

4/20/09

525.442 VHDL / FPGA Design

# Agenda

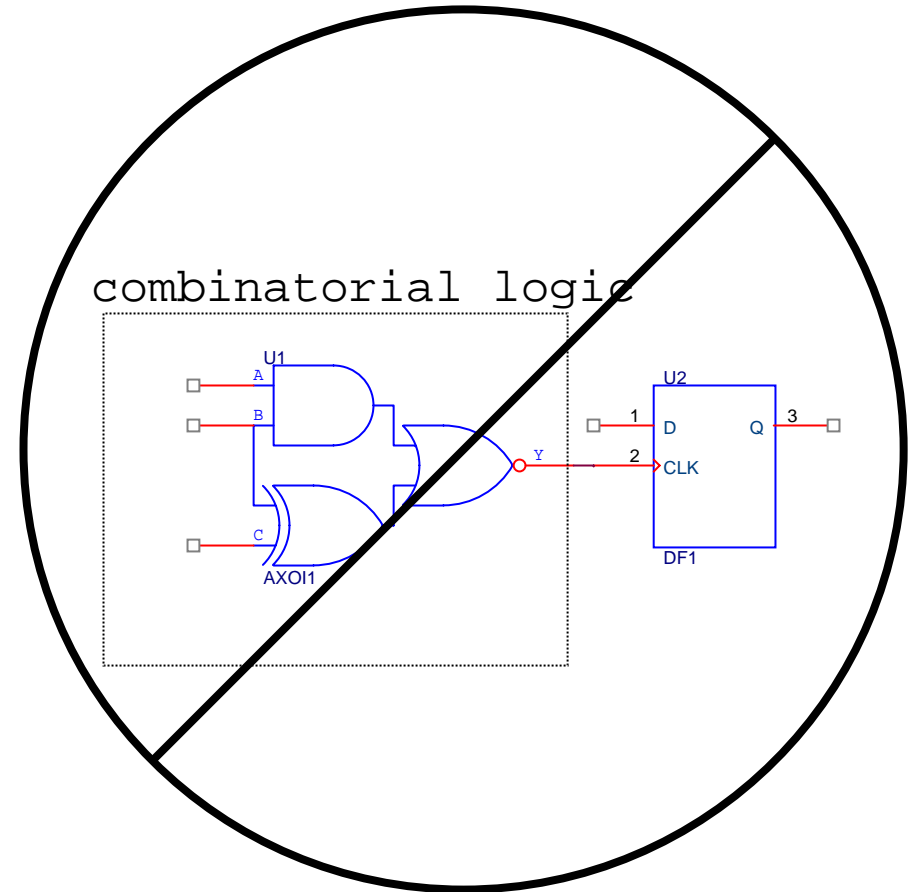
- DDS Solution
- Mini-topic
  - Timing Analysis
  - Timing Hazards
- Final Project Presentations
- DDS Demonstrations

# DDS Solution

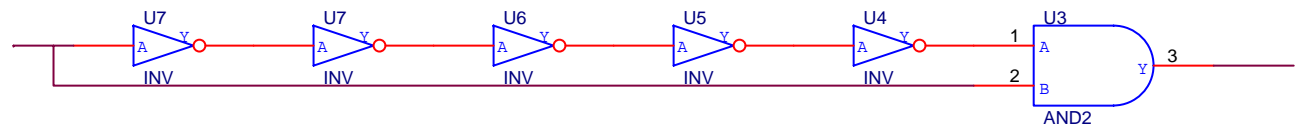
- Existing Macros used
  - Soft
    - Sine table (just a ROM)
    - DAC
    - 7 seg decoder
  - Hard
    - DCM\_SP

# (An Aside...)

unpredictable delays cause  
glitches, which are a problem  
on edge triggered logic

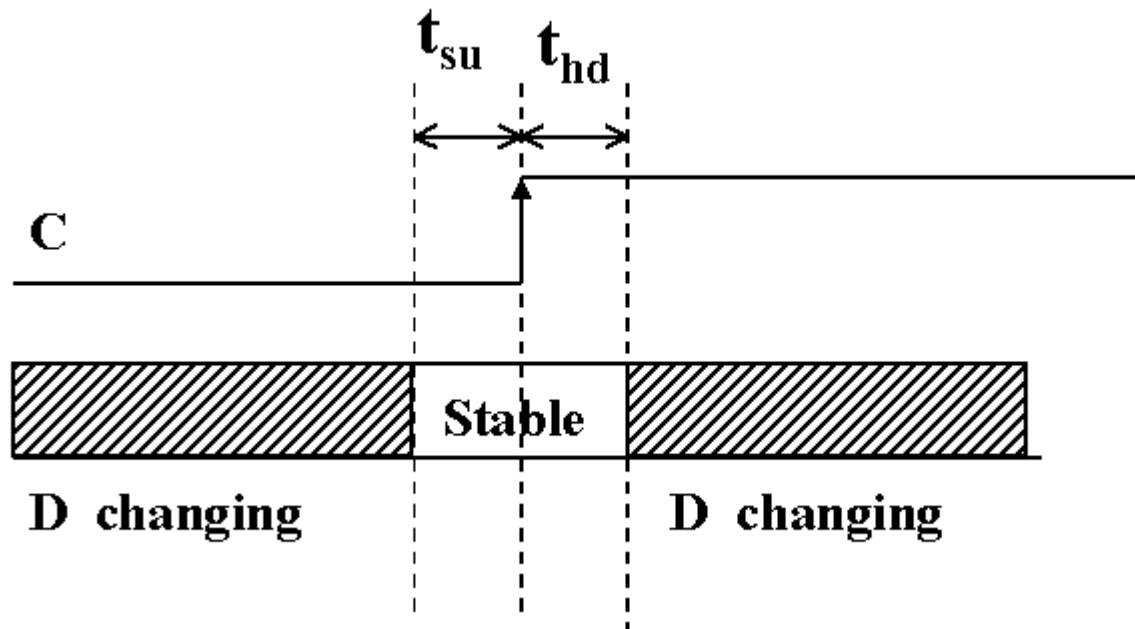


This is a BAD pulse generator  
very unpredictable in an FPGA



# Review: Setup / Hold Time

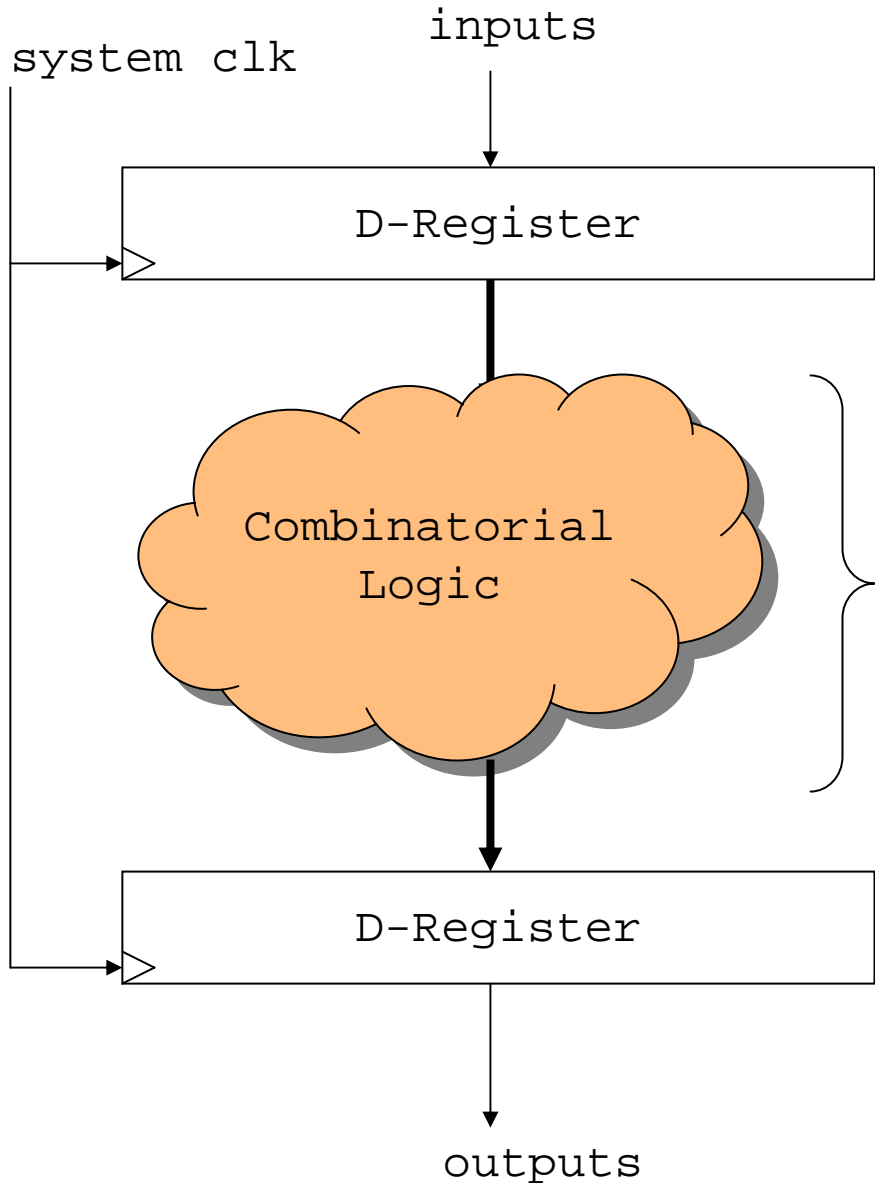
- Data on “D” of Flipflop must be present at “setup time” before clk edge, and remain on “D” for “hold time” after clk edge.



# Timing Issues

- Combinatorial Delay in Synchronous Design
- Asynchronous Inputs
  - Differing Clock Domains
  - Synchronizer
- Metastability
- Clock Skew

# Review: Combinatorial Delay



Combinatorial Logic introduces a **delay**, output of this logic could be moving around during this time

Time difference between time when D input is stable till time that D input is required to be stable is called **slack**, i.e.

$$\text{clk period} - \text{delay} = \text{slack}$$

# Timing Variables

- Temperature :
  - Propagation time through logic elements is *increased* with increasing temperature
- Supply Voltage
  - Propagation time through logic elements is *decreased* with increasing voltage
- Process Variation
  - Timing is affected by variations in the silicon, resistance, capacitance of routing grid..etc.

FASTEST : High Voltage, Low Temperature, BEST process

SLOWEST : Low Voltage, High Temperature, WORST process

# Static Timing Analysis

- Static Timing Analysis
  - analyzes delay through the various logic paths in the design
    - pad to pad
    - pad to register
    - register to register
    - register to pad
  - Gives a timing report to the designer :
    - what are the longest paths?
    - if constraints are specified, are they met? by how much?

# Static Timing Analysis

- Static Timing Analysis can be run at various phases in the design process
  - after translate, map, or P & R
- Benefits
  - no test vector creation : static timing analysis does not depend on exercising the FPGA's functions in simulation – all paths are covered
    - fast, easy
- Subtle issues are not handled, and some problems are not apparent.
  - simple 1-clock designs where the question is : “How fast can this run?” are easily handled by STA.
  - multiple clock domains require some thought. However, if division is clear and managed, its just a matter of doing two separate timing analyses.

# STA in Webpack *(similar in all design platforms)*

1<sup>st</sup> level Done Automatically in P&R

**Initializing temperature to 85.000 Celsius.** (default - Range: -40.000 to 100.000 Celsius)

**Initializing voltage to 1.140 Volts.** (default - Range: 1.140 to 1.320 Volts)

**INFO:Par:282 - No user timing constraints were detected** or you have set the option to ignore timing constraints ("par

-x"). Place and Route will run in "Performance Evaluation Mode" to automatically improve the performance of all

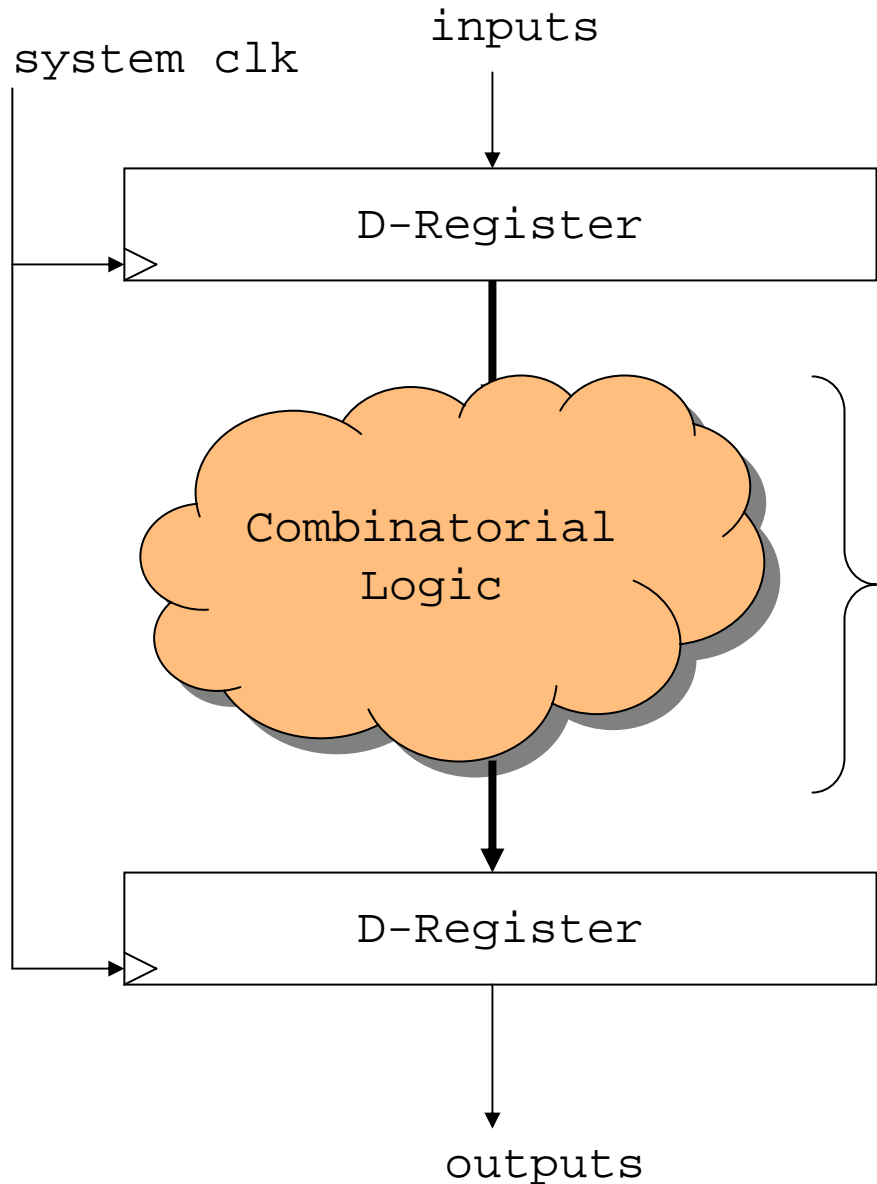
internal clocks in this design. The PAR timing summary will list the performance achieved for each clock. Note: For

the fastest runtime, set the effort level to "std". For best performance, set the effort level to "high". For a

balance between the fastest runtime and best performance, set the effort level to "med".

Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
Autotimespec constraint for clock net dcm_clk0	SETUP HOLD	N/A 1.175ns	4.684ns	N/A 0	0 0
Autotimespec constraint for clock net clk100	SETUP HOLD	N/A 1.276ns	8.762ns	N/A 0	0 0

# STA with Registers



by default, STA tool will assume that data needs to be ready by next clock

Delay

False paths, can complicate STA.

Telling the analyzer about false, or multi-cycle paths is different for every tool.

# Setting Constraints

The screenshot shows the Xilinx ISE Timing Constraints editor. The main window displays a table with the following data:

Clock Net Name	Period	Pad to Setup	Clock to Pad
clk50_in	20 ns. HIGH 50%		

Below the table, there is a text prompt: "Double-click on ucf file:"

The interface also shows the following panels and options:

- Sources:** Constraint Files: top.ucf; Show Constraints from Specified File only (unselected); Show Constraints from All Files (selected).
- Constraint Type:** Timing Constraints (selected), Global, Ports, Advanced.
- Processes:** Processes for: top.ucf; Add Existing Source, Create New Source, User Constraints, Edit Constraints (Text).

# Setting Constraints

```
NET "switches<5>" LOC = "L13" ;
NET "switches<6>" LOC = "N17" ;
NET "switches<7>" LOC = "R17" ;
NET "DACout" LOC = "L15" | IOSTANDARD="LVTTTL" | DRIVE=16;
NET "clk50_in" TNM_NET = clk50_in;
TIMESPEC TS_clk50_in = PERIOD "clk50_in" 20 ns HIGH 50%;
```

# STA with constraints (cont)

Derived Constraints for TS\_clk50\_in

Constraint	Period Requirement	Actual Period		Timing Errors		Paths Analyzed	
		Direct	Derivative	Direct	Derivative	Direct	Derivative
TS_clk50_in	20.000ns	N/A	19.850ns	0	0	0	4593
TS_clk1001	10.000ns	9.925ns	N/A	0	0	4367	0
TS_dcm_clk01	20.000ns	5.367ns	N/A	0	0	226	0

All constraints were met.

Derived Constraints for TS\_clk50\_in

Constraint	Period Requirement	Actual Period		Timing Errors		Paths Analyzed	
		Direct	Derivative	Direct	Derivative	Direct	Derivative
TS_clk50_in	15.000ns	N/A	16.028ns	0	5	0	4593
TS_clk1001	7.500ns	8.014ns	N/A	5	0	4367	0
TS_dcm_clk01	15.000ns	5.097ns	N/A	0	0	226	0

1 constraint not met.

Minimum period is 8.014ns.

**Slack: -0.514ns (requirement - (data path - clock path skew + uncertainty))**

**Source:** [theta\\_1](#) (FF) **clk:** clk100 rising at 0.000ns

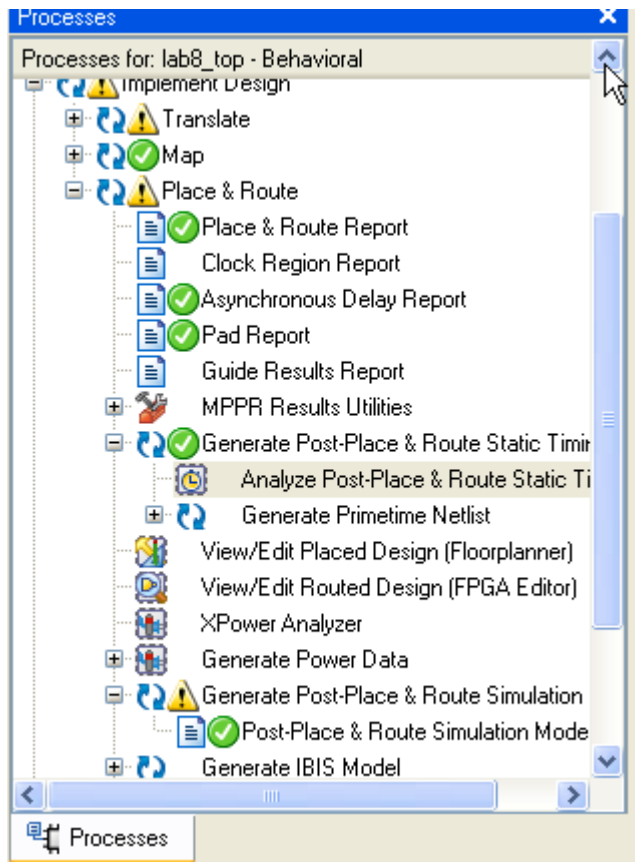
**Destination:** [DAC\\_inst/sigmaLatch\\_11](#) (FF) **clk:** clk100 rising at 7.500ns

Requirement	Data Path Delay	Clock Path Skew:	Clock Uncertainty
7.500ns	8.014ns (Levels of Logic = 8)	0.000ns	0.000ns

[Timing Improvement Wizard](#)

[Maximum Data Path: theta\\_1 to DAC\\_inst/sigmaLatch\\_11](#)

Delay type	Delay(ns)	Logical Resource
Tcko	0.592	<a href="#">theta_1</a>
net (fanout=12)	0.727	<a href="#">theta&lt;1&gt;</a>
Tilo	0.759	<a href="#">lut/BU15</a>
net (fanout=4)	0.348	<a href="#">sine&lt;6&gt;</a>
Tilo	0.759	<a href="#">Sh131_SW0</a>
net (fanout=2)	0.435	<a href="#">N101</a>
Tilo	0.759	<a href="#">Sh131</a>
net (fanout=1)	0.023	<a href="#">Sh131_I0</a>
Tilo	0.759	<a href="#">Sh1360</a>
net (fanout=2)	0.453	<a href="#">Sh13</a>
Topcyf	1.162	<a href="#">DAC_inst/Madd_sigmaAdder_Madd_lut&lt;4&gt;</a> <a href="#">DAC_inst/Madd_sigmaAdder_Madd_cy&lt;4&gt;</a> <a href="#">DAC_inst/Madd_sigmaAdder_Madd_cy&lt;5&gt;</a>
net (fanout=1)	0.000	<a href="#">DAC_inst/Madd_sigmaAdder_Madd_cy&lt;5&gt;</a>



```
sine_unsigned <= unsigned(not sine(9) & sine(8 downto 0)) when rising_edge(clk100);
```

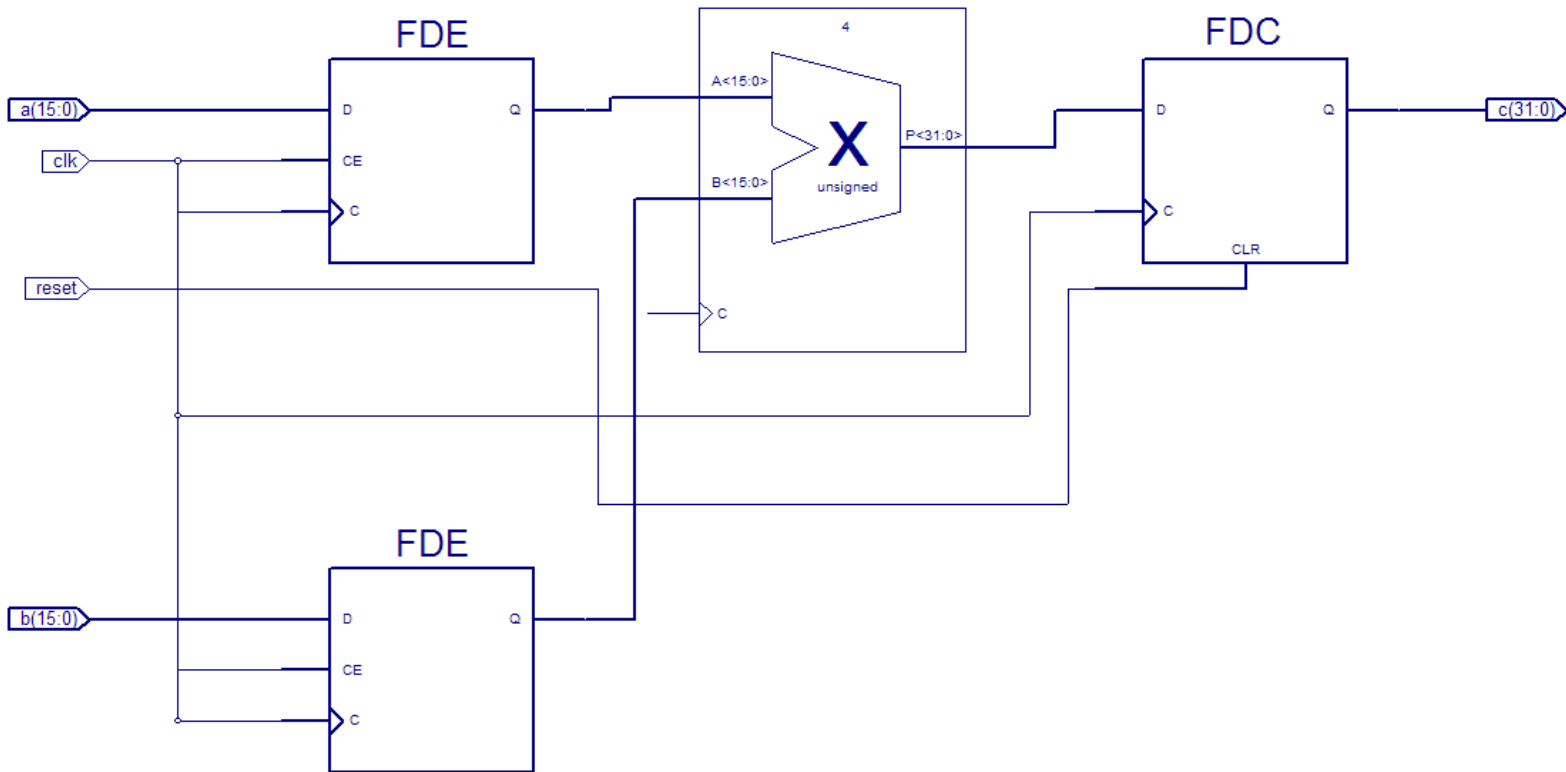
Derived Constraints for TS\_clk50\_in

Constraint	Period Requirement	Actual Period		Timing Errors		Paths Analyzed	
		Direct	Derivative	Direct	Derivative	Direct	Derivative
TS_clk50_in	15.000ns	N/A	14.268ns	0	0	0	1890
TS_clk1001	7.500ns	7.134ns	N/A	0	0	1640	0
TS_dcm_clk01	15.000ns	5.693ns	N/A	0	0	250	0

All constraints were met.

# Multiply Timing Example

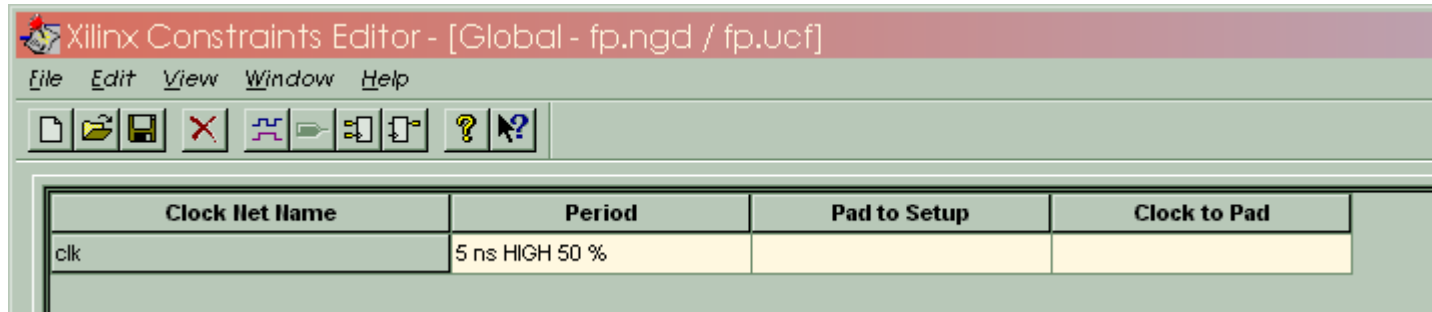
Let's call this a multi-cycle path



# Multiply Example

- Built
- Max clock ~20.5 ns
- Set constraint to 18 ns, 50% high
- Met by re-routing
- Set constraint to 12 ns, 50% high
- 28 setup errors
- Set multipath
  - TIMESPEC “TS\_MULT” = FROM “INPUT\_REGS” to “outputs” “TS\_CLK” \*3;

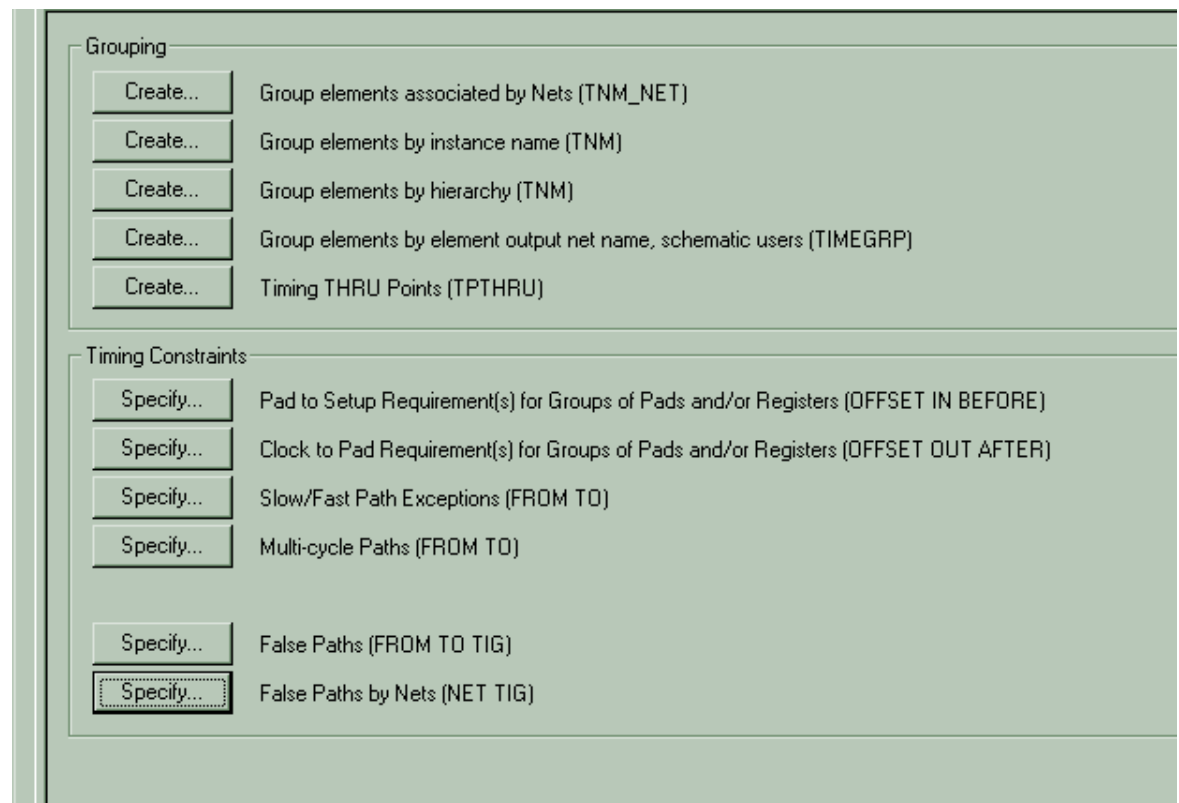
# Setting Constraints



Xilinx Constraints Editor - [Global - fp.ngd / fp.ucf]

File Edit View Window Help

Clock Net Name	Period	Pad to Setup	Clock to Pad
clk	5 ns HIGH 50 %		



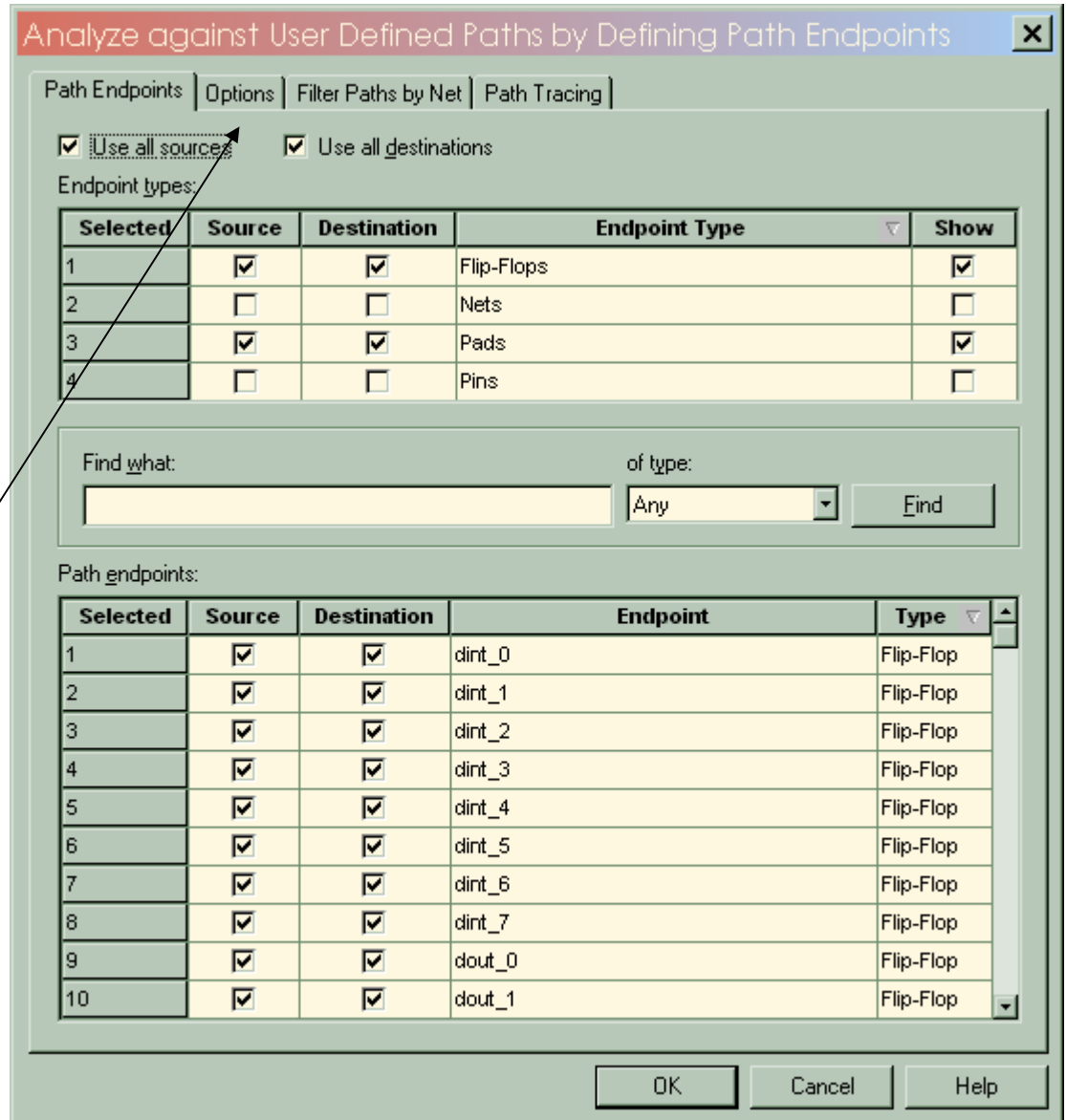
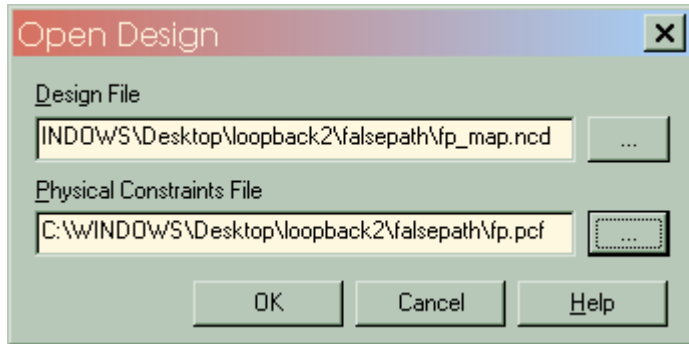
Grouping

- Create... Group elements associated by Nets (TNM\_NET)
- Create... Group elements by instance name (TNM)
- Create... Group elements by hierarchy (TNM)
- Create... Group elements by element output net name, schematic users (TIMEGRP)
- Create... Timing THRU Points (TPTHRU)

Timing Constraints

- Specify... Pad to Setup Requirement(s) for Groups of Pads and/or Registers (OFFSET IN BEFORE)
- Specify... Clock to Pad Requirement(s) for Groups of Pads and/or Registers (OFFSET OUT AFTER)
- Specify... Slow/Fast Path Exceptions (FROM TO)
- Specify... Multi-cycle Paths (FROM TO)
- Specify... False Paths (FROM TO TIG)
- Specify... False Paths by Nets (NET TIG)

# Using Timing Analyzer



to set limits on how many paths to report...etc.

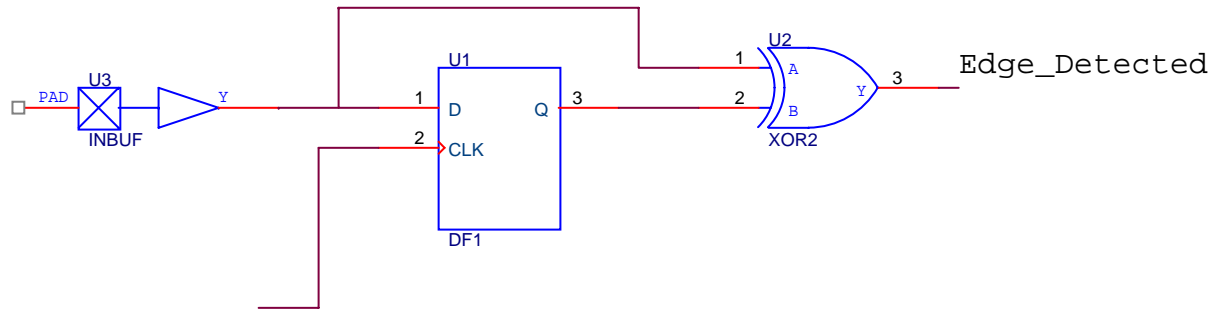
# Constraints Setting

- Set overall clock requirement
- After P+R, if constraints can not be met, examine the report to discover what the problem is
  - multi-cycle paths, or false paths can be ignored
    - Tell it to ignore, and redo analysis
    - or, run STA on what you know to be longest paths
- design can be modified to not have the multi-cycle paths (I.e. change design to include registers between sections...etc.)

# Timing Issues

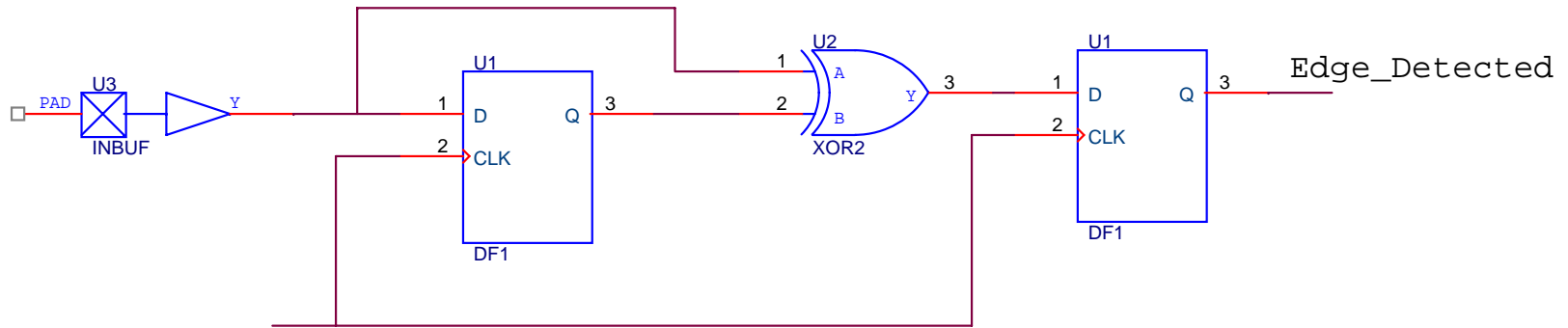
- Combinatorial Delay in Synchronous Design
- **Asynchronous Inputs**
  - Differing Clock Domains
  - Synchronizer
- Metastability
- Clock Skew

# Timing Hazards



If this is used as the control input to a state machine, what could happen?

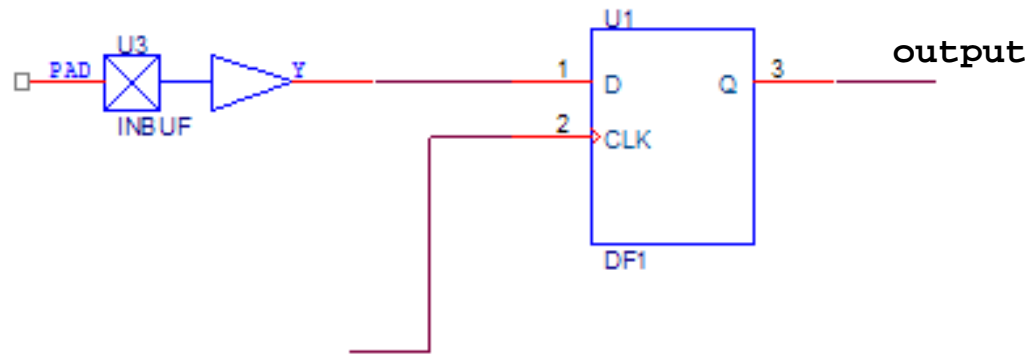
# Timing Hazards (2)



If this is used as the control input to a state machine, could we get in an illegal state?

Is there a timing hazard here? If so, what is it?

# Timing Hazards (3) : synchronizing asynchronous inputs



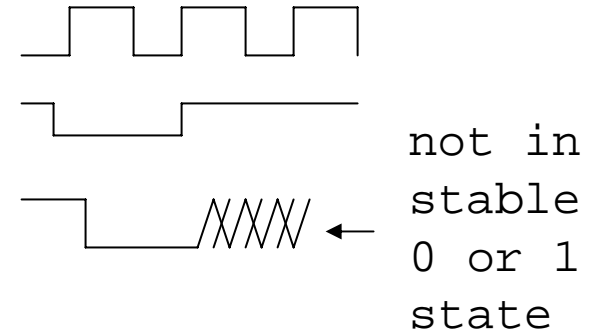
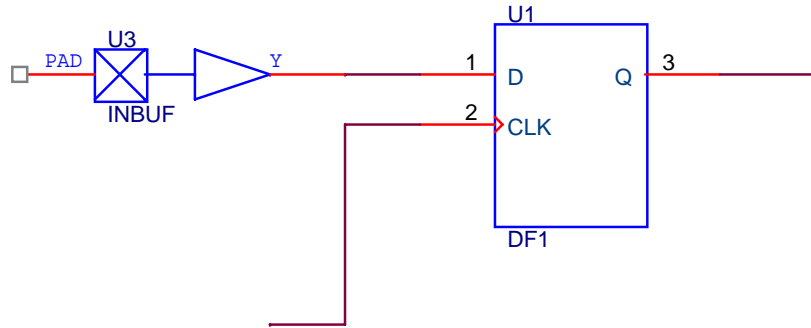
What if the signal **output** went to 200 destination registers? Any additional problems?

(Synthesizer may replicate flipflop)

# Summary

- Given synchronous design, we must guarantee:
  - delay of combinatorial logic between flip flops must be small enough to:
    - get there in time for next clock
    - not violate setup/hold time requirements for the ff.
    - Get to all destination clocks on the same cycle
  - asynchronous inputs WILL violate the setup/hold time of a ff, so we must manage that
    - async input only goes one place
    - Need to double register if output goes lots of other places

# Metastability : see Xapp094

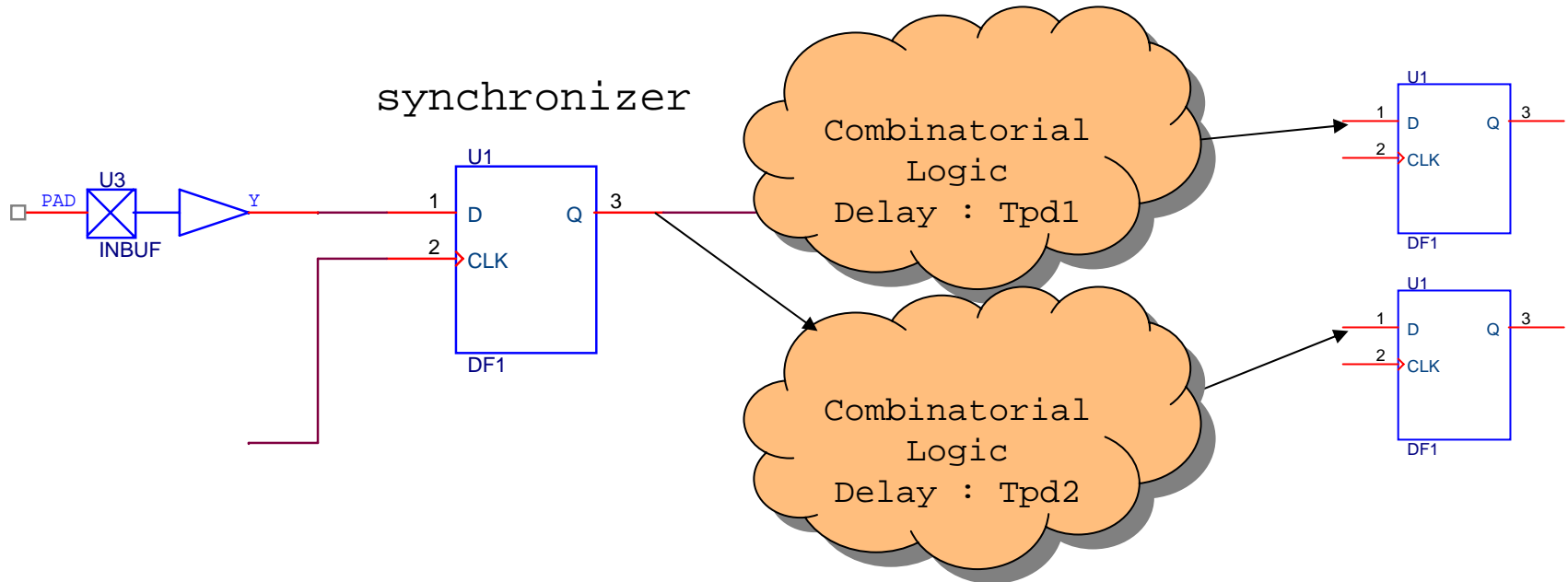


when transition on D occurs NOT ONLY within the setup/hold window, but within the much smaller window of time where the flop "accepts" new data, flipflop may be in an illegal state for **indefinite** period of time

Could theoretically be infinite amount of time before state resolves to legal state. Practically there is enough noise to push it one way or the other.

Still, resolution time will be variable, and events with longer resolution times are possible, but their likelihood decreases exponentially with time.

# Metastability



Under what conditions is this a problem?

# Metastability Quantified

$$\text{MTBF} = e^{(K2*t)} / ( K1 \times F_{\text{clock}} \times F_{\text{data}} )$$

t = slack time available for settling,

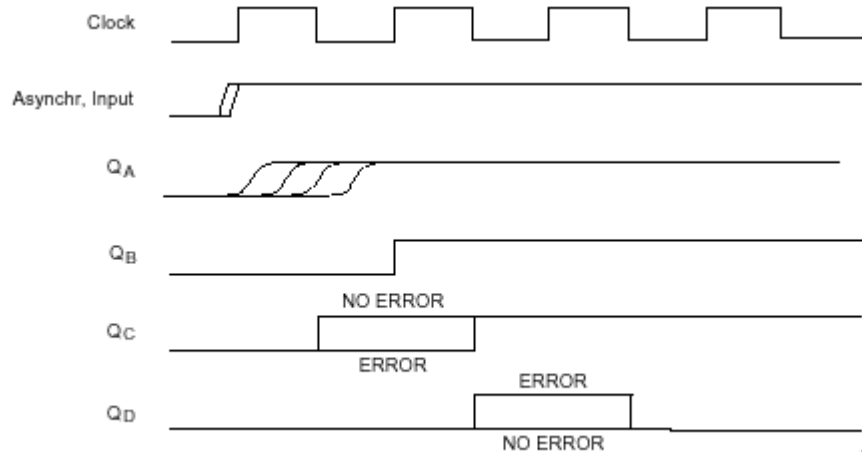
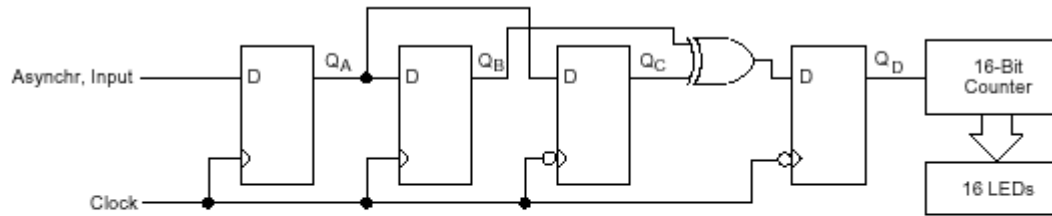
K1 and K2 are flipflop constants:

K1 = size of window

K2 = time to get out of metastable state

Fclock and Fdata are the frequency of the synchronizing clock and asynchronous data.

# Xilinx Metastability Test Ckt.



Test Circuit and Timing Diagram

X5985

# Xilinx Metastability Params

$$\text{MTBF} = e^{(K2*t)} / (K1 \times F_{\text{clock}} \times F_{\text{data}})$$

XC4000 : K1 = .1ns, K2 = approx 10/ns

so : assume Fclock = 50Mhz, Fdata = 1 MHz

MTBF =  $e^{10t/5000}$  where t = slack in ns

1ns of extra slack = MTBF of 4.4 seconds

2ns of extra slack = 97000 seconds = 26 hours

3ns of extra slack = 67 years

500 ps extra delay increases MTBF by a factor of  $199.6 \exp(500/212) = 266\,000$

1000 ps extra delay increases MTBF by a factor of  $199.6 \exp(1000/212) = 7 \times 10 \exp 10$

2000 ps extra delay increases MTBF by a factor of  $199.6 \exp(2000/212) = 5 \times 10 \exp 21$

# Timing Simulation

- After synthesis and/or map and P&R
- Tools create a VHDL model based on the synthesis output
  - Consists of raw technology dependent modules which are in vendor supplied library files
  - Synthesis / functional simulation mismatches appear here
- Tools also create a file with the delays under certain conditions through each of those aforementioned modules.

# Example of Post-Map sim. file

```
I_c : X_LUT4
generic map(
  INIT => X"000F"
)
port map (
  ADR0 => VCC,
  ADR1 => VCC,
  ADR2 => a_IBUF,
  ADR3 => b_IBUF,
  O => c_OBUF_GROM
);
c_OBUF_YUSED : X_BUF
port map (
  I => c_OBUF_GROM,
  O => c_OBUF
);
NlwInverterBlock_c_OUTBUF_GTS_TRI_CTL : X_INV
port map (
  I => GTS,
  O => NlwInverterSignal_c_OUTBUF_GTS_TRI_CTL
);
```

Structure and code from behavioral Model (the one you wrote) is gone – Though some signal names Will be preserved.

This code is for modeling only

# Simulating the structural file

In Modelsim :

- 1) Quit any simulation processes running  
**quit -sim**
- 2) Compile the structural model  
**vcom alu\_16c54\_timesim.vhd**
- 3) Simulate :  
**vsim alu\_16c54**  
**view \***

*You will explicitly have to put things on the wave window ..etc.*

Structure and code from behavioral Model (the one you wrote) is gone –  
Though some signal names  
Will be preserved.


This code is for modeling only

# Standard Delay File (SDF)

- “.SDF” file contains delays through modules under typical, minimum, and maximum delay conditions

```
(DELAYFILE
  (SDFVERSION "2.1")
  (DESIGN "norgate")
  (DATE "[Mon Jun 10 23:59:35 2002] ")
  (VENDOR "Xilinx")
  (PROGRAM "Xilinx SDF Writer")
  (VERSION "E.38")
  (DIVIDER /)
  (VOLTAGE 2.375:2.375:2.375)
  (PROCESS "best=1.0:nom=1.0:worst=1.0")
  (TEMPERATURE 85:85:85)
  (TIMESCALE 1 ps)
  (CELL (CELLTYPE "X_BUF")
    (INSTANCE a_IBUF_2)
    (DELAY
      (ABSOLUTE
        (IOPATH I O (773:773:773)(773:773:773))
      )))
```

Example From Webpack  
ONLY simulates the worst  
Case conditions



# Simulating with sdf timing info

In Modelsim :

1) Quit any simulation processes running  
**quit -sim**

2) Compile the structural model  
**vcom alu\_16c54\_timesim.vhd**

3) Simulate :

**vsim -sdftyp /=alu\_16c54\_timesim.sdf alu\_16c54  
view \***

*You will explicitly have to put things on  
the wave window ..etc.*

Structure and code from behavioral Model (the one you wrote) is gone –  
Though some signal names  
Will be preserved.

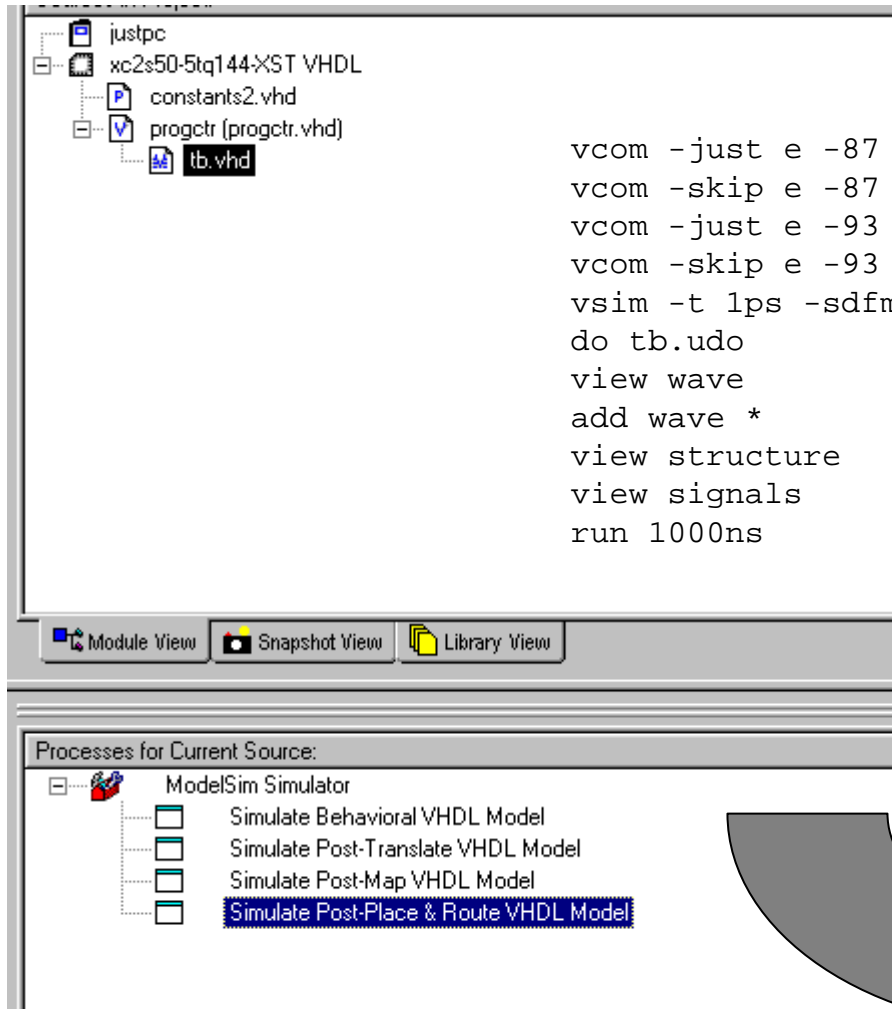
This code is for modeling only

sdftyp, sdfmax, sdfmin are accepted choices

/ = sdffilename (this means root level is associated with that sdf file)

# Webpack automation

IF A TESTBENCH IS CREATED :



The screenshot shows the ModelSim software interface. On the left, a project tree displays a directory structure for 'justpc' containing 'xc2s50-5tq144-XST VHDL', 'constants2.vhd', 'progctr (progctr.vhd)', and 'tb.vhd'. The 'tb.vhd' file is selected. In the center, a command window contains the following text:

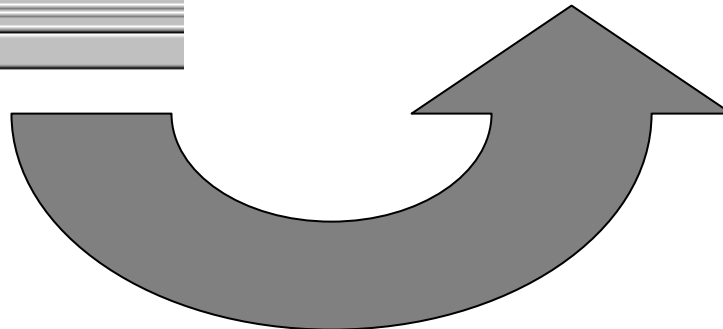
```
vcom -just e -87 -explicit -work work progctr_timesim.vhd
vcom -skip e -87 -explicit -work work progctr_timesim.vhd
vcom -just e -93 -explicit -work work tb.vhd
vcom -skip e -93 -explicit -work work tb.vhd
vsim -t lps -sdfmax /UUT=progctr_timesim.sdf -lib work testbench
do tb.udo
view wave
add wave *
view structure
view signals
run 1000ns
```

At the bottom of the interface, there are three tabs: 'Module View', 'Snapshot View', and 'Library View'. Below these tabs, a section titled 'Processes for Current Source:' lists four simulation options under 'ModelSim Simulator':

- Simulate Behavioral VHDL Model
- Simulate Post-Translate VHDL Model
- Simulate Post-Map VHDL Model
- Simulate Post-Place & Route VHDL Model

The 'Simulate Post-Place & Route VHDL Model' option is highlighted with a blue selection bar.

**script file created**

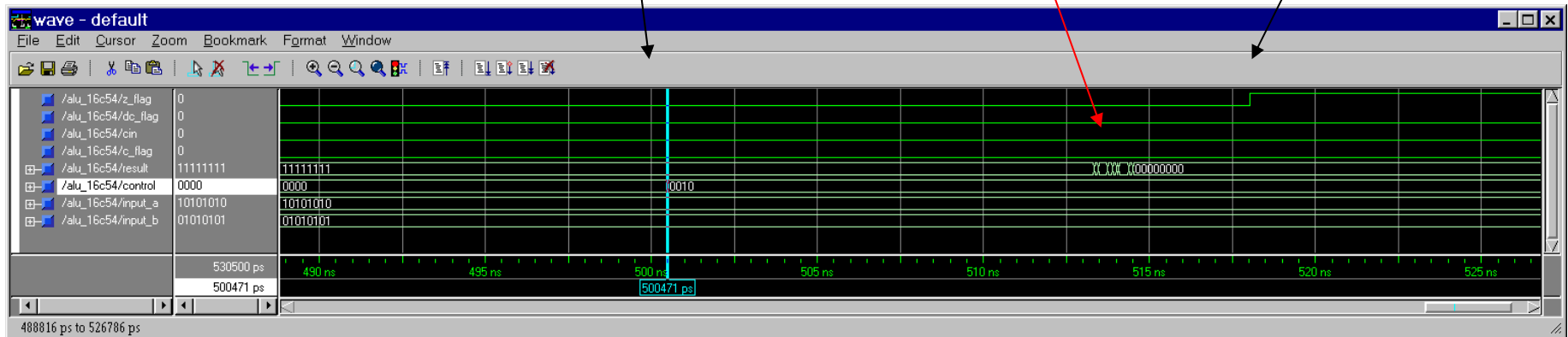


# Timing Simulation

ALU\_OP changes from  
in OR to AND

result bits are changing  
at different times

slow path is  
Z-flag



Summary for those reading the notes: all the detail of the SDF File and what it looks like is for your information. ISE Automates the processing of this such that you can just Simulate the timing model with your testbench.

# Pros / Cons

- Timing Simulation has the capability to catch anything that could go wrong on the real circuit
- In order to catch behavior you must simulate it
  - slow
  - time consuming to generate the test stimulation
- For large designs, full coverage becomes impossible with manual tools
- Use for potentially problematic areas, with Static timing analysis used for coverage.