XILINX
ALL PROGRAMMABLE™

Using Tcl Scripts to Analyze Design Automatically

Agenda

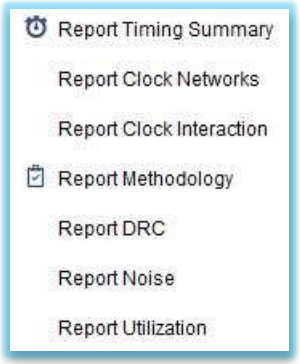
- Scanning the analysis items
- Critical paths that should be emphasized
- Demo

Agenda

- Scanning the analysis items
- Critical paths that should be emphasized
- Demo

The Challenges of Timing Analysis

- The design is more complex, and the fmax is expected to be far more higher
- Fundamental analysis is a necessary part, and can be done through GUI
  - Report timing summary
  - Report clock networks
  - Report clock interaction
  - Report methodology
  - Report utilization



- Further analysis tends to be required, and have to be done by Tcl
  - Paths from FF to the control pins of BRAM with large fanouts
  - Paths between blocks like BRAMs, URAMs, and DSPs
- UFDM remains the focus of timing design and analysis

Features of Tcl Scripts

- Everything is done automatically
  - Open DCP
  - Analyze the design
- Cover many aspects of timing analysis
  - Basis analyses
  - Further analyses
- Identify the potential risks threatening timing closure
  - More accurately
  - More quickly
- Produce analysis report
  - A bunch of reports are generated both in the form of csv and in the GUI
  - Benefit customers as well as ourselves

Benefits By Using Tcl Scripts

- Work more efficiently and effectively
  - Verified Tcl scripts helps to avoid manually inputting commands one by one
  - Check the potential risks as many as possible at a time
  - Allow you to finish other tasks during analysis
- If DCP is accessible
  - A variety of detailed reports will be provided for customers
- If DCP is inaccessible
  - Identify the potential risks by means of these reports

Analysis Items in This Tcl Document (1)

N	Items（Orange:High priority; Green:Low）	Comments
1	生成3个报告： 1. 时钟网络报告(clock networks) 2. 时序报告 (timing summary) 3. 所有时序违例（setup）路径报告	set max_paths_neg_slack 100 报告3会生成neg_slack_timing_paths.csv
2	分析每个时钟域的Logic Level	set max_paths_logic_level 1000 生成ClkName_ClkFreqMHZ_LL.csv num是指该时钟频率下理论上所能支持的最大Logic Level
3	分析从FF到BLOCK (BRAM/URAM/DSP)控制端口的路径	set max_paths_ff2block 1000 set ff2block_target_fanout 4 //设置路径的最大扇出 set ff2block_freq 300 //设置待分析时钟的时钟频率，单位MHz，>=该时钟频率的时钟下的路径会被分析 生成文件：(block可能是bram, uram 或dsp) ClkName_ClkFreqMHz_ff2block_ctrl_path_LL_0.csv (LogicLevel = 0) ClkName_ClkFreqMHz_ff2block_ctrl_path_LL_g_0.csv (LogicLevel > 0)
4	分析从BLOCK到FF的路径	set max_paths_bram2ff 1000 set max_paths_uram2ff 1000 set max_paths_dsp2ff 1000 生成文件：bram2ff.csv, uram2ff.csv, dsp2ff.csv
5	分析以Shift Register为终点的路径	set max_paths_end_srl 100 生成文件：paths_end_srl.csv
6	分析BLOCK (BRAM/URAM/DSP/GT)之间的路径	set max_paths_block2block 1000 生成文件：paths_block2block.csv

Analysis Items in This Tcl Document (2)

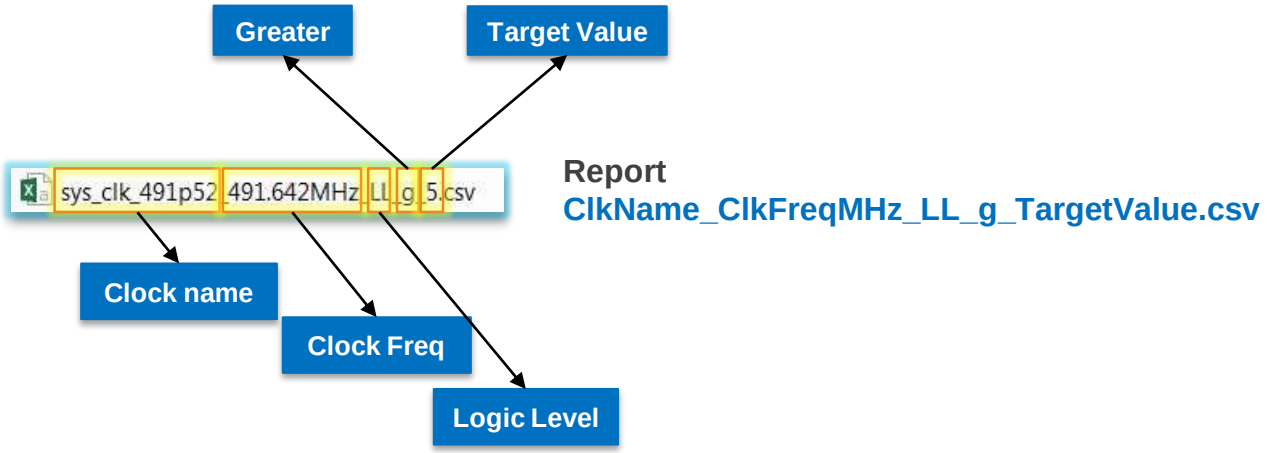
7	分析Clock skew,具体描述见ug949, page 218	set max_paths_mmcmi2o 100 set max_paths_mmcmo2i 100 生成文件：paths_ClkName1_Between_ClkName2.csv
8	分析CDC路径	生成两个报告：1.clock interatcion 2. cdc (report_cdc)
9	分析control set	生成文件：ctrl_set.rpt（当control set需要优化时会生成，在rpt文件夹下）
10	分析congestion level	生成congestion level报告
11	分析complexity	生成complexity报告
12	查找用作MAC时未使用MREG的DSP48	生成相应报告
13	查找用作MAC/Adder时未使用PREG的DSP48	生成相应报告
14	查找DOA/DOB为0的BRAM	生成相应报告，生成bram_no_reg.csv
15	查找深度为1/2/3的SRL	生成相应报告
16	分析LUT6使用率	在Consol窗口中显示信息
17	分析MUXF使用率	在Consol窗口中显示信息
18	锁存器分析	生成相应报告
19	跨SLR路径分析	生成相应报告 （set is_route_design 1，此处必须设置为1）
20	高扇出网络分析	生成相应报告
21	门控时钟分析	生成相应报告
22	约束分析	生成4个文件 1. invalid_constraints.xdc 2. ignored_exceptions.xdc 3. ignored_objects_exceptions.xdc 4. merged_exceptions.xdc
23	QoR分析	生成QoR报告（在qor文件夹下）
24	资源利用率分析	生成相应报告

Agenda

- Scanning the analysis items
- Critical items that should be emphasized
- Demo

Logic Level Analysis

- Logic level is calculated based on device family, speed grade and target clock frequency, so it is more accurate
- Every clock will be evaluated as long as the timing paths are available in the clock group
- Report containing paths with the logic level greater than target value will be produced in the form of csv

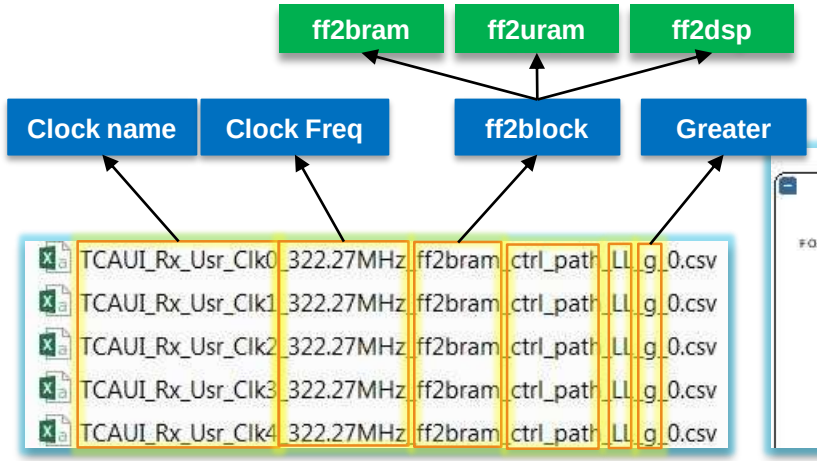


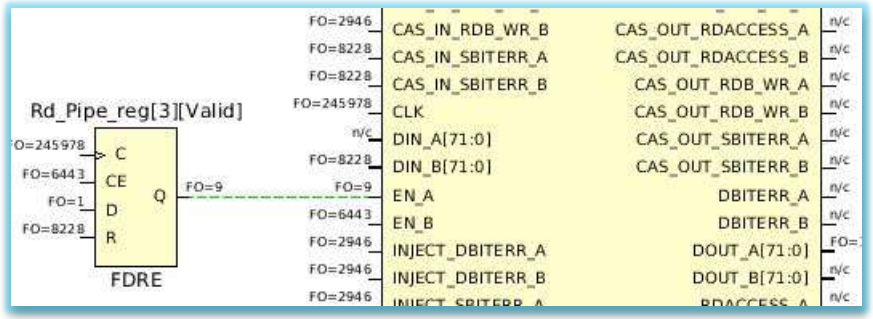
Paths Start at FF and End at the Control Pins of Blocks

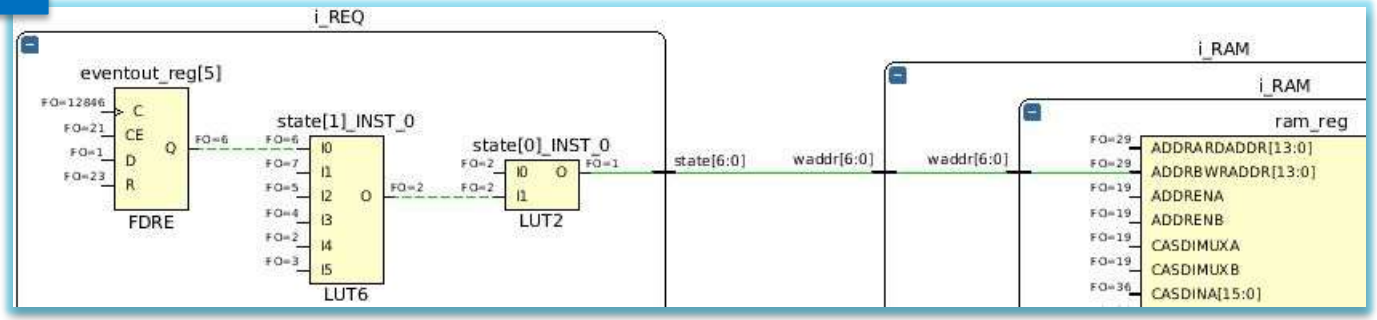
- Blocks refer to BRAMs, URAMs and DSPs
- Control pins include RST/CE/ADDR/WE/BWE/EN
- Two types of paths will be analyzed and recorded
  - Logic level is 0, but fanout is greater than expected value
    - This value can be set beforehand
  - Logic level is larger than 0

Report

ClkName_ClkFreqMHz_ff2block_ctrl_path_LL_g_0.csv
ClkName_ClkFreqMHz_ff2block_ctrl_path_LL_0.csv







Paths with Dedicated Blocks and Macro Primitives

(Chapter 5, page 215, ug949)

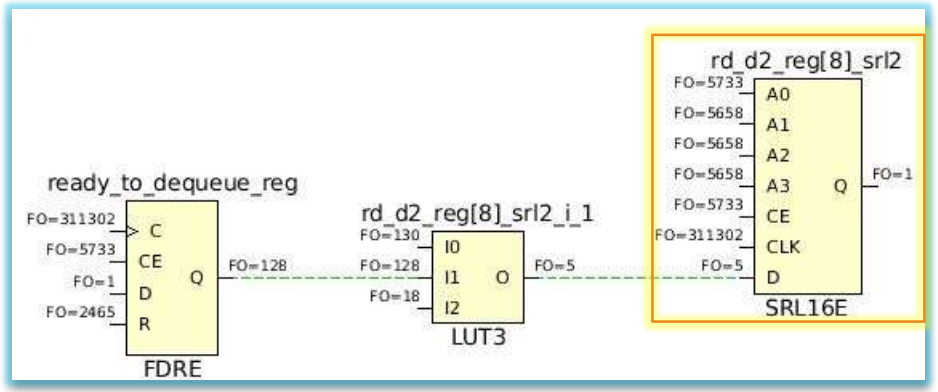
- These primitives usually have the following timing characteristics
  - Higher setup/hold/clock-to-output timing arc values for some pins
  - Higher routing delays than regular FD/LUT connections
  - Higher clock skew variation than regular FD-FD paths
- For these reasons, Xilinx recommends
  - Pipelining paths from and to Dedicated Blocks and Macro Primitives as much as possible
  - Restructuring the combinational logic connected to these cells to reduce the logic levels by at least 1 or 2 cells if latency incurred by pipelining is a concern
  - Meeting setup timing by at least 500 ps on these paths before placement
  - Replicate cones of logic connected to too many Dedicated Blocks or Macro Primitives if they need to be placed far apart

Report
paths_block2block.csv

Paths end at SRL (Shift Register)

- SRL: SRL16E/SRL32E, based on LUT in SLICEM
- Pulling the first register out of the shift register will have a positive effect on timing closure in some situations
  - using the SRL_STYLE attribute in RTL

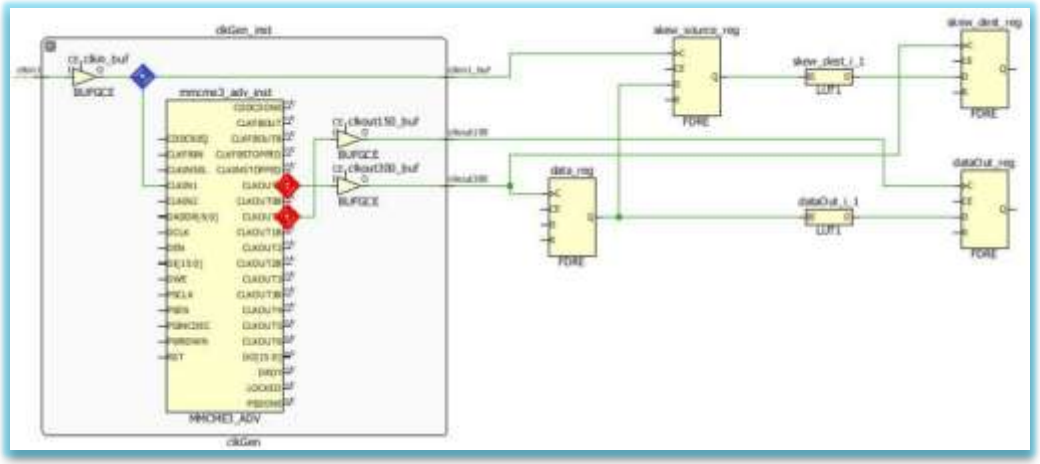
Report
paths_end_srl.csv



Reducing Clock Skew

(Chapter 5, page 225, ug949)

- Timing paths between synchronous clocks driven by separate clock buffers exhibit higher skew
  - Reason: the common node is located before the clock buffers
- The clock skew is even worse for timing paths between unbalanced clock trees
  - the delay difference between the source and destination clock paths
  - Although positive skew helps with meeting setup time, it hurts hold time closure, and vice versa



Report
paths_ClkName1_Between_ClkName2.csv

Crossing Die Paths Analysis

- The Tcl scripts below can be applied to both placed and routed design

```
set slr_num [llength $slrs]
if {$is_placed_design == 1 || $is_routed_design == 1 && $slr_num > 1} {
  set slr_list [get_timing_paths -max 100 -slack_lesser_than 0 -filter \
  {INTER_SLR_COMPENSATION != ""}]
  report_timing -of $slr_list -name FAILING_SLRS –file failing_slr_timing_paths.rpt
}
```

Report
failing_slr.csv

Searching for Critical Cells in the Design

- Critical cells involve
  - DSP48 used as MAC without enabling MREG
    - Report: **dsp48_no_mreg.csv**
  - DSP48 used as MAC/Adder without enabling PREG
    - Report: **dsp48_no_preg.csv**
  - BRAM without enabling DOA_REG or DOB_REG
    - Report: **bram_no_reg.csv**
  - SRL with lower depth (1/2/3)

Symbol	Description	Speed Grade and V_{CCINT} Operating Voltages				Units
		0.90V	0.85V		0.72V	
		-3	-2	-1	-2	
Block RAM and FIFO Clock-to-Out Delays						
T_{RCKO_DO}	Clock CLK to DOUT output (without output register).	0.91	1.02	1.11	1.46	ns, Max
$T_{RCKO\_DO\_REG}$	Clock CLK to DOUT output (with output register).	0.27	0.29	0.30	0.42	ns, Max

Symbol	Description	Speed Grade and V_{CCINT} Operating Voltages				Units
		0.90V	0.85V		0.72V(1)	
		-3	-2	-1	-2	
Maximum Frequency						
F_{MAX}	With all registers used.	891	775	645	644	MHz
F_{MAX_PATDET}	With pattern detector.	794	687	571	562	MHz
$F_{MAX\_MULT\_NOMREG}$	Two register multiply without MREG.	635	544	456	440	MHz
$F_{MAX\_MULT\_NOMREG\_PATDET}$	Two register multiply without MREG with pattern detect.	577	492	410	395	MHz
$F_{MAX\_PREADD\_NOADREG}$	Without ADREG.	655	565	468	453	MHz
$F_{MAX\_NOPIPELINEREG}$	Without pipeline registers (MREG, ADREG).	483	410	338	323	MHz
$F_{MAX\_NOPIPELINEREG\_PATDET}$	Without pipeline registers (MREG, ADREG) with pattern detect.	448	379	314	299	MHz

Control Set

Condition	Typically Acceptable	Analysis Required	Recommended Design Change
Number of Unique Control Sets[a]	< 7.5% of total slices[b]	>15% of total slices[a, b]	>25% of total slices[b] Reducing the number of control sets increases utilization and performance.

```
1 0.#########################################################################
2 1. Number of unique control sets: 15740 --> 10.7% --> Noted
```

This result is available in the Summary.rpt

Constraints Analysis

- Run report_methodology to catch XDC bad practices
- Run write_xdc to catch invalid constraints
  - write_xdc -constraints invalid
- Run report_exceptions for insight on how to reduce constraints size
  - Inefficient timing exceptions waste processing time and memory
  - Use these options to help eliminate unnecessary constraints
    - -ignored: non-existent path, totally overridden by another exception, invalid startpoint/endpoint
    - -ignored_objects: all ignored (invalid) startpoints and endpoints in exceptions
    - -write_valid_exceptions: only exceptions on valid objects, invalid are filtered out
    - -write_merged_exceptions: merged set of timing exceptions seen by the timing engine, including invalid ones
- Four xdc documents will be created
  - invalid_constraints.xdc, ignored_exceptions.xdc
  - ignore_objects_exceptions.xdc, merged_exceptions.xdc

Agenda

- Scanning the analysis items
- Critical items that should be emphasized
- Demo

Summary

Index	Report	Comment
1	clk_networks.rpt	时钟网络报告(report_clock_networks)
2	timing_summary.rpt	时序报告（report_timing_summary)
3	neg_slack_paths.csv	Slack为负的时序路径
4	ClkName_FreqMHz_LL_g_TargetValue.csv	Logic Level大于目标值的时序路径
5	ff2block_ctrl_path_LL_g_0.csv	从FF到Block且扇出大于指定扇出值，Logic Level大于0的时序路径
6	ff2block_ctrl_path_LL_0.csv	从FF到Block且扇出大于指定扇出值，Logic Level等于0的时序路径
7	bram2ff.csv, uram2ff.csv, dsp2ff.csv	从Block到FF的时序路径（Block指BRAM, URAM, DSP）
8	paths_end_srl.csv	终点Cell为SRL的时序路径
9	paths_block2block	Block与Block之间的时序路径
10	paths_ClkName1_Between_ClkName2.csv	MMCM/PLL输入时钟与输出时钟之间的时序路径
11	cdc.rpt	CDC报告（report_cdc.rpt）
12	clk_inter.rpt	时钟交互报告（report_clock_interactions）
13	cong_level.rpt	设计拥塞报告
14	dsp48_no_mreg.csv	用作MAC而MREG未使能的DSP48
15	dsp48_no_preg.csv	用作MAC或ADDER而PREG未使能的DSP48
16	bram_no_reg.csv	输出未寄存的BRAM
17	failing_slr.csv	Slack为负的跨SLR的时序路径
18	high_fanout_nets.rpt	高扇出网络报告
19	gated_clk.rpt	门控时钟报告
20	invalid_constraints.xdc	无效约束报告
21	ignored_exceptions.xdc	被忽略的约束的报告
22	ignored_objects_exceptions.xdc	因objects无法找到而忽略的exception约束的报告
23	merged_exceptions.xdc	合并的exception约束报告
24	utilization.rpt	资源利用率报告
25	Summary.rpt	LUT6/MUXF利用率 + Control Set报告
26	QoR Folder	QoR报告（report_qor_suggestions)
27	ufdm.rpt	UFDM报告（report_methodology）

Reports Format

Timing Paths

File created on Fri Nov 10 10:29:51 +0800 2017
#

Startpoint	Endpoint	Slack	LogicLevel	#Lut	Requirement	PathDelay	LogicDelay	NetDelay	Skew	StartClk	EndClk
cmd_parse_i0/	resp_gen_i0/t	0.713	16	13	10	9.231	1.584(17%)	7.647(83%)	-0.031	clk_rx_clk_	clk_rx_clk_core
cmd_parse_i0/	resp_gen_i0/t	0.778	16	13	10	9.164	1.584(17%)	7.580(83%)	-0.031	clk_rx_clk_	clk_rx_clk_core
cmd_parse_i0/	resp_gen_i0/t	0.835	15	12	10	9.056	1.531(17%)	7.525(83%)	-0.031	clk_rx_clk_	clk_rx_clk_core
cmd_parse_i0/	resp_gen_i0/t	1.003	15	12	10	8.949	1.531(17%)	7.418(83%)	-0.023	clk_rx_clk_	clk_rx_clk_core
cmd_parse_i0/	resp_gen_i0/t	1.196	15	12	10	8.754	1.531(17%)	7.223(83%)	-0.023	clk_rx_clk_	clk_rx_clk_core
cmd_parse_i0/	resp_gen_i0/t	1.198	15	12	10	8.753	1.531(17%)	7.222(83%)	-0.023	clk_rx_clk_	clk_rx_clk_core

BRAM

File created on Fri Nov 10 10:42:14 +0800 2017
#

BRAM	CLKA	CLKA_FREQ	CLKB	CLKB_FREQ	DOA_REG	DOA_CONNECTED	DOB_REG	DOB_CONNECTED
char_fif(	clk_rx_(	200	clk_tx_(	166.67	0	0	0	1
samp_ram_	clk_rx_(	200	clk_tx_(	0	0	1	1	1

DSP48

File created on Thu Nov 09 07:52:28 MST 2017
#

ClkName	ClkFreq	Cell
clk_sys	166.64	u_lane_allip_top/lane_gen[0].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu_0/
clk_sys	166.64	u_lane_allip_top/lane_gen[1].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu_0/
clk_sys	166.64	u_lane_allip_top/lane_gen[2].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu_0/
clk_sys	166.64	u_lane_allip_top/lane_gen[3].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu_0/
clk_sys	166.64	u_lane_allip_top_u2/lane_gen[0].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu
clk_sys	166.64	u_lane_allip_top_u2/lane_gen[1].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu
clk_sys	166.64	u_lane_allip_top_u2/lane_gen[2].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu
clk_sys	166.64	u_lane_allip_top_u2/lane_gen[3].lane_100ge_gen.lane_100ge_allip_wrapper/otn_sub_100gbe_map_liu

Thank you!