XILINX ALL PROGRAMMABLE™

XILINX

A GENERATION AHEAD
SEMINAR SERIES

TCL, the Tool Control Language, and Vivado

VIVADO Productivity. Multiplied.

Agenda

- Expectations
- Controlling Vivado with TCL
- Vivado Physical Constraints
- Vivado Timing Constraints
- More Information

I. Expectations

Expectations for the Next 44 Minutes

- **There is NO WAY I can teach you the entire TCL language and how to use it with Vivado in 43 minutes**
- **It is absolutely possible to expose you to *enough* TCL in 42 minutes such that you'll get it**
  - And this will be done in the context of using Vivado
- **For now don't think of TCL as a language. Think of it as a bunch of similarly structured commands.**
  - As you write more powerful scripts, and you find that you need more power and flexibility in your scripts, TCL enables that

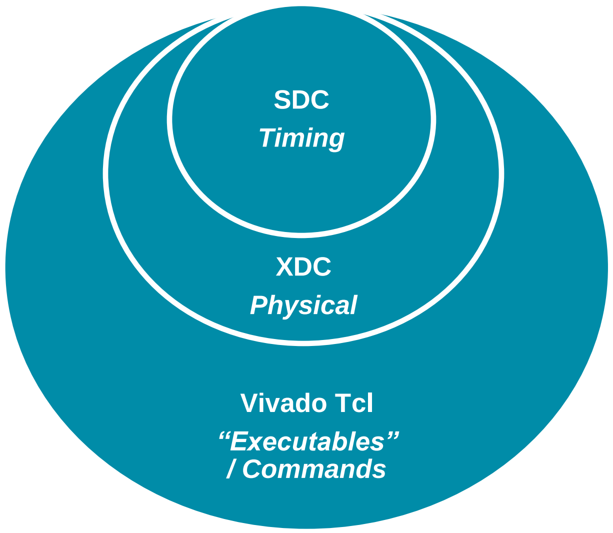
Homework: Get a TCL book, explore TCL on the WEB, play in Vivado

II. Controlling Vivado with TCL

Vivado Tcl

Include in .xdc file

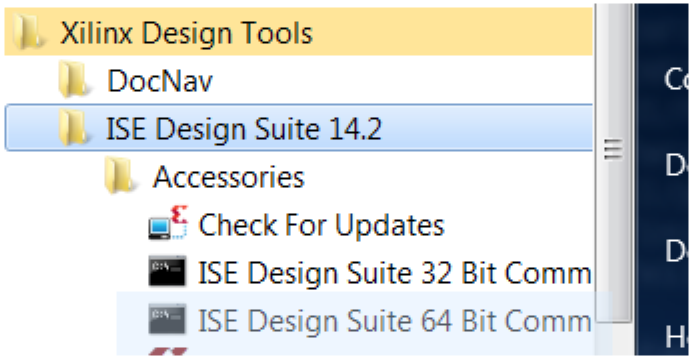
- **SDC - Synopsys Design Constraints**
  - Timing Constraints
- **XDC - Xilinx Design Constraints**
  - Physical Constraints
- **Vivado Tcl**
  - SDC + XDC
  - Xilinx commands
    - Project management, synthesize, place and route, compilation, report,…
  - Xilinx objects
    - Netlist, device, timing, project,…
  - General Tcl (8.5)
    - List-related commands are overloaded to support primary objects
    - Primary objects can directly be used with general commands



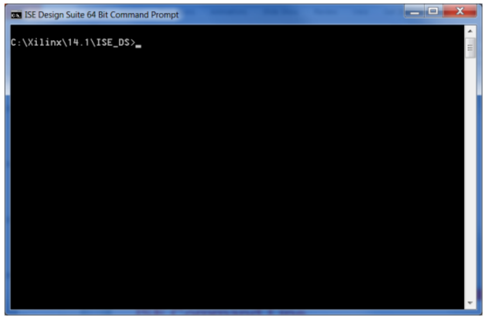
Things that used to go in a UCF are now TCL commands and are now included in a .xdc constraints file

Things that used to be separate executables in ISE are now TCL commands and can be included in a .tcl batch file

Launching Vivado From the ISE Command Prompt



Launches Xilinx Command Prompt



```
C:\Xilinx\14.1\ISE_DS>_
```

Then type

`Vivado`	Launches GUI
`vivado –mode tcl`	Launches Vivado in command line mode
`vivado –mode batch –source <filename.tcl>`	Launches Vivado in command line, executes filename.tcl

- **`– mode tcl`**:
  - Vivado is running, no GUI
  - **`start_GUI`** TCL command will launch the GUI without forgetting what you've done
- **`– mode batch`**
  - Same as –mode tcl, but executes a tcl script on start-up

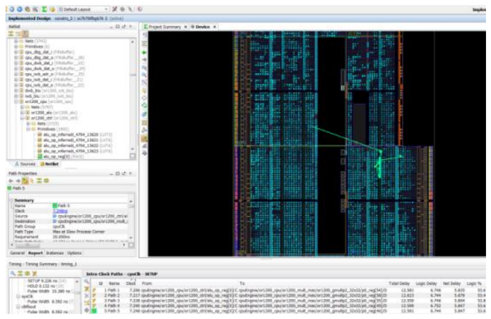
A Vivado "Batch Mode" TCL Script Might Look Something Like this

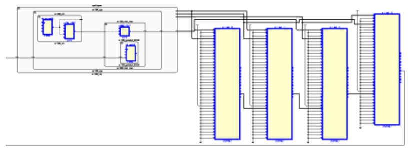
Commands Used to Synthesize / Implement Design

```
read_verilog
read_vhdl
read_edif
synth_design
report_timing_summary
write_checkpoint
opt_design
place_design
report_timing_summary
write_checkpoint
route_design
report_timing_summary
write_checkpoint
```

You can still load the design into PlanAhead GUI for Design Analysis

```
start_gui
read_checkpoint
```



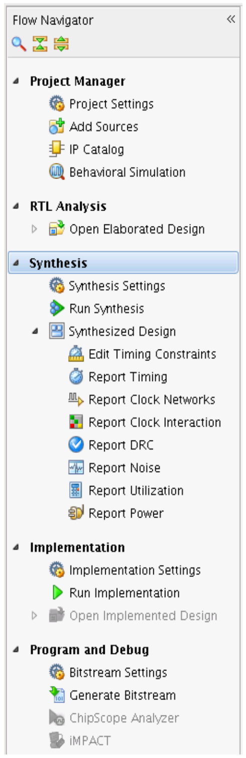


ISE Command Line Equivalence in Vivado

ISE Command Line Tool	Vivado Equivalent TCL Commands
xst	synth_design
Ngdbuild (Translate)	link_design (Not required if running synth_design)
map	opt_design power_opt_design (optional) phys_opt_design (optional)
par	place_design route_design
trce	report_timing
bitgen	write_bitstream

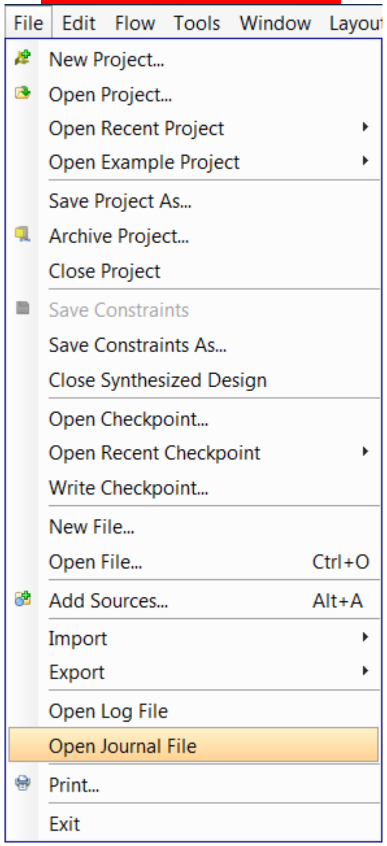
A Separate "Project Flow" Uses DIFFERENT TCL Commands

Vivado IDE for Project Management	Commands Used to Accomplish Same Thing

	`create_project …`
	`add_files …`
	`import_files …`
	`…`
	`launch_run synth_1`
	`wait_on_run …`
	`open_run …`
	`report_timing_summary`
	`launch_runs impl_1`
	`wait_on_run …`
	`open_run …`

- **Design runs use specific source sets**
  - Design Sources
  - Constraints Sets
  - Simulation Sets
- **Uses the concept of design runs**
  - `launch_run` (synthesis / implementation)
  - `wait_on_run`
  - `open_run` (synthesis / implementation)

Confused?



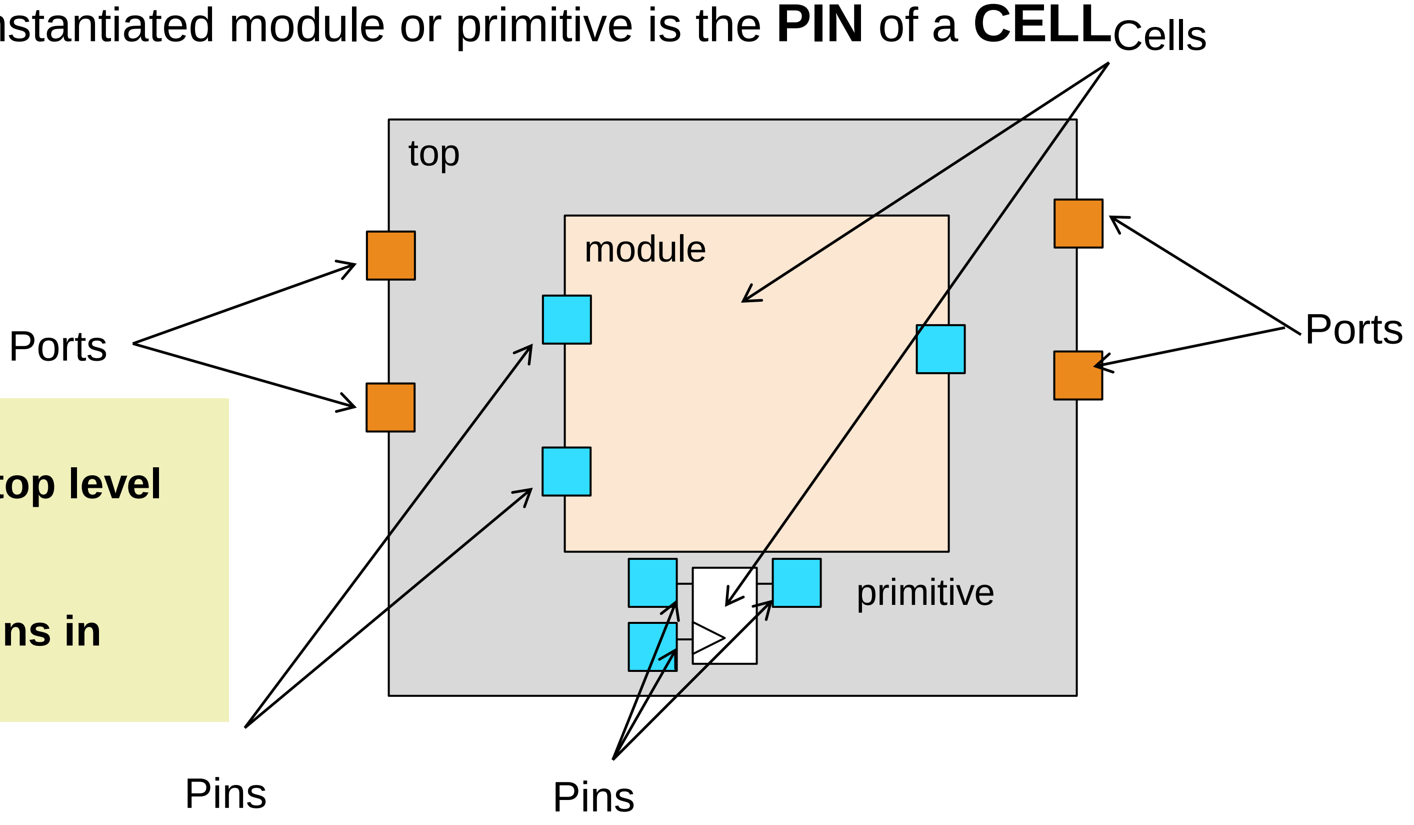
AVNET electronics marketing

XILINX ALL PROGRAMMABLE.

Pins and Ports May Not Be What you Expect

- **Ports are the pads or balls of the FPGA. Literally.**
  - You apply things like LOC and IOSTANDARD to *ports*

- **There's also CELLs and PINS**
  - The I/O of an instantiated module or primitive is the **PIN** of a **CELL**



EXAMPLE

`get_ports` returns all top level IO ports of the design

`get_pins` returns all pins in design

ALL PROGRAMMABLE™

III Vivado Physical Constraints

Finding Netlist Objects in the Design

Command	Description
get_cells	Cell objects based on name/hierarchy or connectivity
get_pins	Pin objects based name/hierarchy or connectivity
get_nets	Net objects by name/hierarchy or connectivity
get_ports	Top-level netlist ports by name or connectivity
get_iobanks	IO Bank objects, needed for constraining
all_inputs	Get a list of all input ports in the current design
all_outputs	Get a list of all output ports in the current design
all_ffs	Get a list of all flip flops in current design
all_latches	Get a list of all latches in current design
all_dsps	Get a list of dsp cells in the current design
all_rams	Get a list of ram cells in the current design

To create constraints or analyze designs, you need to access netlist objects

Simple Example 1: Utilization

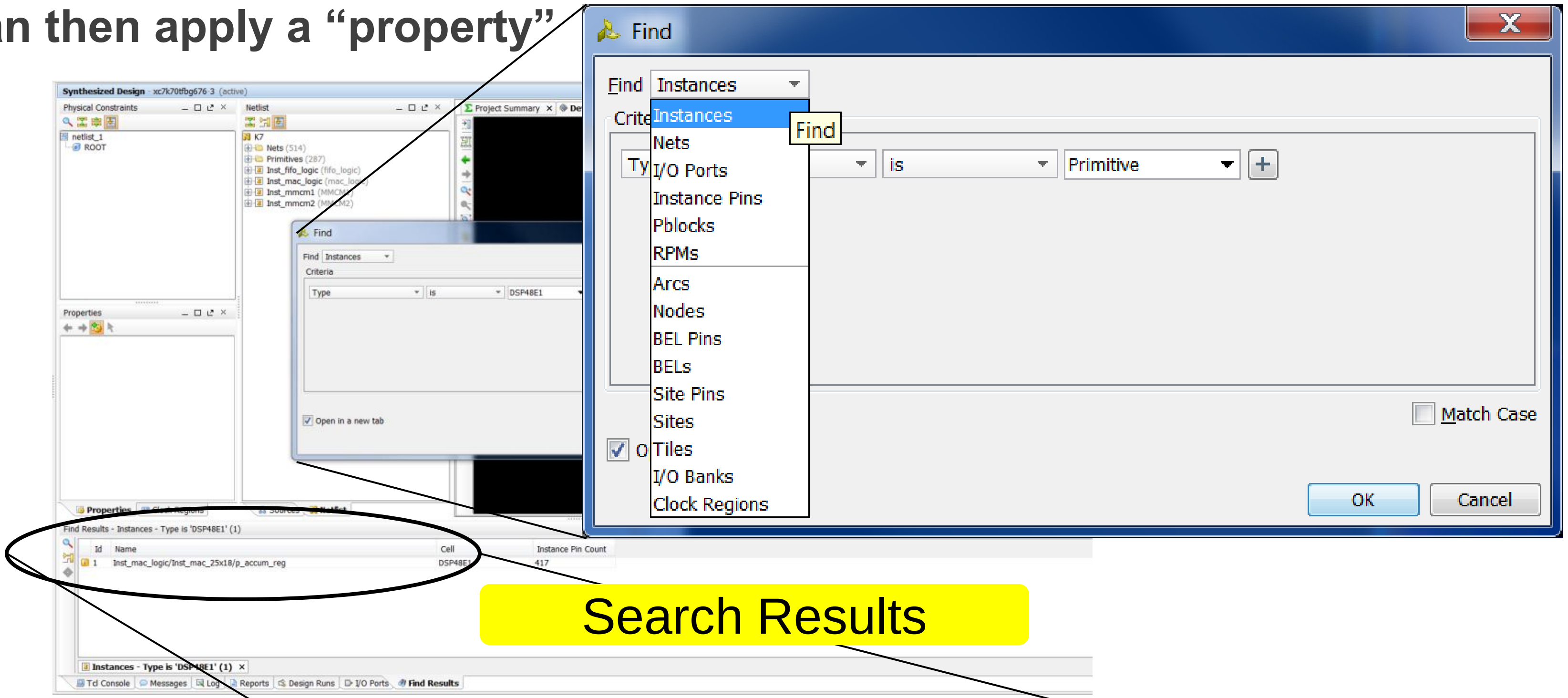
- Utilization
  - `get_cells` :returns list of all cells in the design on one line
  - `join [get_cells] "\n"` : return list of all cells in the design, but on separate lines
  - `llength [get_cells [all_dsps]]` : How many are DSPs are in the desgin

OR

- `report_utilization`
  - Just a bunch of stuff scrolling by
- `report_utilization –file my_util.txt`
  - Creates a file, sticks it . . . somewhere
- **Click the GUI button that says Report Utilization.**
  - Gives you a great table
  - But you didn't learn as much TCL

Simple Example #2: Find Netlist Objects

- Use the Find (Ctrl-F) function in the Vivado GUI to find objects in the netlist:
  - Choose <u>Instances</u> to search for Cells
- Use the Search Results with "get_cells".
- You can then apply a "property"



Find Results - Instances - Type is 'DSP48E1' (1)

Id	Name	Cell	Instance Pin Count
1	Inst_mac_logic/Inst_mac_25x18/p_accum_reg	DSP48E1	417

```
place_cell [get_cells Inst_mac_logic/Inst_mac_25x18/p_accum_reg] DSP48_X0Y24
```

Simple Example #3: Checking Your Work

```
set_property PACKAGE_PIN  A25     [get_ports COUNT_OUT_OSERDES_N]
set_property IOSTANDARD   LVDS_25 [get_ports COUNT_OUT_OSERDES_N]

set_property  IOSTANDARD LVCMOS18 [get_ports  MAC_OUT[*]]
set_property  DRIVE      12       [get_ports  MAC_OUT[*]]
set_property  SLEW       SLOW     [get_ports  MAC_OUT[*]]
```

Check your work!

```
get_property PACKAGE_PIN [get_port MAC_OUT[47]]
get_property SLEW        [get_port MAC_OUT[47]]
get_property DRIVE       [get_port MAC_OUT[47]]
```

Simple Example #4: More Sanity Checking

- You’ve set Location Constraints on all of your I/O
  - `llength [get_ports]:` Returns number of I/O in your design
  - `llength [get_property PACKAGE_PIN [get_ports]]`
  - *Should return the same number*
- Of course the tools will tell you this too after you `place_design`

Master Class: PROBES (1)

- In ISE, PROBEs allows you to tap off of a net, connecting it to an unused I/O. *How would you do the same thing in Vivado TCL?*

1. *Create a port*
2. *Select an I/O Standard*
3. *Select an unused pin location*
4. *Connect the net from the design to the port created in 1.*

Master Class: PROBES (2)

“FIND” the net

Pick a pin

Pick whatever works

Fred

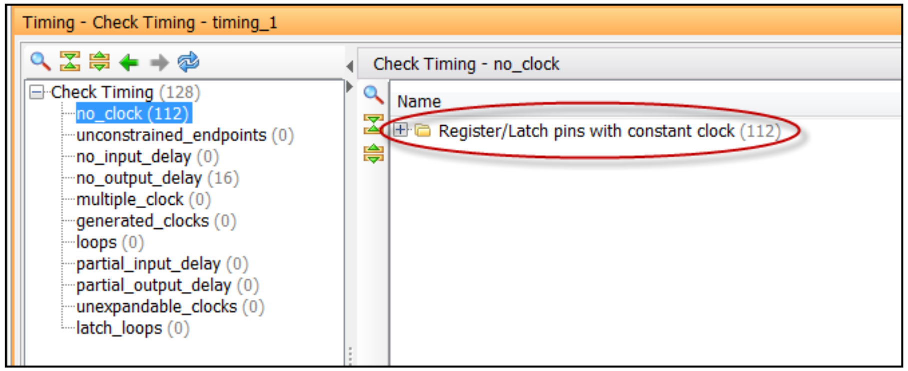
```
proc addProbe {signal pin IOStandard name} {
    create_port -direction OUT $name
    set_property IOSTANDARD $IOStandard [get_port $name]
    set_property LOC $pin [get_port $name]
    connect_net -net $signal -objects [get_port $name]
    }
```

- Save this file as probes.tcl
- After design is loaded, type “`source probes.tcl`” at the TCL prompt
- Type “`addProbe mem_en V25 LVCMOS18 Fred`” at the TCL prompt

IV. Vivado Timing Constraints

Baselining Designs With Vivado

- **Need a common starting point for wide variety of designs**
- **Constraint Development**
  - check_timing: no unconstrained clocks
  - No unreasonable (ie. 20 ps) path requirements
  - No unexpanded clocks

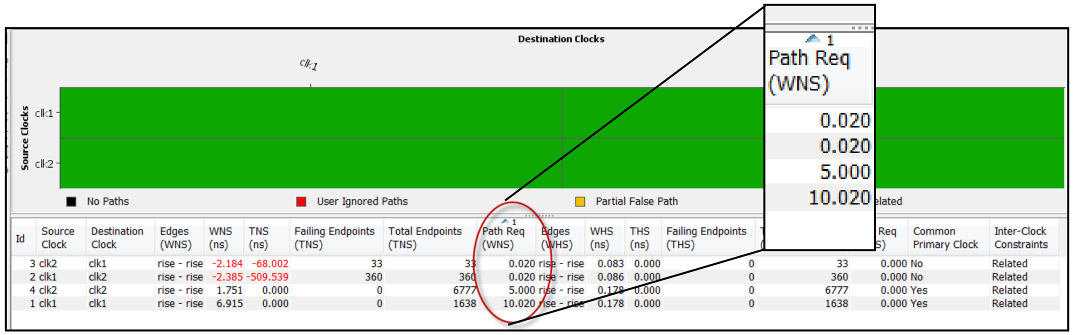


Timing - Check Timing - timing_1

- Check Timing (128)
  - no_clock (112)
  - unconstrained_endpoints (0)
  - no_input_delay (0)
  - no_output_delay (16)
  - multiple_clock (0)
  - generated_clocks (0)
  - loops (0)
  - partial_input_delay (0)
  - partial_output_delay (0)
  - unexpandable_clocks (0)
  - latch_loops (0)

Check Timing - no_clock

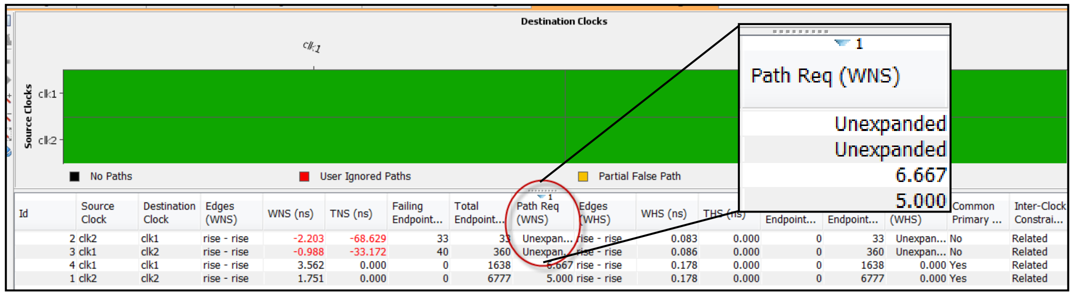
Name
Register/Latch pins with constant clock (112)



■ No Paths 🟥 User Ignored Paths 🟨 Partial False Path

Id	Source Clock	Destination Clock	Edges (WNS)	WNS (ns)	TNS (ns)	Failing Endpoints (TNS)	Total Endpoints (TNS)	Path Req (WNS)	Edges (WHS)	WHS (ns)	THS (ns)	Failing Endpoints (THS)	T	Req S)	Common Primary Clock	Inter-Clock Constraints
3	clk2	clk1	rise - rise	-2.184	-68.002	33	33	0.020	rise - rise	0.083	0.000	0	33	0.000	No	Related
2	clk1	clk2	rise - rise	-2.385	-509.539	360	360	0.020	rise - rise	0.086	0.000	0	360	0.000	No	Related
4	clk2	clk2	rise - rise	1.751	0.000	0	6777	5.000	rise - rise	0.178	0.000	0	6777	0.000	Yes	Related
1	clk1	clk1	rise - rise	6.915	0.000	0	1638	10.020	rise - rise	0.178	0.000	0	1638	0.000	Yes	Related

Path Req (WNS)
0.020
0.020
5.000
10.020



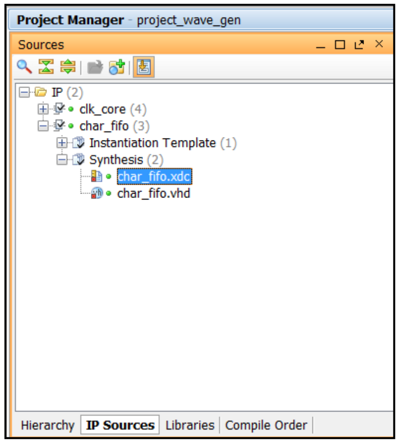
■ No Paths 🟥 User Ignored Paths 🟨 Partial False Path

Id	Source Clock	Destination Clock	Edges (WNS)	WNS (ns)	TNS (ns)	Failing Endpoint...	Total Endpoint...	Path Req (WNS)	Edges (WHS)	WHS (ns)	THS (ns)	Endpoint...	Endpoint...	(WHS)	Common Primary ...	Inter-Clock Constrai...
2	clk2	clk1	rise - rise	-2.203	-68.629	33	33	Unexpan...	rise - rise	0.083	0.000	0	33	Unexpan...	No	Related
3	clk1	clk2	rise - rise	-0.988	-33.172	40	360	Unexpan...	rise - rise	0.086	0.000	0	360	Unexpan...	No	Related
4	clk1	clk1	rise - rise	3.562	0.000	0	1638	6.667	rise - rise	0.178	0.000	0	1638	0.000	Yes	Related
1	clk2	clk2	rise - rise	1.751	0.000	0	6777	5.000	rise - rise	0.178	0.000	0	6777	0.000	Yes	Related

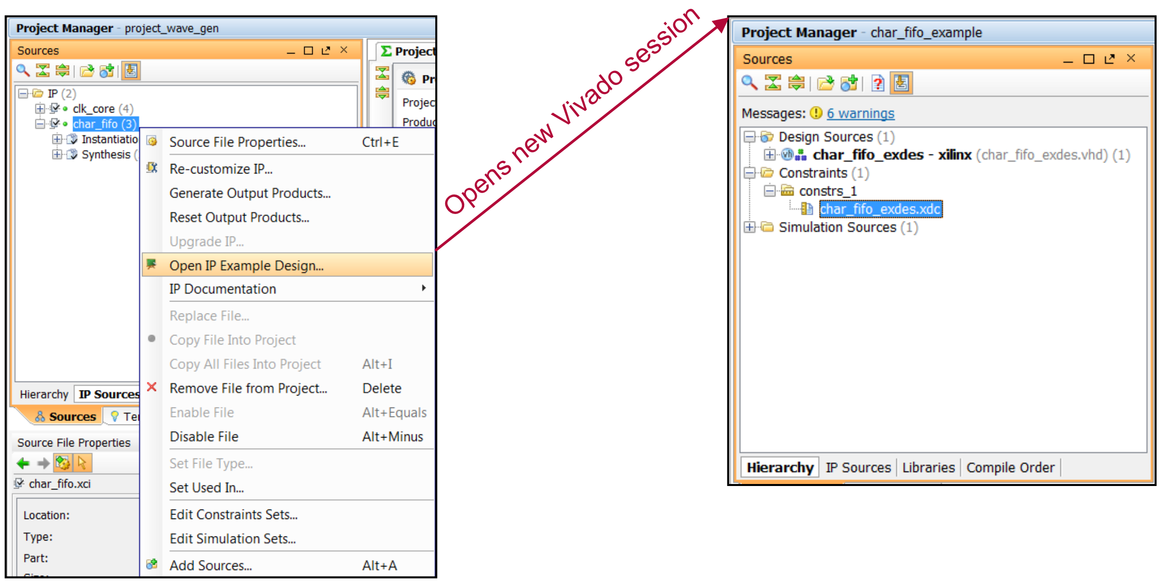
Path Req (WNS)
Unexpanded
Unexpanded
6.667
5.000

Constraint Development Tips For Baselining

- **Do Not Forget To Include IP Timing Constraints**
  - Native IP: review BOTH xdc file that comes with core AND example project xdc for timing exceptions



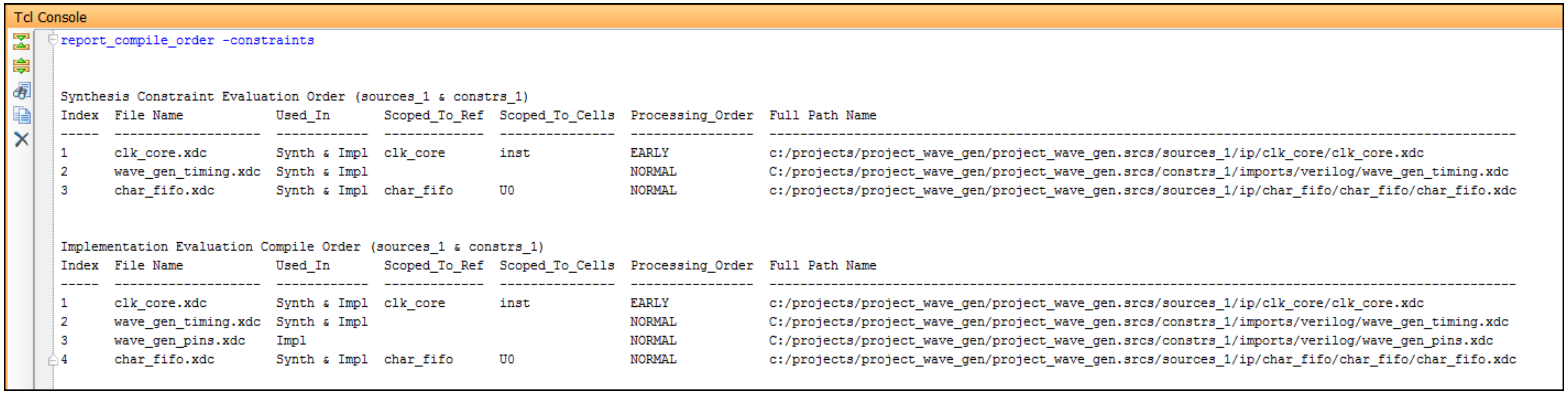
Project Manager - project_wave_gen
Sources
IP (2)
clk_core (4)
char_fifo (3)
Instantiation Template (1)
Synthesis (2)
char_fifo.xdc
char_fifo.vhd
Hierarchy | IP Sources | Libraries | Compile Order



XILINX ALL PROGRAMMABLE.

Constraint Development Tips For Baselining

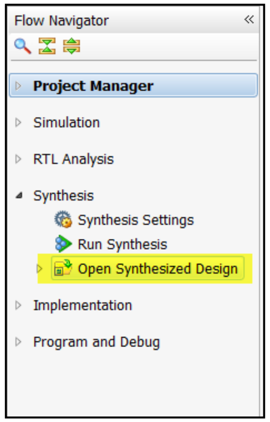
- **Do Not Forget To Include IP Timing Constraints**
  – Many cores have their own timing constraints that include important exceptions (PCIE, MIG, 2-clock distributed FIFOs…)
  – Use report_compile_order –constraints to identify all constraint file sources


Tcl Console

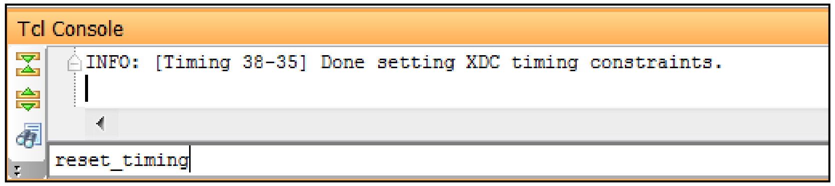
```
report_compile_order -constraints


Synthesis Constraint Evaluation Order (sources_1 & constrs_1)
Index  File Name           Used_In       Scoped_To_Ref  Scoped_To_Cells  Processing_Order  Full Path Name
-----  ------------------  ------------  -------------  ---------------  ----------------  --------------------------------------------------------------------------------------
1      clk_core.xdc        Synth & Impl  clk_core       inst             EARLY             c:/projects/project_wave_gen/project_wave_gen.srcs/sources_1/ip/clk_core/clk_core.xdc
2      wave_gen_timing.xdc Synth & Impl                                  NORMAL            C:/projects/project_wave_gen/project_wave_gen.srcs/constrs_1/imports/verilog/wave_gen_timing.xdc
3      char_fifo.xdc       Synth & Impl  char_fifo      U0               NORMAL            c:/projects/project_wave_gen/project_wave_gen.srcs/sources_1/ip/char_fifo/char_fifo/char_fifo.xdc


Implementation Evaluation Compile Order (sources_1 & constrs_1)
Index  File Name           Used_In       Scoped_To_Ref  Scoped_To_Cells  Processing_Order  Full Path Name
-----  ------------------  ------------  -------------  ---------------  ----------------  --------------------------------------------------------------------------------------
1      clk_core.xdc        Synth & Impl  clk_core       inst             EARLY             c:/projects/project_wave_gen/project_wave_gen.srcs/sources_1/ip/clk_core/clk_core.xdc
2      wave_gen_timing.xdc Synth & Impl                                  NORMAL            C:/projects/project_wave_gen/project_wave_gen.srcs/constrs_1/imports/verilog/wave_gen_timing.xdc
3      wave_gen_pins.xdc   Impl                                          NORMAL            C:/projects/project_wave_gen/project_wave_gen.srcs/constrs_1/imports/verilog/wave_gen_pins.xdc
4      char_fifo.xdc       Synth & Impl  char_fifo      U0               NORMAL            c:/projects/project_wave_gen/project_wave_gen.srcs/sources_1/ip/char_fifo/char_fifo/char_fifo.xdc
```

Developing Constraints From Scratch

- **Q. How do I start?**
  - A. Open synthesized design in Vivado IDE



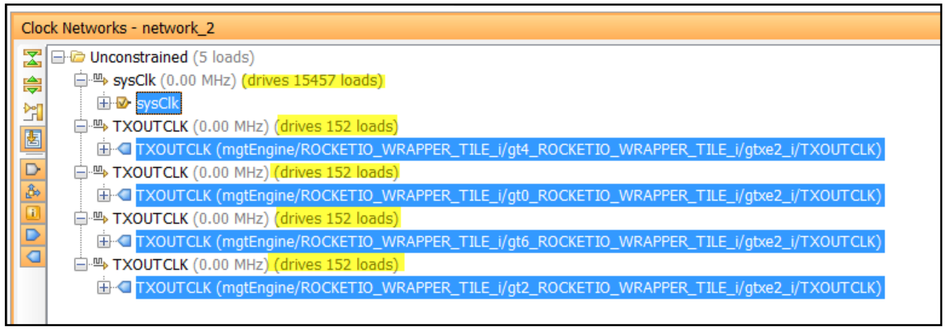
- **Q. How do I make sure I'm starting from scratch?**
  - A. tcl_console> reset_timing


Tcl Console

```
INFO: [Timing 38-35] Done setting XDC timing constraints.
```

```
reset_timing
```

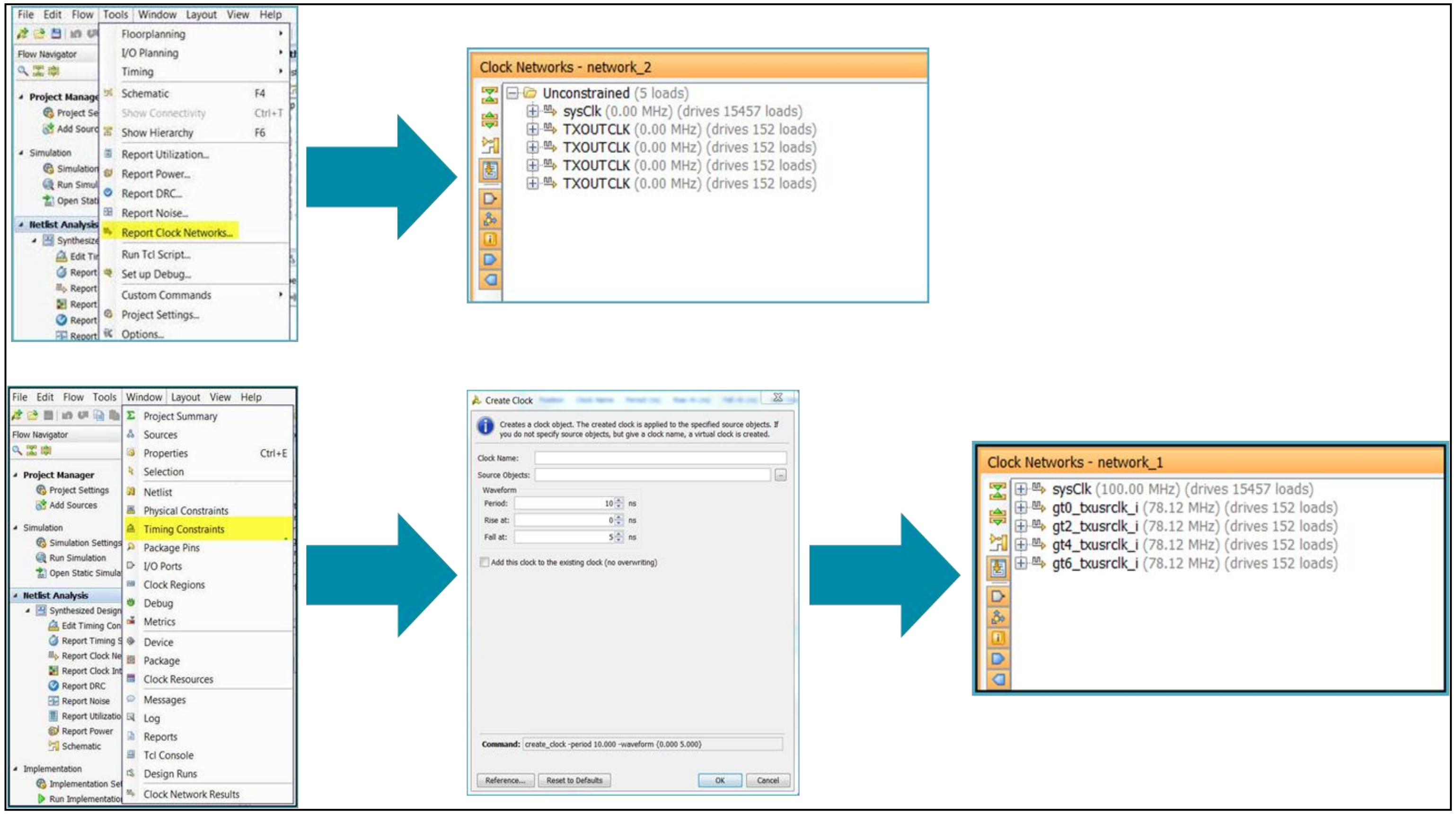
- **Q. How do I know what to constrain?**
  - A. report_clock_networks


Clock Networks - network_2

```
Unconstrained (5 loads)
  sysClk (0.00 MHz) (drives 15457 loads)
    sysClk
  TXOUTCLK (0.00 MHz) (drives 152 loads)
    TXOUTCLK (mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt4_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
    TXOUTCLK (mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt0_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
    TXOUTCLK (mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt6_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
    TXOUTCLK (mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt2_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK)
```

Developing Constraints From Scratch

- **Q. How do I know when I’m done constraining clocks?**
  - A. When report_clock_networks shows no unconstrained networks



Clock Networks - network_2

```
Unconstrained (5 loads)
  sysClk (0.00 MHz) (drives 15457 loads)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
  TXOUTCLK (0.00 MHz) (drives 152 loads)
```

Create Clock

Creates a clock object. The created clock is applied to the specified source objects. If you do not specify source objects, but give a clock name, a virtual clock is created.

Clock Name:
Source Objects:
Waveform
Period: 10 ns
Rise at: 0 ns
Fall at: 5 ns
☐ Add this clock to the existing clock (no overwriting)

Command: `create_clock -period 10.000 -waveform {0.000 5.000}`

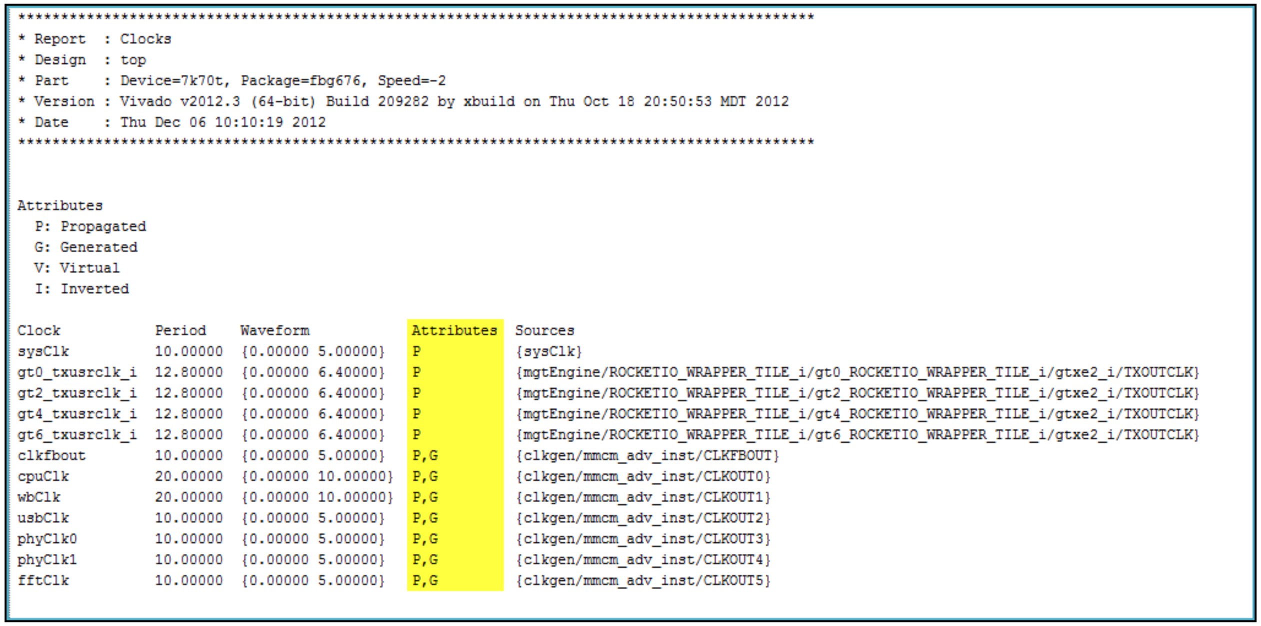
Reference... | Reset to Defaults | OK | Cancel

Clock Networks - network_1

```
sysClk (100.00 MHz) (drives 15457 loads)
gt0_txusrclk_i (78.12 MHz) (drives 152 loads)
gt2_txusrclk_i (78.12 MHz) (drives 152 loads)
gt4_txusrclk_i (78.12 MHz) (drives 152 loads)
gt6_txusrclk_i (78.12 MHz) (drives 152 loads)
```

Developing Constraints From Scratch

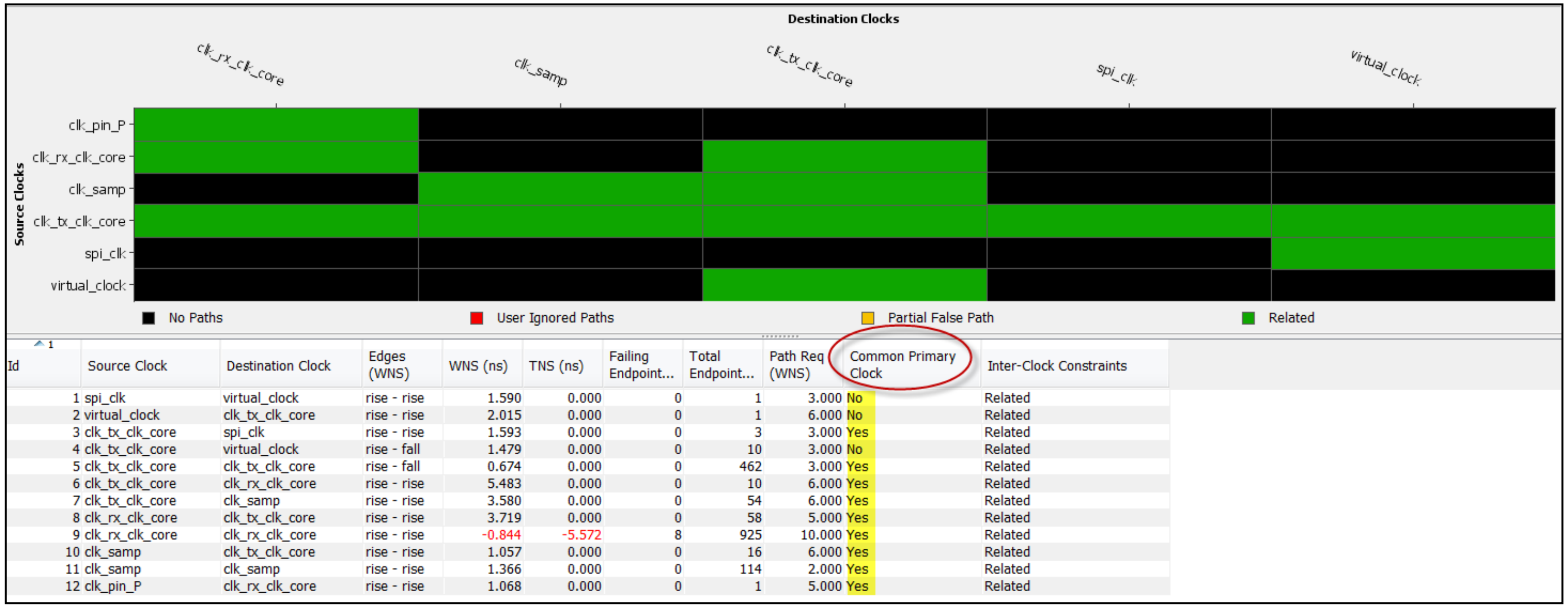
- **Q. How do I make sure my clocks are correct?**
  - A. report_clocks – shows period and waveform of every clock in the design



```
********************************************************************************************
* Report  : Clocks
* Design  : top
* Part    : Device=7k70t, Package=fbg676, Speed=-2
* Version : Vivado v2012.3 (64-bit) Build 209282 by xbuild on Thu Oct 18 20:50:53 MDT 2012
* Date    : Thu Dec 06 10:10:19 2012
********************************************************************************************


Attributes
  P: Propagated
  G: Generated
  V: Virtual
  I: Inverted

Clock            Period    Waveform            Attributes  Sources
sysClk           10.00000  {0.00000 5.00000}   P           {sysClk}
gt0_txusrclk_i   12.80000  {0.00000 6.40000}   P           {mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt0_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK}
gt2_txusrclk_i   12.80000  {0.00000 6.40000}   P           {mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt2_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK}
gt4_txusrclk_i   12.80000  {0.00000 6.40000}   P           {mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt4_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK}
gt6_txusrclk_i   12.80000  {0.00000 6.40000}   P           {mgtEngine/ROCKETIO_WRAPPER_TILE_i/gt6_ROCKETIO_WRAPPER_TILE_i/gtxe2_i/TXOUTCLK}
clkfbout         10.00000  {0.00000 5.00000}   P,G         {clkgen/mmcm_adv_inst/CLKFBOUT}
cpuClk           20.00000  {0.00000 10.00000}  P,G         {clkgen/mmcm_adv_inst/CLKOUT0}
wbClk            20.00000  {0.00000 10.00000}  P,G         {clkgen/mmcm_adv_inst/CLKOUT1}
usbClk           10.00000  {0.00000 5.00000}   P,G         {clkgen/mmcm_adv_inst/CLKOUT2}
phyClk0          10.00000  {0.00000 5.00000}   P,G         {clkgen/mmcm_adv_inst/CLKOUT3}
phyClk1          10.00000  {0.00000 5.00000}   P,G         {clkgen/mmcm_adv_inst/CLKOUT4}
fftClk           10.00000  {0.00000 5.00000}   P,G         {clkgen/mmcm_adv_inst/CLKOUT5}
```

Developing Constraints From Scratch

- Q. How do I know what clocks should be related?
  - A. report_clock_interaction – sort by Common Primary Clock



Destination Clocks

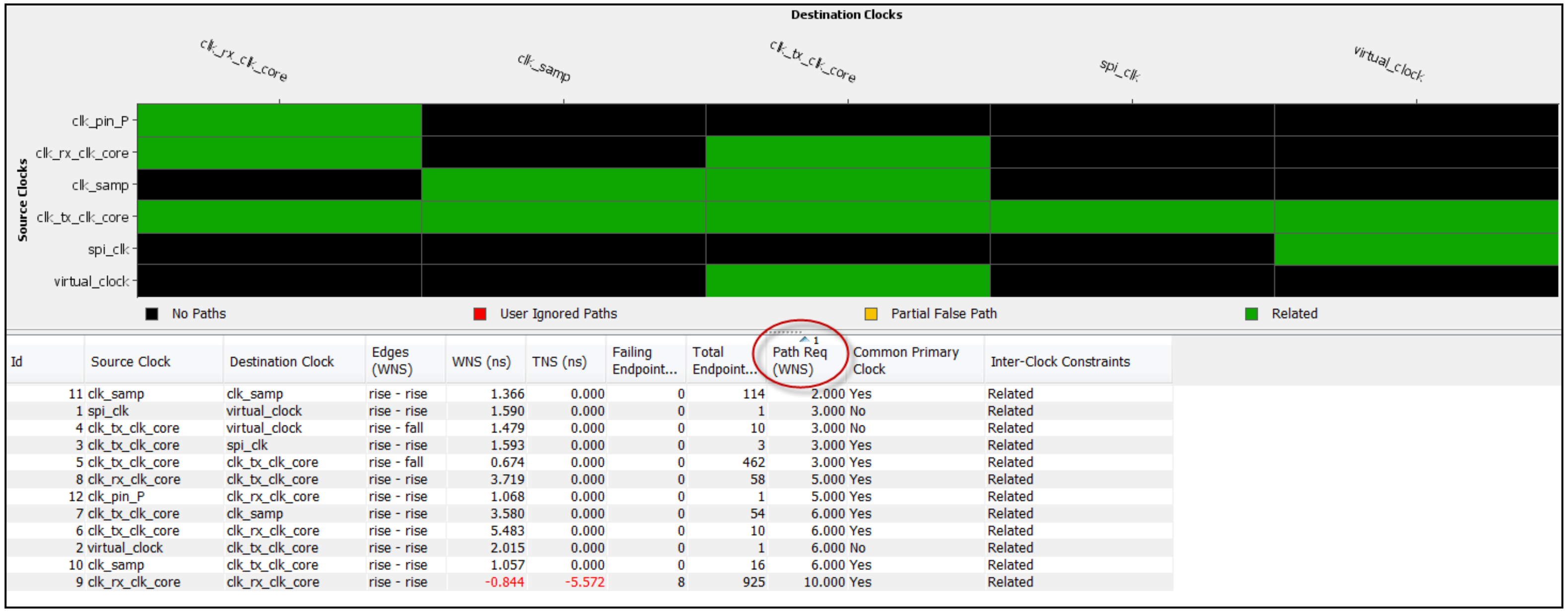
Source Clocks	clk_rx_clk_core	clk_samp	clk_tx_clk_core	spi_clk	virtual_clock
clk_pin_P	🟩	⬛	⬛	⬛	⬛
clk_rx_clk_core	🟩	⬛	🟩	⬛	⬛
clk_samp	⬛	🟩	🟩	⬛	⬛
clk_tx_clk_core	🟩	🟩	🟩	🟩	🟩
spi_clk	⬛	⬛	⬛	⬛	🟩
virtual_clock	⬛	⬛	🟩	⬛	⬛

⬛ No Paths 🟥 User Ignored Paths 🟨 Partial False Path 🟩 Related

Id	Source Clock	Destination Clock	Edges (WNS)	WNS (ns)	TNS (ns)	Failing Endpoint...	Total Endpoint...	Path Req (WNS)	Common Primary Clock	Inter-Clock Constraints
1	spi_clk	virtual_clock	rise - rise	1.590	0.000	0	1	3.000	No	Related
2	virtual_clock	clk_tx_clk_core	rise - rise	2.015	0.000	0	1	6.000	No	Related
3	clk_tx_clk_core	spi_clk	rise - rise	1.593	0.000	0	3	3.000	Yes	Related
4	clk_tx_clk_core	virtual_clock	rise - fall	1.479	0.000	0	10	3.000	No	Related
5	clk_tx_clk_core	clk_tx_clk_core	rise - fall	0.674	0.000	0	462	3.000	Yes	Related
6	clk_tx_clk_core	clk_rx_clk_core	rise - rise	5.483	0.000	0	10	6.000	Yes	Related
7	clk_tx_clk_core	clk_samp	rise - rise	3.580	0.000	0	54	6.000	Yes	Related
8	clk_rx_clk_core	clk_tx_clk_core	rise - rise	3.719	0.000	0	58	5.000	Yes	Related
9	clk_rx_clk_core	clk_rx_clk_core	rise - rise	-0.844	-5.572	8	925	10.000	Yes	Related
10	clk_samp	clk_tx_clk_core	rise - rise	1.057	0.000	0	16	6.000	Yes	Related
11	clk_samp	clk_samp	rise - rise	1.366	0.000	0	114	2.000	Yes	Related
12	clk_pin_P	clk_rx_clk_core	rise - rise	1.068	0.000	0	1	5.000	Yes	Related

Developing Constraints From Scratch

- Q. How do I know if I have unrealistic path requirements?
  - A. report_clock_interaction – sort by Path Req (WNS)



Destination Clocks

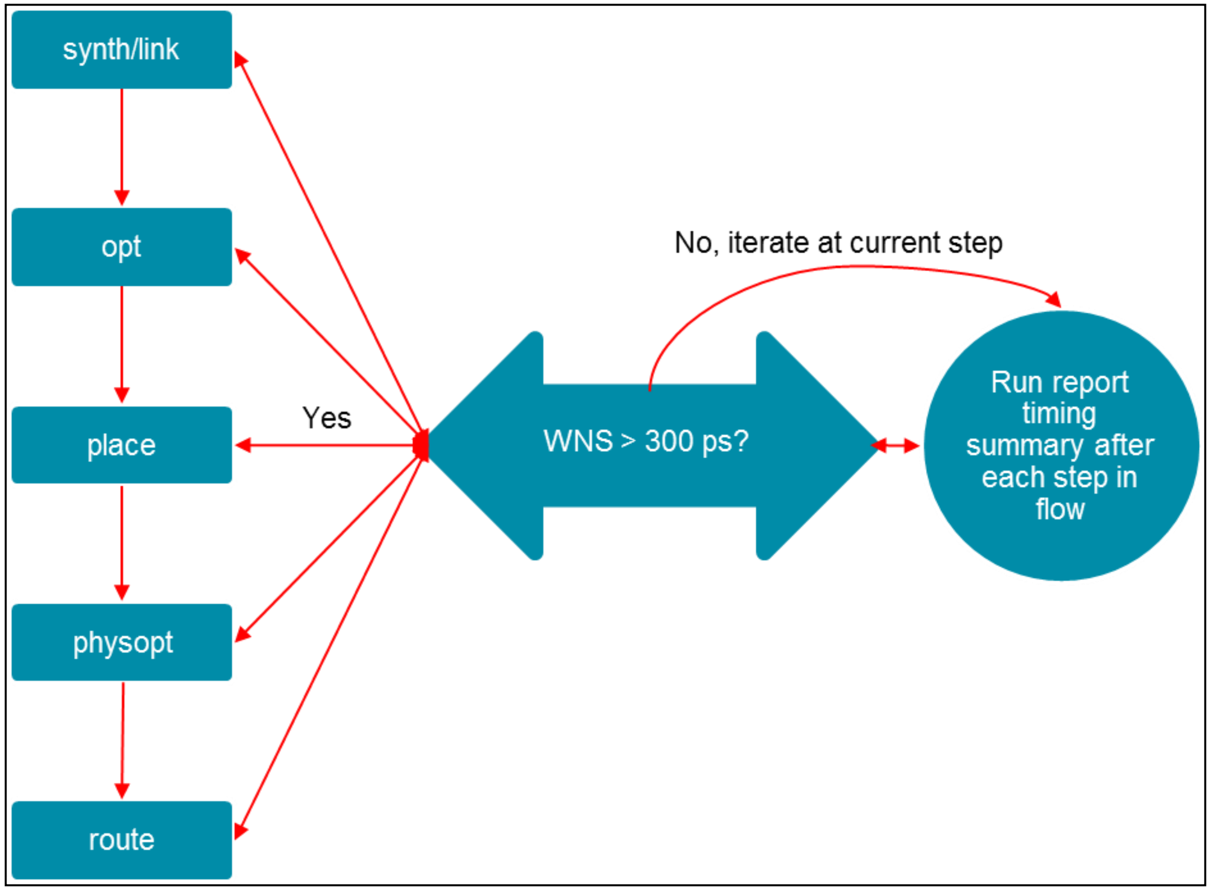
Source Clocks	clk_rx_clk_core	clk_samp	clk_tx_clk_core	spi_clk	virtual_clock
clk_pin_P	🟩	⬛	⬛	⬛	⬛
clk_rx_clk_core	🟩	⬛	🟩	⬛	⬛
clk_samp	⬛	🟩	🟩	⬛	⬛
clk_tx_clk_core	🟩	🟩	🟩	🟩	🟩
spi_clk	⬛	⬛	⬛	⬛	🟩
virtual_clock	⬛	⬛	🟩	⬛	⬛

⬛ No Paths 🟥 User Ignored Paths 🟨 Partial False Path 🟩 Related

Id	Source Clock	Destination Clock	Edges (WNS)	WNS (ns)	TNS (ns)	Failing Endpoint...	Total Endpoint...	Path Req (WNS)	Common Primary Clock	Inter-Clock Constraints
11	clk_samp	clk_samp	rise - rise	1.366	0.000	0	114	2.000	Yes	Related
1	spi_clk	virtual_clock	rise - rise	1.590	0.000	0	1	3.000	No	Related
4	clk_tx_clk_core	virtual_clock	rise - fall	1.479	0.000	0	10	3.000	No	Related
3	clk_tx_clk_core	spi_clk	rise - rise	1.593	0.000	0	3	3.000	Yes	Related
5	clk_tx_clk_core	clk_tx_clk_core	rise - fall	0.674	0.000	0	462	3.000	Yes	Related
8	clk_rx_clk_core	clk_tx_clk_core	rise - rise	3.719	0.000	0	58	5.000	Yes	Related
12	clk_pin_P	clk_rx_clk_core	rise - rise	1.068	0.000	0	1	5.000	Yes	Related
7	clk_tx_clk_core	clk_samp	rise - rise	3.580	0.000	0	54	6.000	Yes	Related
6	clk_tx_clk_core	clk_rx_clk_core	rise - rise	5.483	0.000	0	10	6.000	Yes	Related
2	virtual_clock	clk_tx_clk_core	rise - rise	2.015	0.000	0	1	6.000	No	Related
10	clk_samp	clk_tx_clk_core	rise - rise	1.057	0.000	0	16	6.000	Yes	Related
9	clk_rx_clk_core	clk_rx_clk_core	rise - rise	-0.844	-5.572	8	925	10.000	Yes	Related

Baseline Stage 2

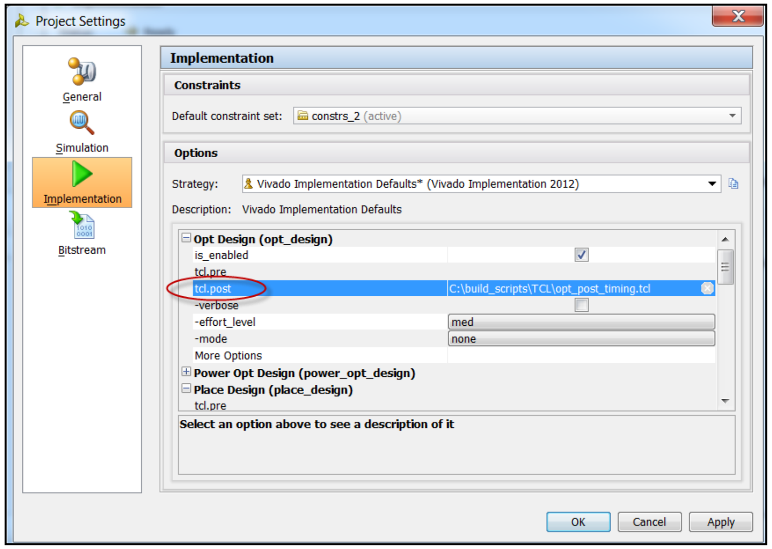
- Run **report_timing_summary** after each step
- Ensure WNS < 300 ps



synth/link
opt
place
physopt
route
Yes
No, iterate at current step
WNS > 300 ps?
Run report timing summary after each step in flow

Setting Up report_timing_summary

- GUI



Implementation

Constraints

Default constraint set: constrs_2 (active)

Options

Strategy: Vivado Implementation Defaults* (Vivado Implementation 2012)

Description: Vivado Implementation Defaults

Opt Design (opt_design)	
is_enabled	✓
tcl.pre	
tcl.post	C:\build_scripts\TCL\opt_post_timing.tcl
-verbose	
-effort_level	med
-mode	none
More Options	
Power Opt Design (power_opt_design)	
Place Design (place_design)	
tcl.pre	

Select an option above to see a description of it

opt_post_timing.tcl file:

```
report_timing_summary -file opt_timing.rpt
```

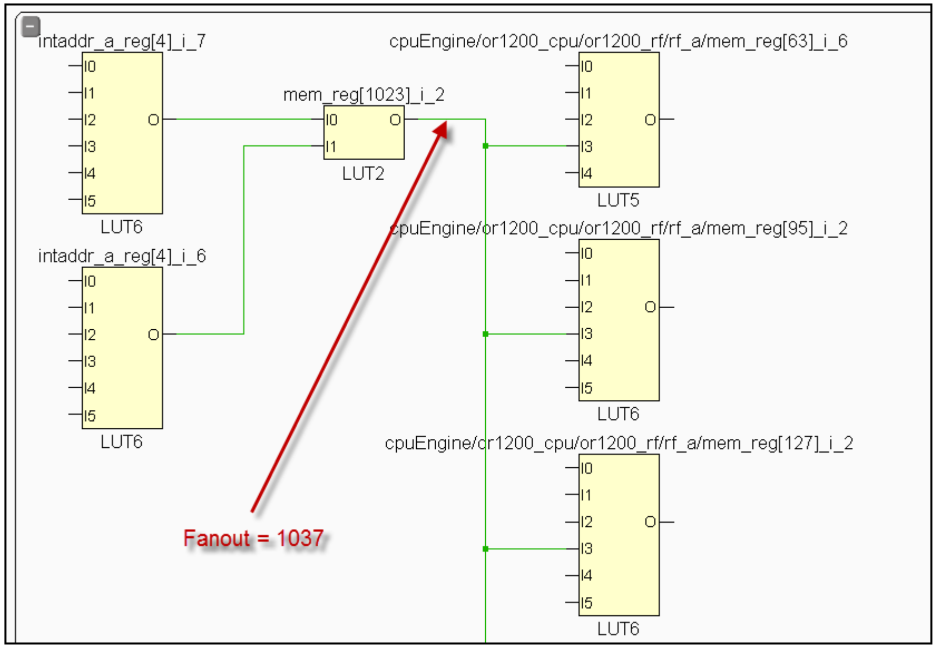
- Batch

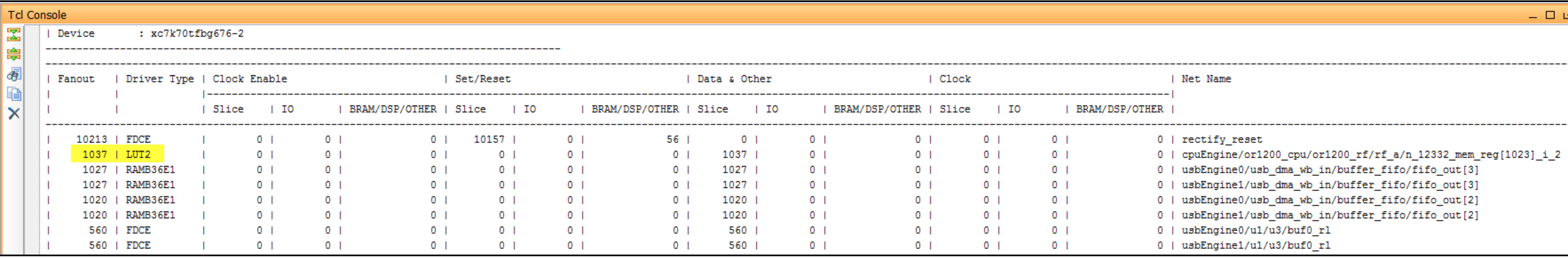
Build.tcl file:

```
link_design -name top -part xc7vx1140tflg1928-2
read_xdc top.xdc

opt_design
report_timing_summary -file opt_timing.rpt
write_checkpoint -force opt.dcp

place_design
report_timing_summary -file place_timing.rpt
write_checkpoint -force place.dcp

phys_opt_design
report_timing_summary -file popt_timing.rpt
write_checkpoint -force popt.dcp

route_design
report_timing_summary –file routed_timing.rpt
write_checkpoint –force routed.dcp
```

High Fanout Nets Driven by LUTs

- Recommended to drive high fanout nets from a synchronous start point
- Identify high fanout nets driven by LUTs
  report_high_fanout_nets –load_types –max_nets 100
  – 2012.4 requires <u>placed</u> design
  – 2013.1 hope to be able to do this before placement


intaddr_a_reg[4]_i_7
I0
I1
I2
I3
I4
I5
O
LUT6
mem_reg[1023]_i_2
LUT2
cpuEngine/or1200_cpu/or1200_rf/rf_a/mem_reg[63]_i_6
LUT5
cpuEngine/or1200_cpu/or1200_rf/rf_a/mem_reg[95]_i_2
LUT6
intaddr_a_reg[4]_i_6
LUT6
cpuEngine/or1200_cpu/or1200_rf/rf_a/mem_reg[127]_i_2
LUT6
Fanout = 1037



```
| Device        : xc7k70tfbg676-2
---------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Fanout    | Driver Type | Clock Enable                                      | Set/Reset                                         | Data & Other                                      | Clock                                             | Net Name
|           |             |---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |             | Slice     | IO        | BRAM/DSP/OTHER | Slice     | IO        | BRAM/DSP/OTHER | Slice     | IO        | BRAM/DSP/OTHER | Slice     | IO        | BRAM/DSP/OTHER |
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|     10213 | FDCE        |         0 |         0 |              0 |     10157 |         0 |             56 |         0 |         0 |              0 |         0 |         0 |              0 | rectify_reset
|      1037 | LUT2        |         0 |         0 |              0 |         0 |         0 |              0 |      1037 |         0 |              0 |         0 |         0 |              0 | cpuEngine/or1200_cpu/or1200_rf/rf_a/n_12332_mem_reg[1023]_i_2
|      1027 | RAMB36E1    |         0 |         0 |              0 |         0 |         0 |              0 |      1027 |         0 |              0 |         0 |         0 |              0 | usbEngine0/usb_dma_wb_in/buffer_fifo/fifo_out[3]
|      1027 | RAMB36E1    |         0 |         0 |              0 |         0 |         0 |              0 |      1027 |         0 |              0 |         0 |         0 |              0 | usbEngine1/usb_dma_wb_in/buffer_fifo/fifo_out[3]
|      1020 | RAMB36E1    |         0 |         0 |              0 |         0 |         0 |              0 |      1020 |         0 |              0 |         0 |         0 |              0 | usbEngine0/usb_dma_wb_in/buffer_fifo/fifo_out[2]
|      1020 | RAMB36E1    |         0 |         0 |              0 |         0 |         0 |              0 |      1020 |         0 |              0 |         0 |         0 |              0 | usbEngine1/usb_dma_wb_in/buffer_fifo/fifo_out[2]
|       560 | FDCE        |         0 |         0 |              0 |         0 |         0 |              0 |       560 |         0 |              0 |         0 |         0 |              0 | usbEngine0/u1/u3/buf0_rl
|       560 | FDCE        |         0 |         0 |              0 |         0 |         0 |              0 |       560 |         0 |              0 |         0 |         0 |              0 | usbEngine1/u1/u3/buf0_rl
```

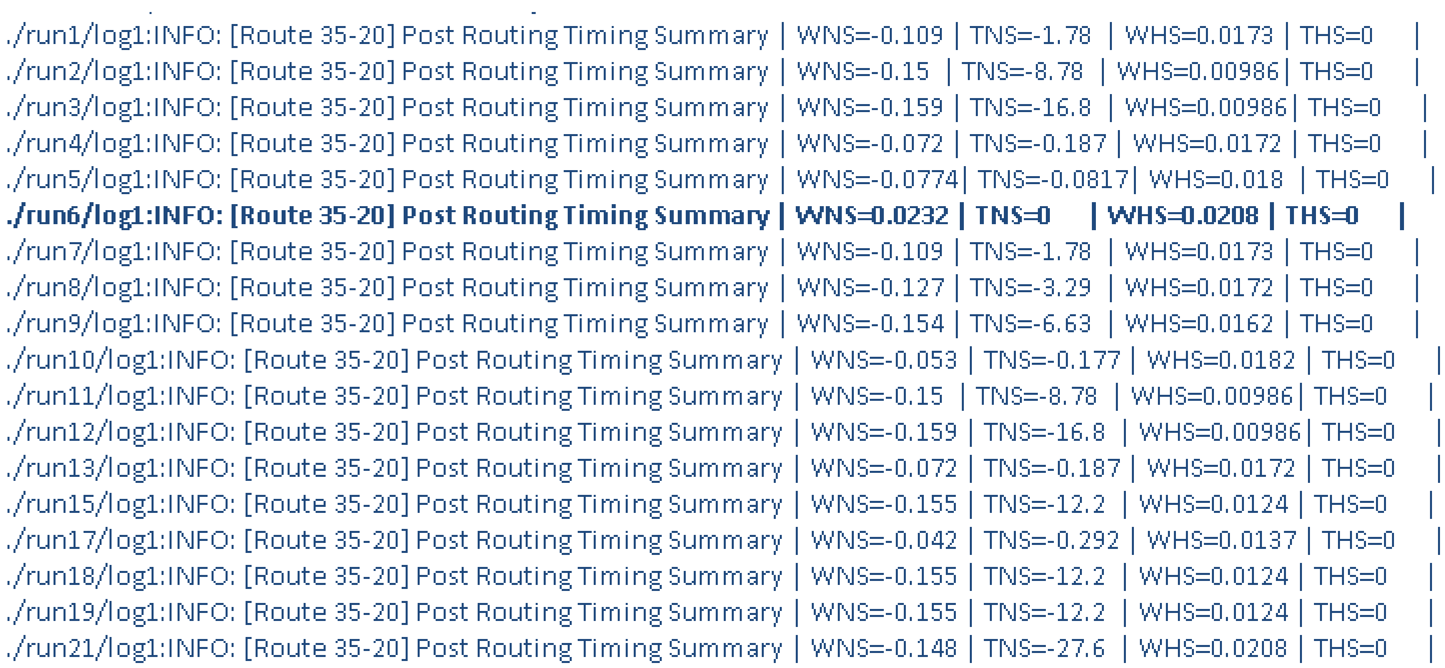
Incremental Placement

- **Design checkpoint from near-timing-closed run can be used to guide placement for a second run through the tools**

```
opt_design
place_design
phys_opt_design
route_design
write_checkpoint –guide.dcp

place_design -incremental guide.dcp
phys_opt_design
route_design
```

Vivado Strategies

- **2013.1 will include TCL based strategies**
  - Requires manual application until 2013.3 (est)
- **Useful for that last mile of timing closure**



```
./run1/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.109 | TNS=-1.78  | WHS=0.0173 | THS=0    |
./run2/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.15  | TNS=-8.78  | WHS=0.00986| THS=0    |
./run3/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.159 | TNS=-16.8  | WHS=0.00986| THS=0    |
./run4/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.072 | TNS=-0.187 | WHS=0.0172 | THS=0    |
./run5/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.0774| TNS=-0.0817| WHS=0.018  | THS=0    |
./run6/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=0.0232 | TNS=0      | WHS=0.0208 | THS=0    |
./run7/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.109 | TNS=-1.78  | WHS=0.0173 | THS=0    |
./run8/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.127 | TNS=-3.29  | WHS=0.0172 | THS=0    |
./run9/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.154 | TNS=-6.63  | WHS=0.0162 | THS=0    |
./run10/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.053 | TNS=-0.177 | WHS=0.0182 | THS=0    |
./run11/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.15  | TNS=-8.78  | WHS=0.00986| THS=0    |
./run12/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.159 | TNS=-16.8  | WHS=0.00986| THS=0    |
./run13/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.072 | TNS=-0.187 | WHS=0.0172 | THS=0    |
./run15/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.155 | TNS=-12.2  | WHS=0.0124 | THS=0    |
./run17/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.042 | TNS=-0.292 | WHS=0.0137 | THS=0    |
./run18/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.155 | TNS=-12.2  | WHS=0.0124 | THS=0    |
./run19/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.155 | TNS=-12.2  | WHS=0.0124 | THS=0    |
./run21/log1:INFO: [Route 35-20] Post Routing Timing Summary | WNS=-0.148 | TNS=-27.6  | WHS=0.0208 | THS=0    |
```

V. For More Information

There is No Shortage of Information on TCL

- Vivado Design Suite Tcl Command Reference Guide (Xilinx.com)
- Vivado Design Suite Properties Reference Guide (Xilinx.com)
- The GUI
  - `help` returns a number of categories, `help` –[category] returns additional detail
  - [command] `–help` returns detail on the command
- Xilinx Tcl Cheat Sheet
- Tcl and the TK Toolkit (John K. Ousterhout, Amazon.com)