, Faster, Cheaper...

Peter Alfke
Xilinx, Inc.
peter.alfke@xilinx.com

XILINX®

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
- List of good URLs

XILINX®

1

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*

2

ASIC Problems

- **ASIC user must control design details:**
  - ***Fault coverage***
  - ***Clock-tree structure***
  - ***Second / third-order electrical effects***
    - *clock distribution delay and clock skew*
    - *glitch-free clock multiplexing and enable*
    - *cross-talk, hold-time issues*
- **FPGA user can concentrate on system design**
  - *Xilinx designers have solved the above issues*

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.

3

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*

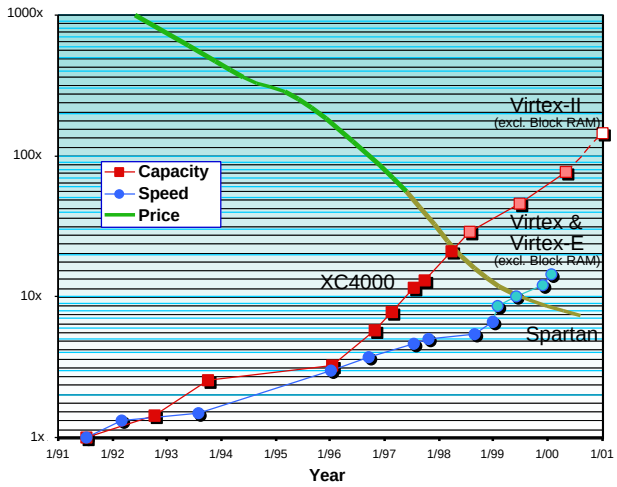
Bigger, Faster, Cheaper FPGAs

- **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

4

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*

5

FPGA Technology in Production

- **In step with the best microprocessors**
  - *0.15 micron - 8 layer metal - up to 6 million gates*
  - *Up to 420 MHz clock rate, 840 Mbps interface*
  - *Certified cores for PCI64 etc*

Virtex and Spartan II:	*0.25 micron*	*5-layer metal*
Virtex-E:	*0.18 micron*	*6-layer metal*
Virtex-II:	*0.15 micron*	*8-layer metal*

  - *Copper technology in 2000*
  - *Copper with low-k dielectric in 2001*

FPGA Architecture Today

- **Logic, RAM, arithmetic, abundant interconnect**
  - *up to 67,000 LUTs and flip-flops*
  - *up to 144 dual-ported 18K-bit BlockRAM*
  - *up to 144 multipliers (18 x 18 bits, < 7 ns)*
- **Versatile I/O**
  - *20 different standards, LVDS, LVPECL, etc.*
  - *Controlled impedance provides pc-board termination*
- **Clock management eliminates clock delay**
  - *and provides frequency synthesis and phase control*
- **Encrypted configuration provides security**
  - *Triple-DES encryption of the configuration bitstream*

6

Design Methodology

- **Design entry**
  - *VHDL/Verilog, schematic, verified cores*
- **Synthesis**
  - *optimized for FPGA architecture*
- **Timing-driven design**
  - *optimizes for the requested performance*
- **Place and route**
  - *significantly smarter and faster algorithms*
- **Team-based design**
- **Internet reconfiguration**

Wide Range of Users and Applications

Small devices, <150K gates	Large devices, up to 6M gates
Small systems	***Large Systems***
Small budget	***Existing EDA tools***
Short learning curve	***ASIC experience***
Schematic design	***VHDL/Verilog, cores***
Single designer	***Team-based design***
One-stop shopping	***Compatibility with existing tools***
Foundation *software*	Alliance *software*
Spartan *and* CPLD	Virtex and Virtex-II

7

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**

8

Performance Parameters II

Internal register-to-register	*Virtex-II-5*
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

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

9

Provide High-Level Floorplanning

Intelligent pin assignment

prevents routing congestion and poor performance

Natural structure:

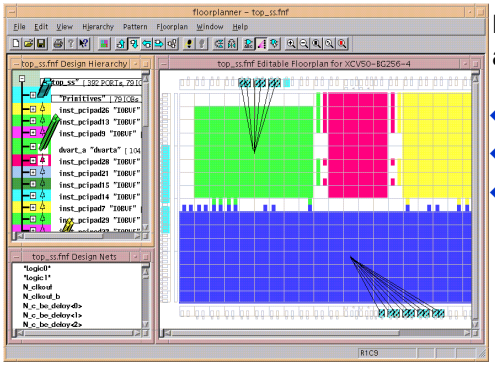
Data flows horizontally, Control flows vertically

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

High Level Floorplanning



High Level Floorplanner allows definition of:

- **Module area**
- **Location of device pins**
- **Location of module ports**

Reduces Compile Time

Increases Performance

10

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*

11

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 implemention,**
  - *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 is 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)*

13

Evolution

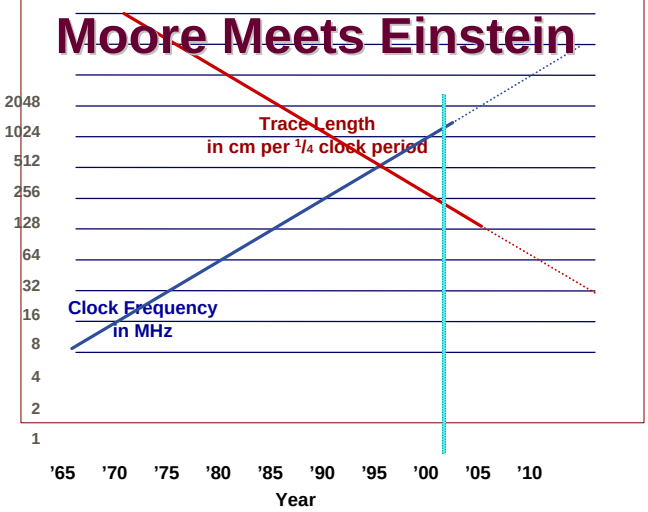
	1965	1980	**1995**	2010 (?)
Max Clock Rate (MHz)	1	10	**100**	1000
Min IC Geometry (µ)	-	5	**0.5**	0.05
Number of IC Metal Layers	1	2	**3**	10
PC Board Trace Width (µ)	2000	500	**100**	25
Number of Board Layers	1-2	2-4	**4-8**	8-16

Every 5 years:
System speed doubles, IC geometry shrinks 50%

Every 7-8 years: PC-board min trace width shrinks 50%

XILINX®

Moore Meets Einstein



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

XILINX®

14

Designing for Signal Integrity

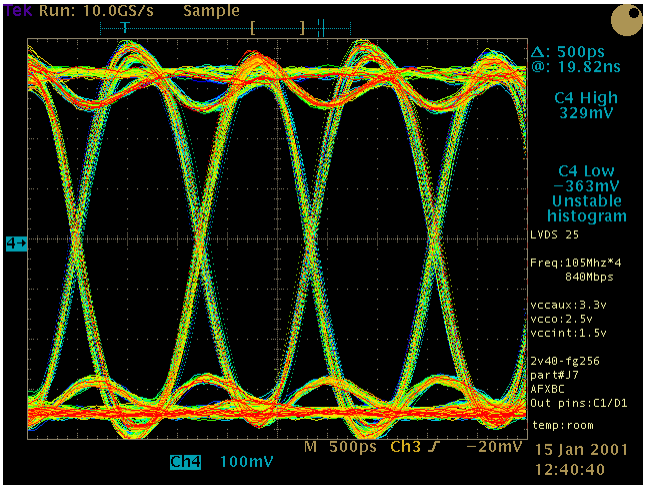
- **Devices need good Vcc bypassing**
  - *Bypass capacitor is the only source of dynamic current*
- **Output driver needs IBIS models**
  - *http://www.xilinx.com/support/troubleshoot/htm_index/sw_ibis.htm*
- **User needs understanding of transmission line effects**
  - *Characteristic impedance, reflections, 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*

15

840 Mbps Eye Pattern at Virtex-II LVDS Output



Power Supply Decoupling

- **CMOS current is dynamic**
  - *Icc 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*

16

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 Vcc = slow operation*
  - *Cold and high Vcc = 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"*

17

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*

XILINX®

Designing for Low Power

XILINX®

18

Designing for Low Power Consumption

- **To extend battery life**
- **To reduce chip temperature and cooling requirements**
  - *Tjmax = 125 degr.C (150 degre.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

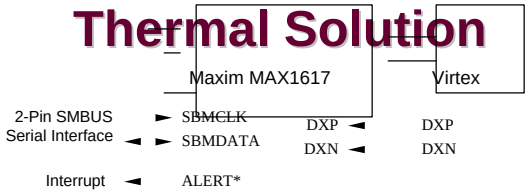
- **Clock Power + I/O Power + Logic Power**
- **Clock Power**
  - Minimize # of high-speed clock nets
  - Use DLLs for phase-aligned sub-clocks
  - CE does not reduce clock power
- **I/O power**
  - Avoid wasted current in input buffers
  - Use fast, full-swing input signals
  - Use output registers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

19

Low Logic Power

- **Control Vcc tightly**
  - *Power is proportional to Vcc²*
- **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*

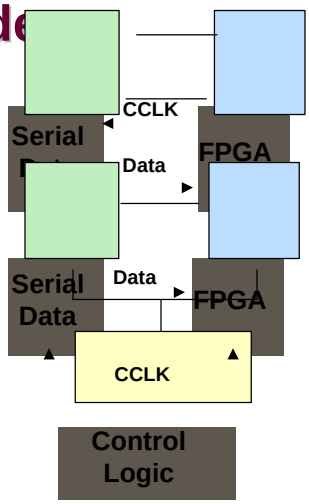
Precise thermal management is now easy

20

Configuration and Bitstream Security

Configuration Modes: Serial Mode

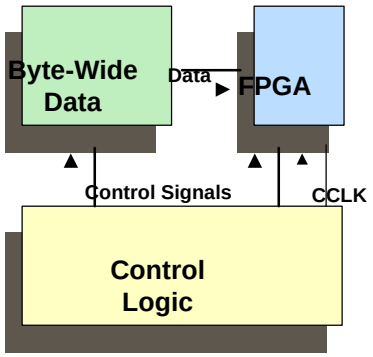
- **Data is loaded one bit per CCLK**
- **Master serial**
  - *FPGA drives configuration clock (CCLK)*
  - *FPGA provides all control logic*
  - *Note that CCLK is also an input !*
- **Slave serial**
  - *External control logic generates CCLK*
    - Microprocessor
    - Xilinx download cable
    - Another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



21

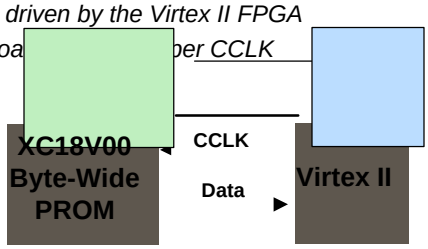
Configuration Modes: Byte-Wide SelectMAP Mode

- **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 loa per CCLK*

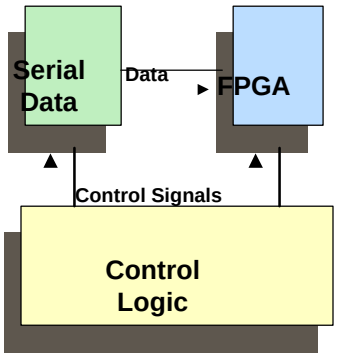


New to Virtex II
by popular demand...

22

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



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*
  - *Incoming bitstream is decrypted on-the-fly*
  - *Keys are kept alive by an external battery (1.0 to 3.6 V)*
  - *Extremely low current consumption, <100 nA*
  - *Battery must be changed while Vcc is alive*

23

Asynchronous Issues

Understanding Asynchronous Design Issues

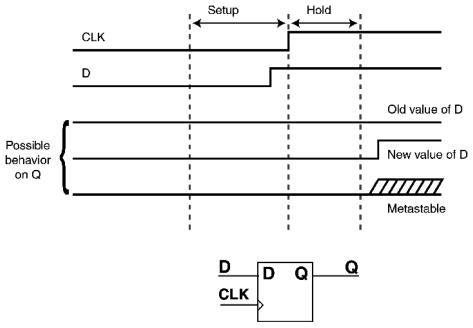
- **Most systems operate synchronously inside**
  - *But asynchronous inputs are a fact of life*
- **Occasionally, an asynchronous input will cause a flip-flop to go metastable**
  - *This is a rare, but unavoidable, probabilistic event*
- **Solution:**
  - *Faster flip-flops recover faster*
  - *Double-synchronization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

Awareness and understanding are crucial

24

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

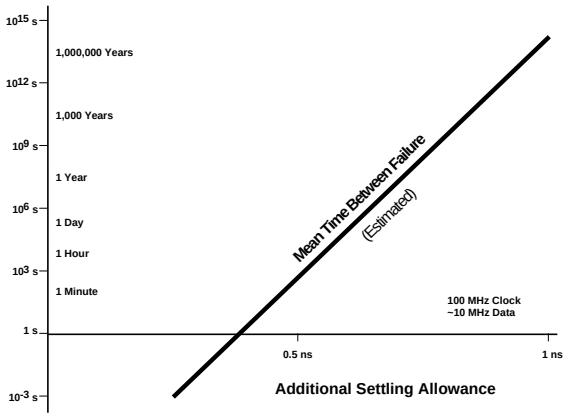
- **Caused by asynchronous data input**
  - *Violates set-up time requirement*
  - *Usually gets synchronized in the flip-flop without problem*
- **But if data changes within a tiny set-up time window**
  - *Then the flip-flop can go metastable*
  - *Resulting in unpredictable delay to reach stable 1 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*

25

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*

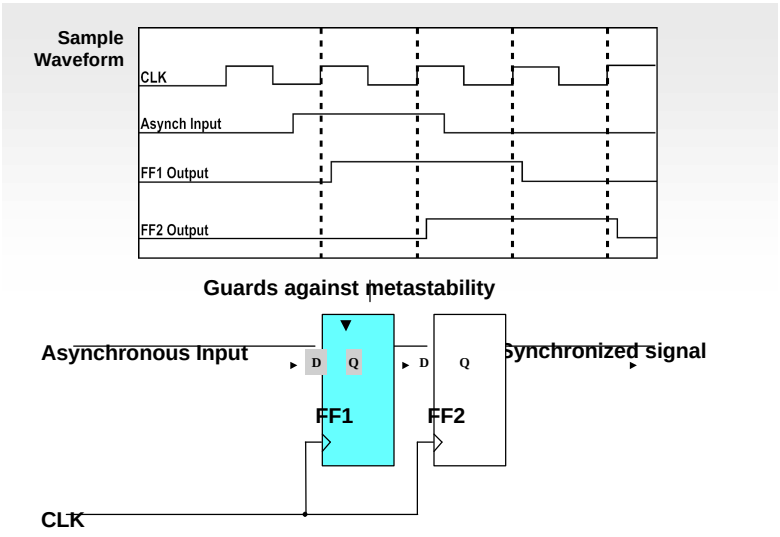
Metastability Data



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

26

Synchronization Circuit



XILINX®

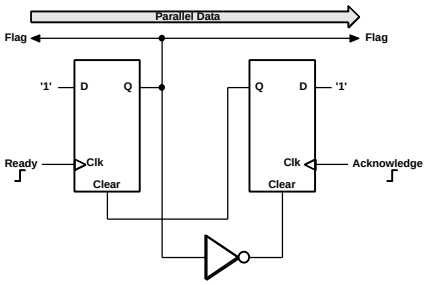
Moving Data Across Asynchronous Clock Boundaries

- **Worst-case timing happens, sooner or later**
- ***Murphy does not sleep !***
- **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*

XILINX®

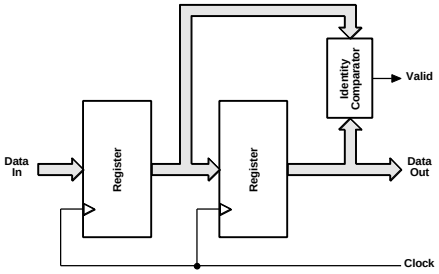
27

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**

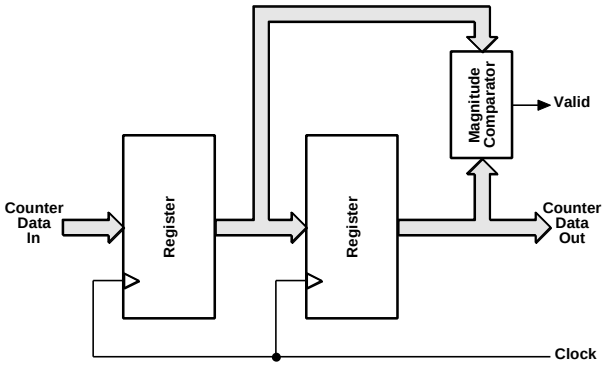
Moving Parallel Data without Handshake



- **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**

28

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

29

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

30

Full and Empty Control

Identity-compare write and read addresses

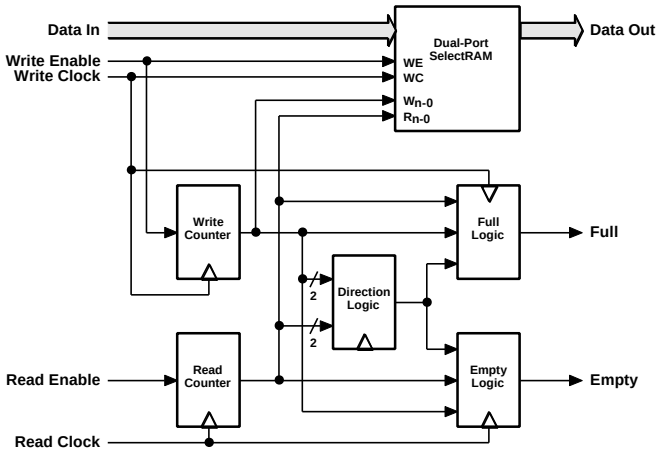
— *identical addresses mean either Full or Empty*

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



31

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*

32

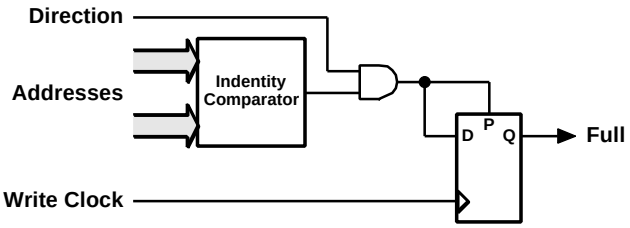
Synchronize to the Proper Clock

- **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.

33

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

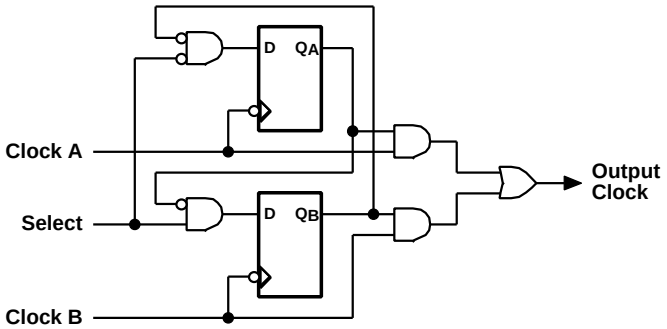
- **180 MHz** *asynchronous* **operation**
  - *4K deep, 1n bits wide*
  - *2048 deep, 2n bits wide*
  - *1024 deep, 4n bits wide*
  - *512 deep, 8n bits wide*
- **Uses n BlockRAMs plus 16 to 20 CLBs**
  - *BlockRAMs for data storage*
  - *CLBs for address counters, direction detection, EMPTY and FULL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

34

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



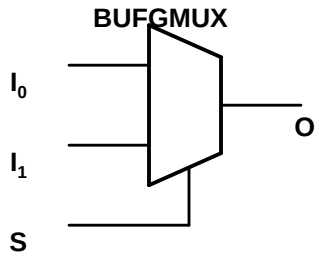
- **This circuit waits for the present clock to go Low**
  - *Output then stays low until the new clock is Low*

Guaranteed to switch glitch-free, no runt pulses

  - *http://www.xilinx.com/xcell/xl24/xl24_20.pdf*

35

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

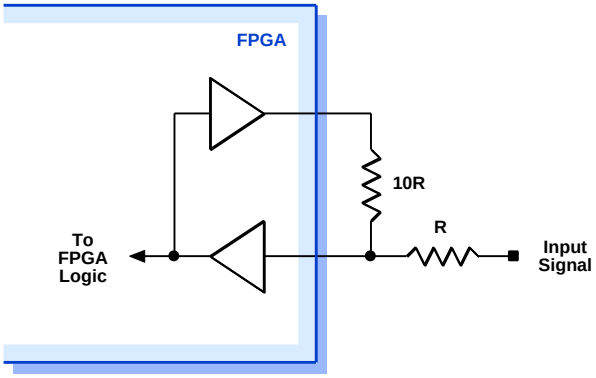
- **Asynchronous data transfer is dangerous**
  - *but not if you understand the issues and know how to design around them*
- **Clock gating is unhealthy**
  - *but not if you use smart circuits*
- **Metastabilty can hurt very badly**
  - *but only if inside a very tight timing budget*

Modern CMOS resolves very fast (within a few ns)

36

Tips and Tricks from the Xilinx Archives

Schmitt Trigger

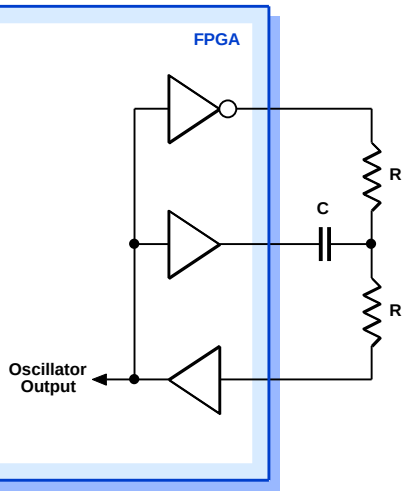


- Hysteresis = 10% of Vcc

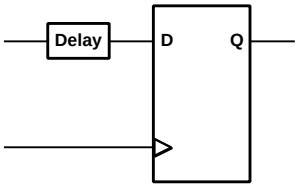
37

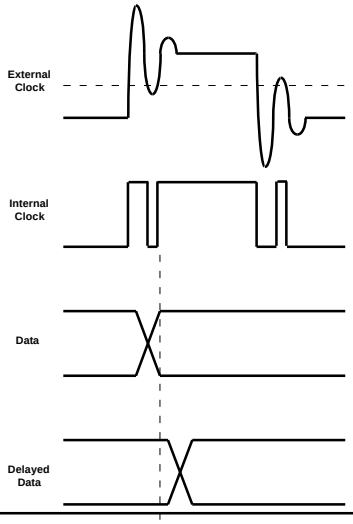
RC Oscillator

- **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

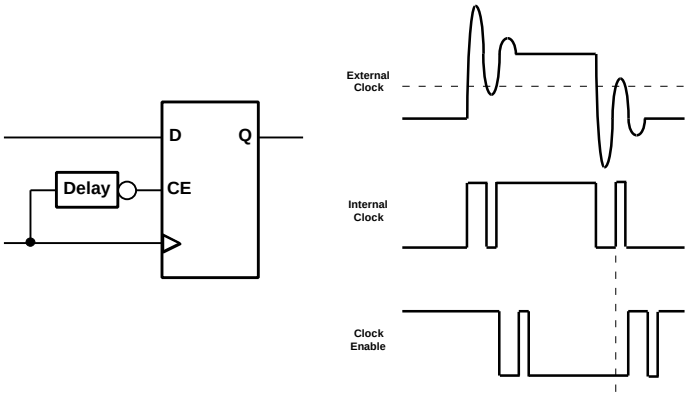




- **Problem: Double pulse on the active edge**
- **Solution: Delay D, to prevent the flip-flop from toggling soon again**

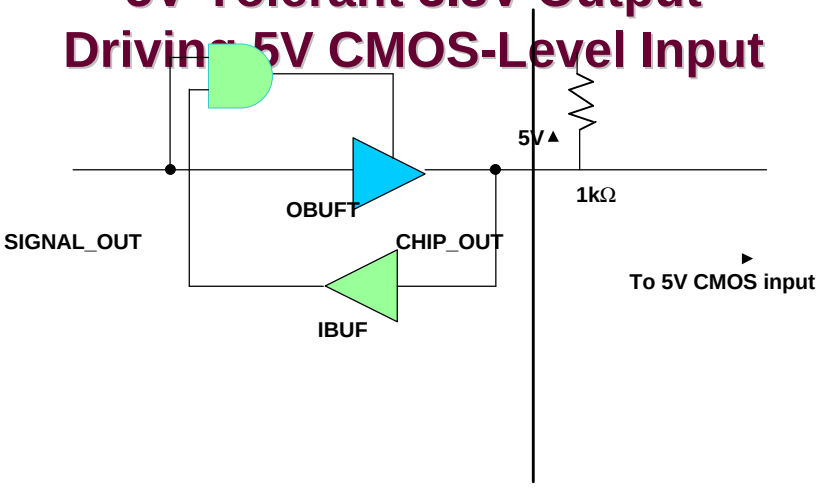
38

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



39

List of Good URLs

Xilinx:

- **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

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

40