Implementation and STA

Objectives

- **After completing this module, you will be able to:**
  - Implement a design to completion
  - Generate useful implementation reports
  - Describe static timing analysis and static timing paths
  - Describe setup and hold checks
  - Understand the relationship between clocks and setup and hold checks
  - Generate a custom timing report to perform basic static timing analysis
  - Use the timing summary report to verify your timing constraints were met
  - Generate a bitstream and verify the functionality in hardware

Outline

- *Implementation*
- Reports
- Basic Static Timing Analysis
- Bitstream Generation and Verification in Hardware
- Summary

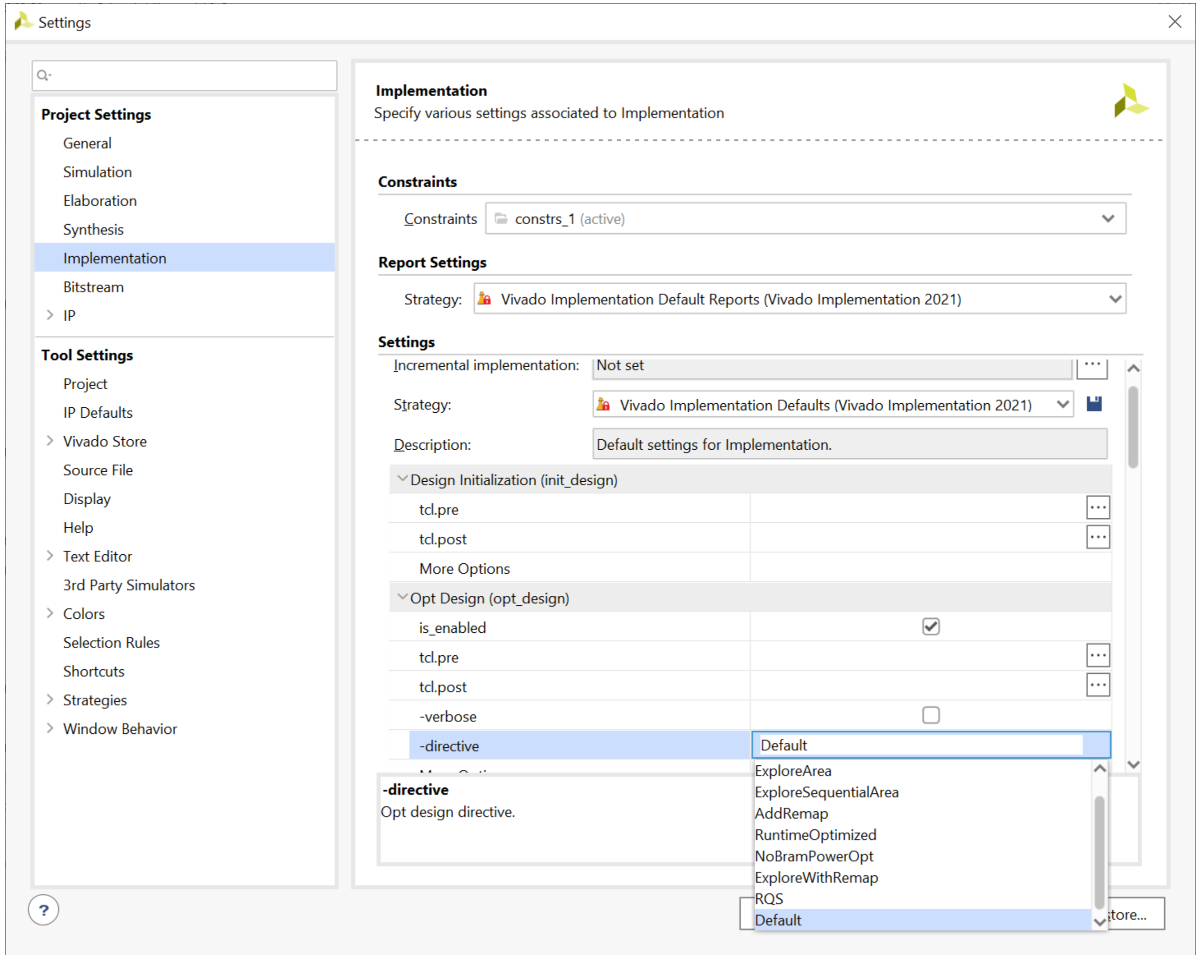
Vivado Implementation Phases

Vivado Tools Implementation Flow: Tcl Commands	
link_design*	Translate design, apply constraints
opt_design*	Logic Optimization
power_opt_design*	Power Optimization
place_design	Placement
phys_opt_design*	Physical Synthesis
route_design	Routing
report_timing_summary	Timing Analysis
write_bitstream	Bitstream Generation

* Optional Command

opt_design: Logic Optimization

- **Ensures optimal netlist for placement**
  - Further logic optimization on fully-assembled netlist built from synthesized RTL, IP blocks
    - Performs logic trimming on incoming netlist
    - Constant propagation – remove unnecessary static logic
    - LUT equation remapping
- **Optional in non-project batch flow (but recommended)**
  - Example: needed to trim unused bank cells in MIG IP (phaser/iodelay/….)
- **Automatically enabled in the project-based flow**



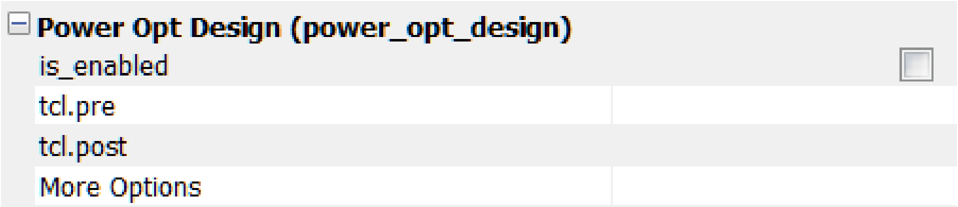
opt_design Options

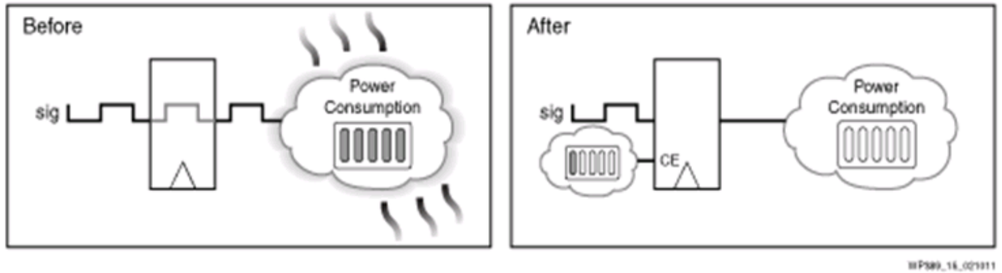
- **Non-project batch flow**
  - Can specify which optimizations to perform from a script

```
opt_design

Description:
Optimize the current netlist. This will perform the retarget, propconst, sweep and bram_power_opt optimizations by default.

Syntax:
opt_design  [-retarget] [-propconst] [-sweep] [-bram_power_opt] [-remap]
            [-resynth_area] [-resynth_seq_area] [-directive <arg>] [-quiet]
            [-verbose]

Usage:
  Name                 Description
  --------------------------------
  [-retarget]          Retarget
  [-propconst]         Propagate constants across leaf-level instances
  [-sweep]             Remove unconnected leaf-level instances
  [-bram_power_opt]    Perform Block RAM power optimizations
  [-remap]             Remap logic optimally in LUTs
  [-resynth_area]      Resynthesis
  [-resynth_seq_area]  Resynthesis (with Sequential optimizations)
  [-directive]         Mode of behavior (directive) for this command. Please
                       refer to Arguments section of this help for values for
                       this option
                       Default: Default
  [-quiet]             Ignore command errors
  [-verbose]           Suspend message limits during command execution
```

power_opt: Power Optimization

- **Power optimization includes a fine-grained clock gating solution that can reduce dynamic power by up to 30%**
- **Intelligent clock gating optimizations are automatically performed on the entire design and will generate no changes to the existing logic or clocks**
- **Algorithm performs analysis on all portions of the design**
  - Legacy and third-party IP blocks

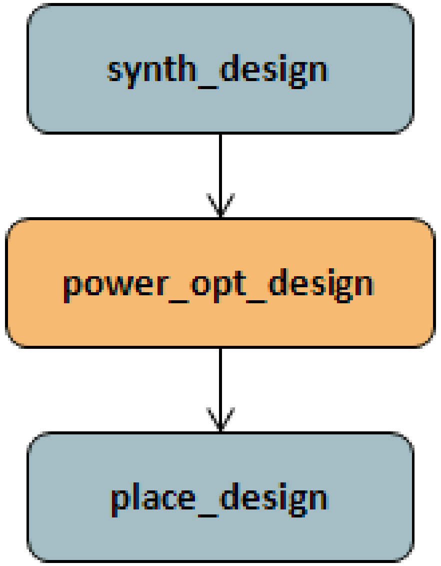




Power Optimization Commands

- **Automatic power reduction**
  - Automatically turns off unused portions of the design
  - Does not require deep system level knowledge
- **Vivado IDE provides optimization control at global and object level**
  - Global command for optimizing the design:
    `power_opt_design`
  - Local level control through SDC command:

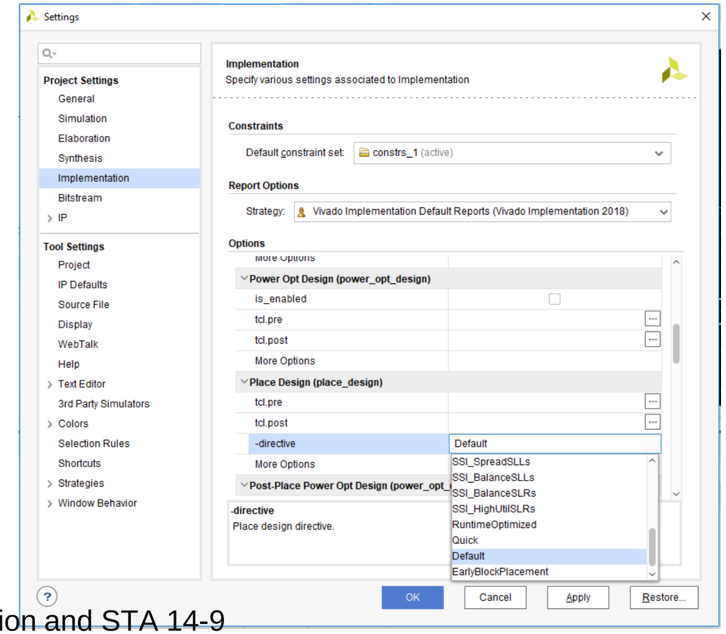
    `set_power_opt`
    - Instance level: Include/exclude instances for power opt
    - Clock domain: Optimize instances clocked by the specified clock
    - Cell-type level: Block RAM, registers, SRL



place_design: placer

- **Project-based flow**
  - Included in implementation stage
- **Non-project batch flow**

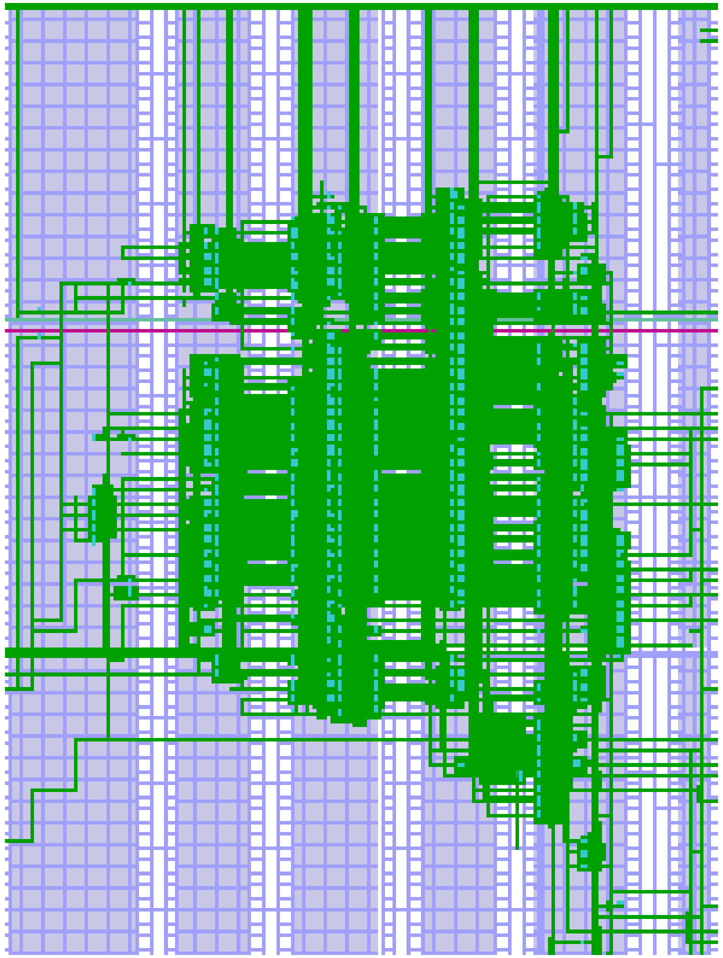
  `place_design`
- **Can use an input XDEF as a starting point for placement**




```
place_design

Description:
Automatically place ports and leaf-level instances

Syntax:
place_design  [-directive <arg>] [-no_timing_driven] [-timing_summary]
              [-unplace] [-post_place_opt] [-no_psip] [-no_bufg_opt]
              [-ultrathreads] [-quiet] [-verbose]

Usage:
  Name                Description
  --------------------------------
  [-directive]        Mode of behavior (directive) for this command. Please
                      refer to Arguments section of this help for values for
                      this option.
                      Default: Default
  [-no_timing_driven] Do not run in timing driven mode
  [-timing_summary]   Enable accurate post-placement timing summary.
  [-unplace]          Unplace all the instances which are not locked by
                      Constraints.
  [-post_place_opt]   Run only the post commit optimizer
  [-no_psip]          Disable PSIP (Physical Synthesis In Placer)
                      optimization during placement.
  [-no_bufg_opt]      Disable global buffer insertion during placement
  [-ultrathreads]     Enable ultra-threads mode to speed up place_design
  [-quiet]            Ignore command errors
  [-verbose]          Suspend message limits during command execution
```

Placement

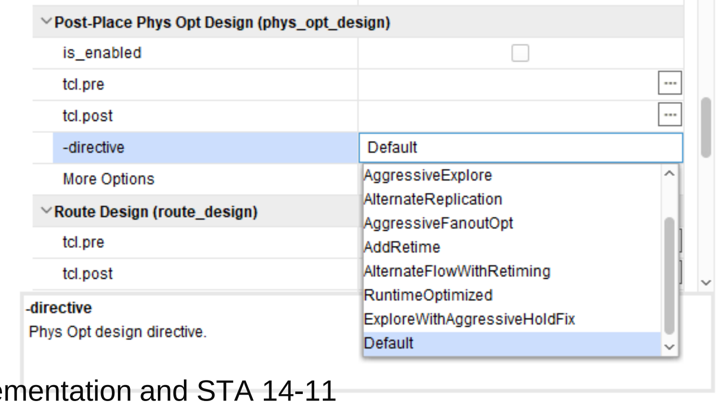
- **Phases of a complete placement**
  - Pre-placement DRC
    - Check for unroutable connections, valid physical constraints, overutilization
  - Placement
    - I/O and clock placement
- **Macro and primitive placement**
  - Timing-driven and wire length-driven
  - Congestion-aware
- **Detailed placement**
  - Refine locations of small "shapes," flip-flops, LUTs
  - Commit to location sites – pack into slices
- **Post-commit optimizations**



phys_opt_design: Physical Synthesis

- **Post-placement timing-driven optimization**
  - Replicates and places drivers of high fanout nets with negative slack
    - Replication only performed if it improves timing
    - Slack must be within critical range
      - Approximately 10% of worst negative slack (WNS)
- **Available in all flows and can be de-activated in the GUI**

  `phys_opt_design`
  - Run between `place_design` and `route_design`




```
phys_opt_design

Description:
Optimize the current placed netlist.

Syntax:
phys_opt_design  [-fanout_opt] [-placement_opt] [-routing_opt]
                 [-slr_crossing_opt] [-rewire] [-insert_negative_edge_ffs]
                 [-critical_cell_opt] [-dsp_register_opt] [-bram_register_opt]
                 [-uram_register_opt] [-bram_enable_opt] [-shift_register_opt]
                 [-hold_fix] [-aggressive_hold_fix] [-retime]
                 [-force_replication_on_nets <args>] [-directive <arg>]
                 [-critical_pin_opt] [-clock_opt] [-path_groups <args>]
                 [-tns_cleanup] [-sll_reg_hold_fix] [-quiet] [-verbose]
```

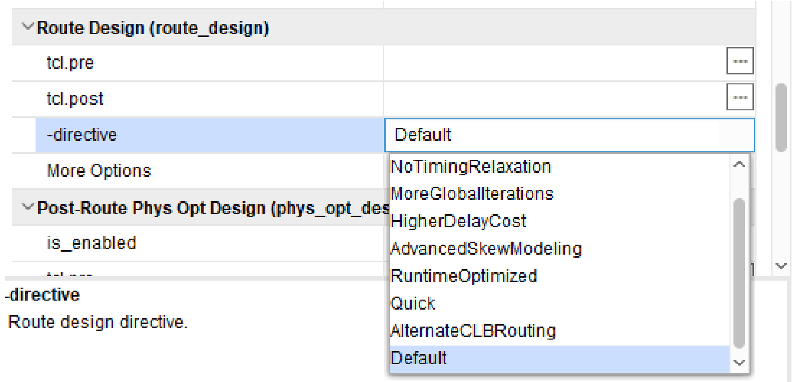
```
Usage:
  Name                         Description
  ------------------------------------------
  [-fanout_opt]                Do cell-duplication based optimization on
                               high-fanout timing critical nets
  [-placement_opt]             Do placement based optimization on timing
                               critical nets
  [-routing_opt]               Do routing based optimization on timing
                               critical nets
  [-slr_crossing_opt]          Do placement optimization of SLR-crossing
                               timing critical nets
  [-rewire]                    Do rewiring optimization
  [-insert_negative_edge_ffs]  Insert negative edge triggered FFs for hold
                               optimization
  [-critical_cell_opt]         Do cell-duplication based optimization on
                               timing critical nets
  [-dsp_register_opt]          Do DSP register optimization
  [-bram_register_opt]         Do BRAM register optimization
  [-uram_register_opt]         Do UltraRAM register optimization
  [-bram_enable_opt]           Do BRAM enable optimization
  [-shift_register_opt]        Do Shift register optimization
  [-hold_fix]                  Attempt to improve slack of high hold
                               violators
  [-aggressive_hold_fix]       Attempt to aggressively improve slack of high
                               hold violators
  [-retime]                    Do retiming optimization
  [-force_replication_on_nets] Force replication optimization on nets
  [-directive]                 Mode of behavior (directive) for this
                               command. Please refer to Arguments section of
                               this help for values for this option
                               Default: Default
  [-critical_pin_opt]          Do pin-swapping based optimization on timing
                               critical nets
  [-clock_opt]                 Do clock skew optimization in post-route
                               optimization
  [-path_groups]               Work only on specified path groups
  [-tns_cleanup]               Work on all nets in the design that meet
                               criteria for the specified optimizations to
                               improve design tns
  [-sll_reg_hold_fix]          Do hold fixing on SLL Tx-Rx paths
  [-quiet]                     Ignore command errors
  [-verbose]                   Suspend message limits during command
                               execution
```

route_design: Router

- **Project-based flow**
  - Included in implementation stage
- **Non-project batch flow**

  `route_design`
- **Router reporting**

  `report_route_status` command
  - Check route status of individual nets
    - Fully routed: lists routing resources
    - Failed routes




```
route_design

Description:
Route the current design

Syntax:
route_design  [-unroute] [-release_memory] [-nets <args>] [-physical_nets]
              [-pins <arg>] [-directive <arg>] [-tns_cleanup]
              [-no_timing_driven] [-preserve] [-delay] [-auto_delay]
              -max_delay <arg> -min_delay <arg> [-timing_summary] [-finalize]
              [-ultrathreads] [-eco] [-quiet] [-verbose]

Usage:
  Name                Description
  ----------------------------------
  [-unroute]          Unroute whole design or the given nets/pins if used
                      with -nets or -pins.
  [-release_memory]   Release Router memory. Not compatible with any other
                      options.
  [-nets]             Operate on the given nets.
  [-physical_nets]    Operate on all physical nets.
  [-pins]             Operate on the given pins.
  [-directive]        Mode of behavior (directive) for this command. Please
                      refer to Arguments section of this help for values for
                      this option.
                      Default: Default
  [-tns_cleanup]      Do optional TNS clean up.
  [-no_timing_driven] Do not run in timing driven mode.
  [-preserve]         Preserve existing routing.
  [-delay]            Use with -nets or -pins option to route in delay
                      driven mode.
  [-auto_delay]       Use with -nets or -pins option to route in constraint
                      driven mode.
  -max_delay          Use with -pins option to specify the max_delay
                      constraint on the pins.When specified -delay is
                      implicit.
  -min_delay          Use with -pins option to specify the max_delay
                      constraint on the pins.When specified -delay is
                      implicit.
  [-timing_summary]   Enable post-router signoff timing summary.
  [-finalize]         finalize route_design in interactive mode.
  [-ultrathreads]     Enable Turbo mode routing.
  [-eco]              runs incremental router if there was eco modification
                      on routed netlist.
  [-quiet]            Ignore command errors
  [-verbose]          Suspend message limits during command execution
```

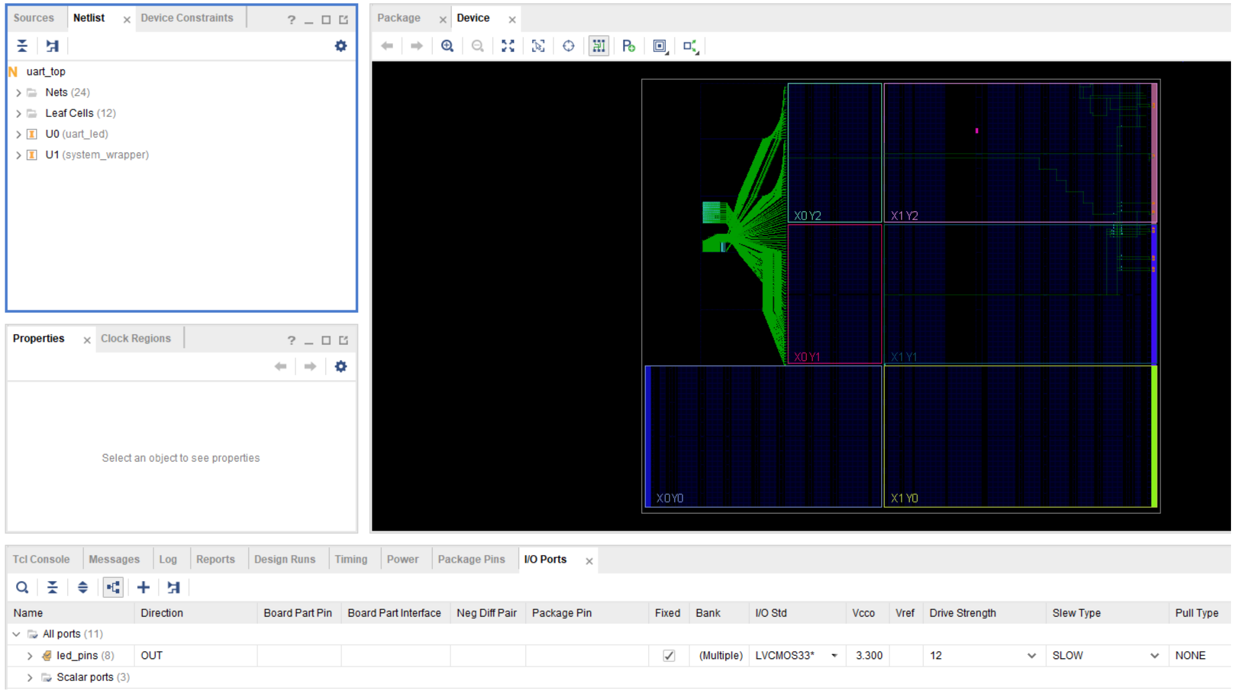
Router

- **Phases of a complete route**
  - Special net and clock routing
  - Timing-driven routing
    - Prioritized by setup/hold path criticality
    - Swap LUT inputs pin to improve critical paths
    - Fix reasonable hold time violations
- **Two modes**
  - Normal (default): Router starts with a placed design and attempts to route all nets
  - Re-Entrant (non-project batch only): Router can route/unroute as well as lock/unlock specific nets

Reports

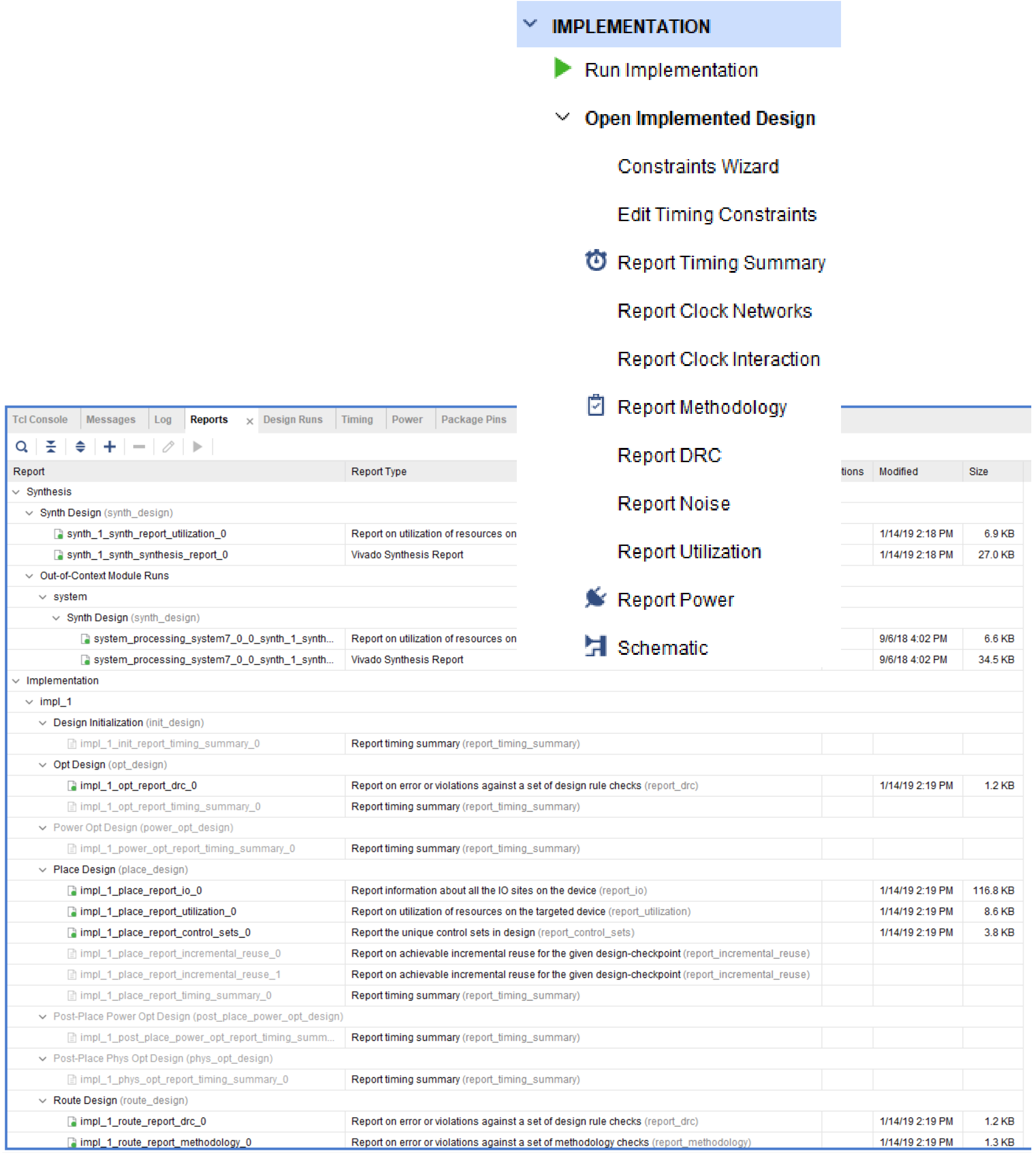
After Implementation

- **Sources and Netlist tabs do not change**
  - As each resources is selected, it will show the exact placement of the resource on the die
- **Timing results have to be generated with the Report Timing Summary**
- **As each path is selected, the placement of the logic and its connections is shown in the Device view**
  - This is the cross-probing feature that helps with static timing analysis



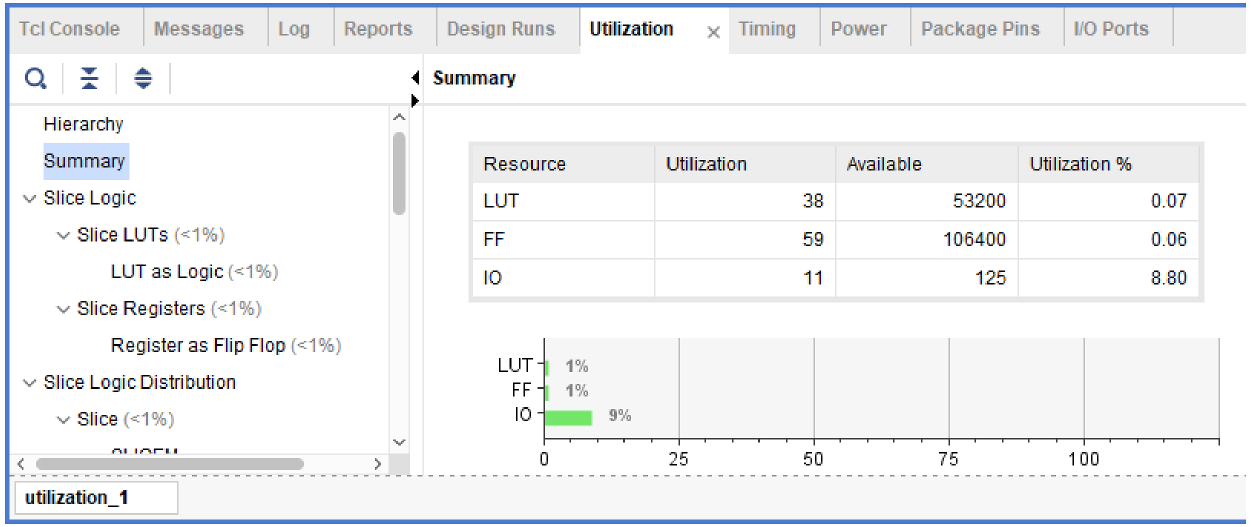
Implementation Reports

- While the Flow Navigator points to the most important reports, the Reports tab contains several other useful reports
  - Post Optimization DRC Report: Lists the I/O DRC checks that were completed
  - Post Power Optimization DRC Report: Lists the power DRC checks that were completed
  - Place and Route Log: Describes the implementation process and any issues it encountered
  - IO Report: Lists the final pinout for your design
  - Clock Utilization Report: Describes the clock resources used and the clock utilization resource on a region-by-region basis
  - Utilization Report: Describes the amount of FPGA resources used in a text format
  - Control Sets Report: describes how your control signals were grouped



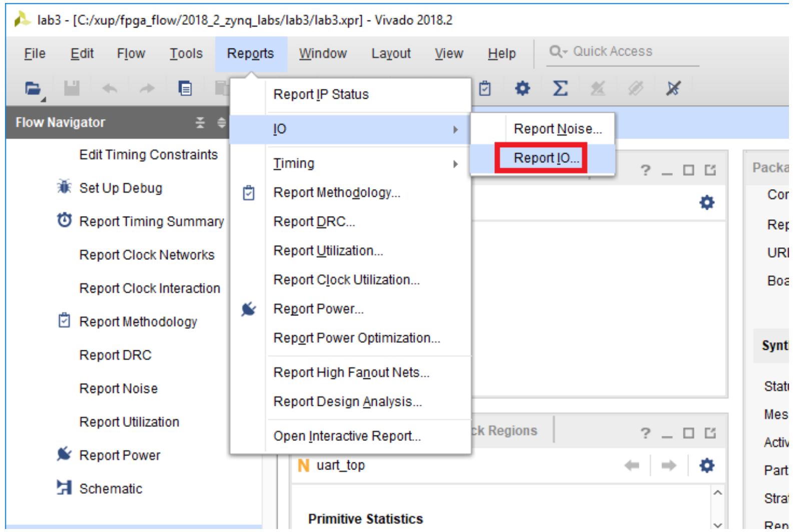
Utilization Reports

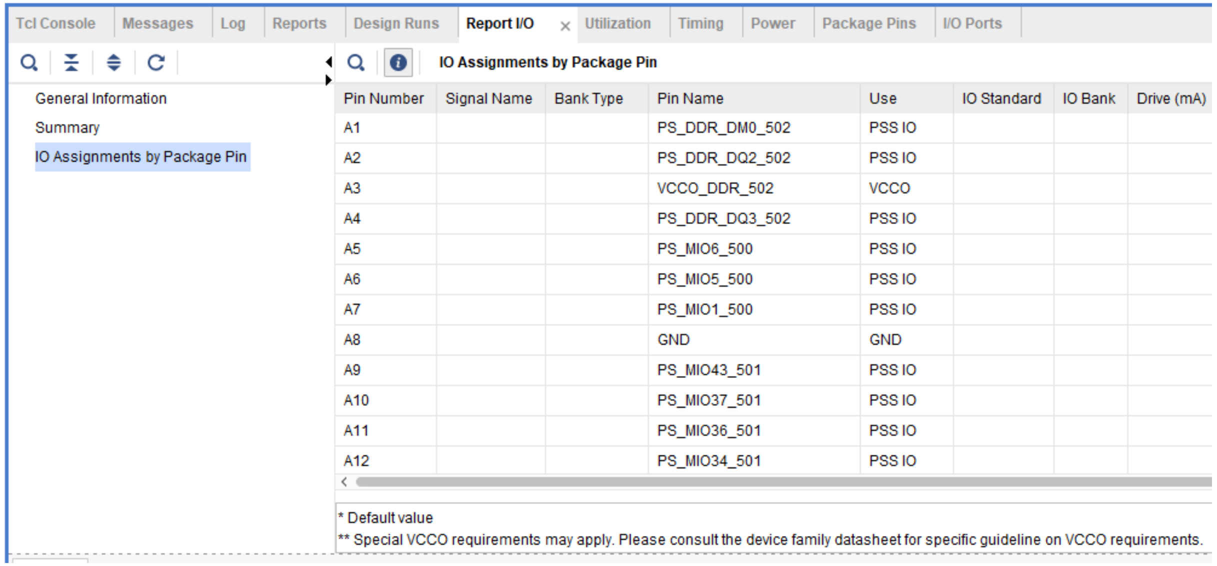
- Double-click in the Reports tab to view in text form
- Click on Report Utilization under Implementation Result in the Flow Navigator pane to view in chart and table format



I/O Report

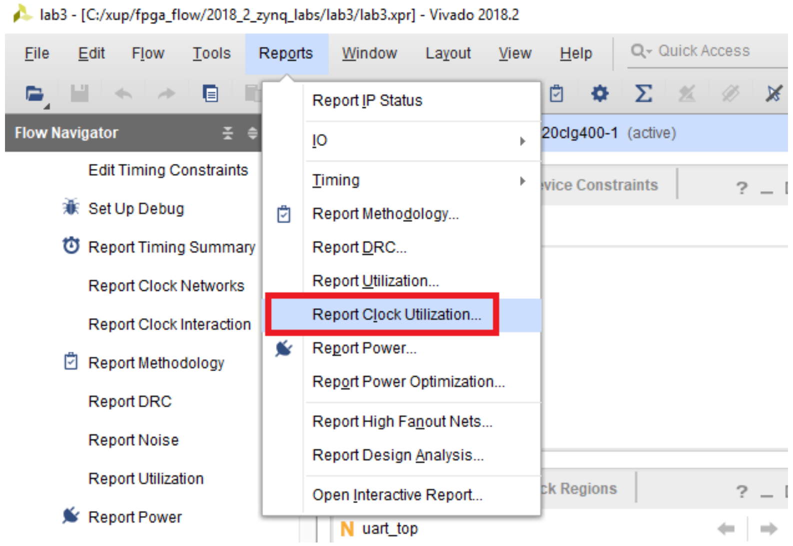
- **This report provides a table that lists every signal, its attributes, and its final location**
  - It is always important to double-check pin assignments before implementing because the tools can move any pin that is unassigned

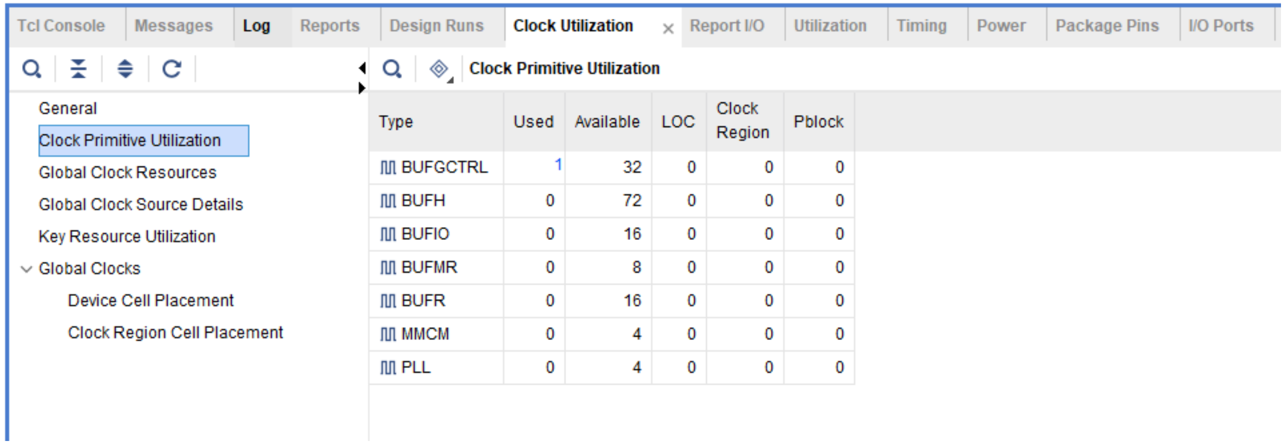




Clock Utilization Report

- This report describes the clocking resources used in the design
  - BUFG, BUFH, BUFHCE, MMCM, and a clock region analysis

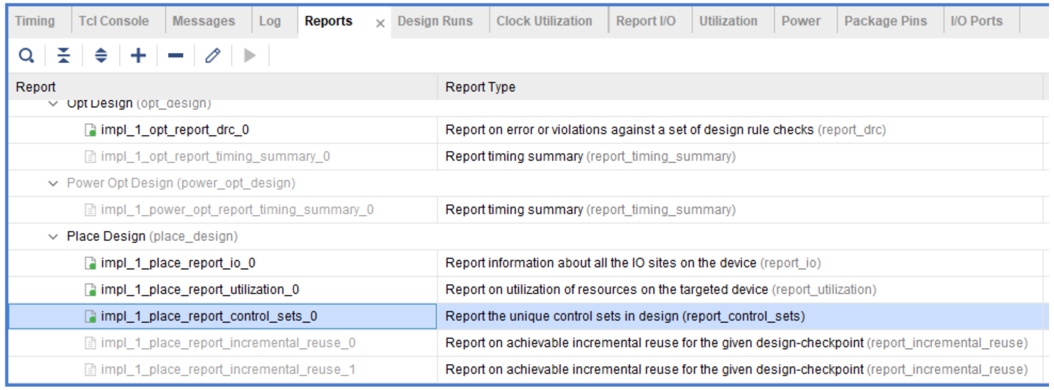


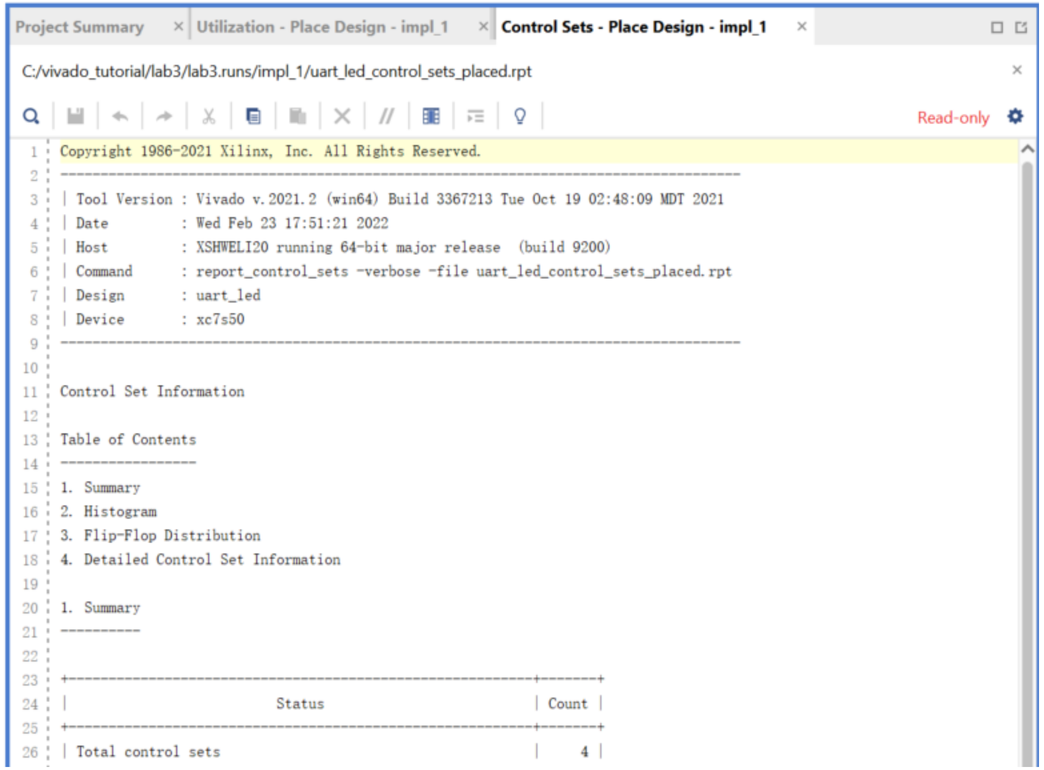


Type	Used	Available	LOC	Clock Region	Pblock
BUFGCTRL	1	32	0	0	0
BUFH	0	72	0	0	0
BUFIO	0	16	0	0	0
BUFMR	0	8	0	0	0
BUFR	0	16	0	0	0
MMCM	0	4	0	0	0
PLL	0	4	0	0	0

Control Sets Report

- **This report describes the number of unique control sets in the design (ideally this will be as small as possible)**
- **Number of control sets describes how control signals were grouped**
  - This determines the ability of the tools to reach high device utilization
  - Number of controls signals in the design is determined by the designer's inference of sets, resets, and clock enable signals
  - Number of control signals can be reduced if the designer attempts to share controls signals throughout the design as much as possible





power_opt: Report

- **`report_power_opt` command details the gating performed on the design by the power_opt program**
  - Can be run before power optimization to create a report detailing all the user-gated logic in the design for a before and after picture of your design

Basic Static Timing Analysis

Static Timing Analysis (STA)

- **A design is an interconnected set of cells and nets**
- **The functionality of a design is determined by the RTL design sources**
  - The functionality can be verified by a simulation tool
- **The performance of a device is determined by the delays of the cells that comprise the design**
  - This is verified by static timing analysis
- **In STA the functionality of the components of the design are not important**
  - Only the performance of the components
  - Cells need only be classified as combinatorial or sequential

Component Delays

- **Each component takes time to perform its function**
  - A LUT has propagation delay from its inputs to its outputs
  - A net has propagation delay from the driver to the receiver(s)
  - A flip-flop requires stable data for a required time around its sample point
- **These delays are dependent on a several factors**
  - Some are determined by the composition of the FPGA and the implementation of the design
    - The physical characteristics of the element (how it is constructed)
    - The location of the object (where it is placed with respect to other objects)
  - Some are determined by environmental factors (PVT)
    - The process variation of the device
    - The voltage applied to the cells
    - The temperature of the cells

Delays

- **Component and net delays are provided by Xilinx, and are extracted by careful characterization of production devices**
  - Timing is extracted over the allowed operating range of the device
    - Process is within a specific range
      - Different ranges are used for different speed grades (-1, -2, -3)
    - Voltage is between the minimum and maximum allowed for the device
      - Different speed grades may allow different voltages (i.e. -1L)
    - Temperature is between the maximum and minimum specified
      - Commercial and industrial parts allow different temperature ranges
  - These characterized delays may be extracted at various process corners
    - Fastest PVT, slowest PVT
  - The characterized delays at the appropriate corner are used by the tools during STA

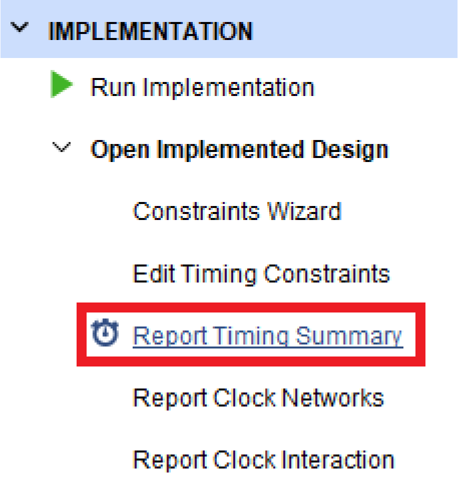
AMD XILINX

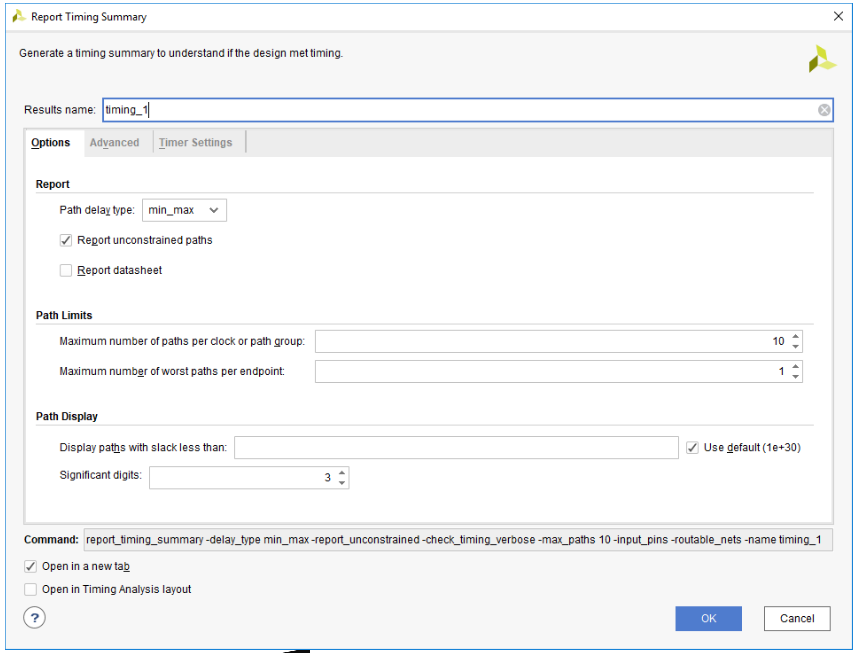
Why Do We Need STA?

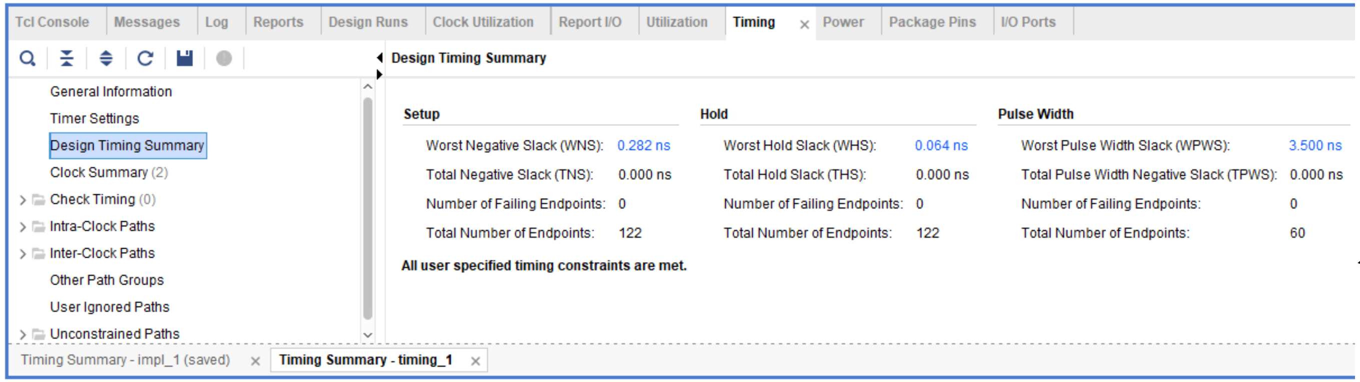
- **Many of the processes in FPGA implementation are timing driven**
  - Synthesis for circuit construction
  - Placer for optimal cell locations
  - Router for choosing routing elements
- **Tools must have constraints to determine the desired performance goals**
- **STA is used during the processes, and afterwards for generating reports**
- **Ultimately, STA determines if a design will provide the desired performance**

report_timing_summary: Timing Report Summary

- Timing reports with true timing information can be generated from the Flow Navigator after implementation has been completed





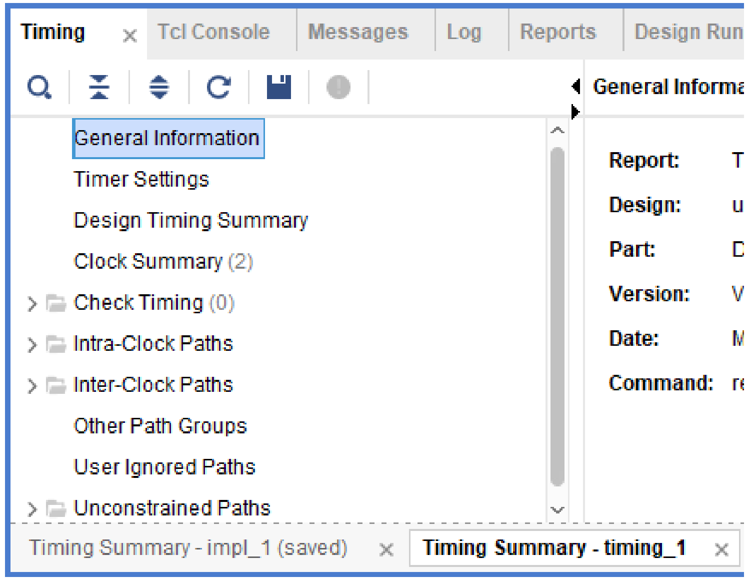


Timing Summary

- **Summary reports Setup, Hold, and Pulse Width related results**
  - Setup
    - Worst Negative Slack (WNS): The worst slack of all the timing paths for max delay analysis. It can be positive or negative; positive means no violation
    - Total Negative Slack (TNS): The sum of all WNS violations, when considering only the worst violation of each timing path endpoint- 0ns when all timing constraints are met, Negative when there are some violations
    - Number of Failing Endpoints: The total number of endpoints with a violation (WNS<0ns)
  - Hold
    - Worst Hold Slack (WHS): Corresponds to the worst slack of all the timing paths for min delay analysis. It can be positive or negative
  - Pulse Width
    - Worst Pulse Width Slack (WPWS): Corresponds to the worst slack of all the timing checks listed above when using both min and max delays

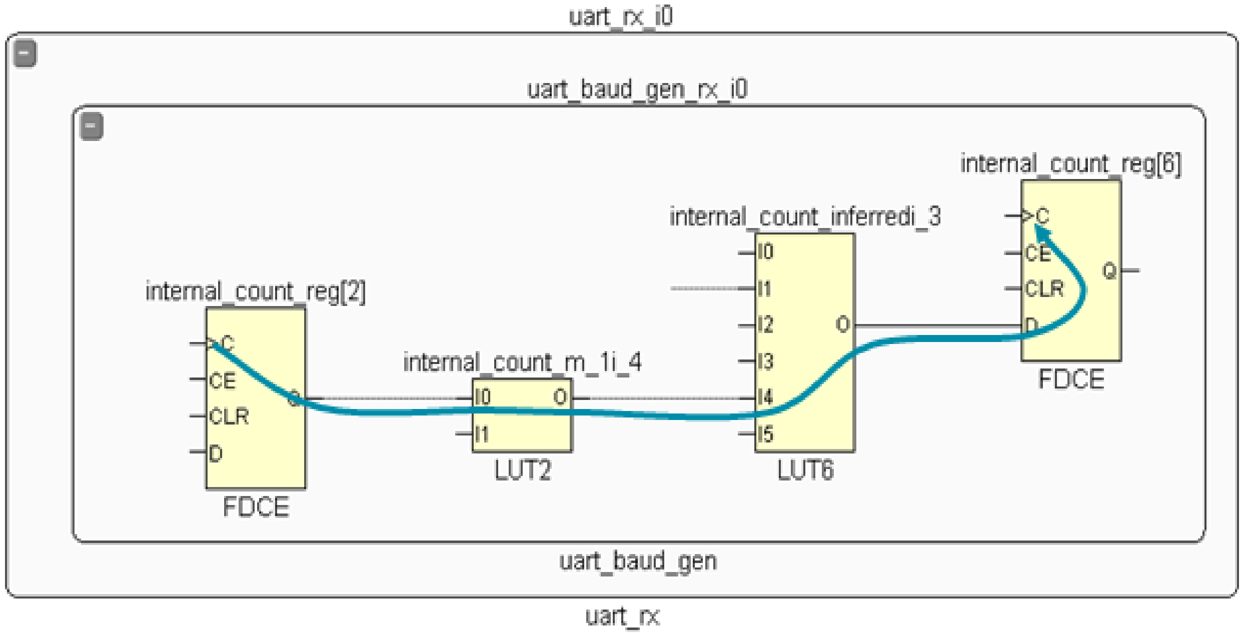
The Timing Summary Table

- **The Timing Summary table shows**
  - General Information provides design name, device, package, speed grades etc.
  - Timer Settings provides timing analysis engine settings
  - Design Timing Summary provides a summary of all the timing reports
  - Clock Summary includes information similar to that produced by report_clocks
  - Check Timing includes information about missing timing constraints or paths with constraints issues that need to be reviewed
  - Intra-Clock Paths includes summary of the worst slack and total violations of the timing paths with the same source and destination clocks
  - Inter-Clock Paths includes summary of the worst slack and total violations of timing paths with different source and destination clocks
  - Other Path Groups displays paths not covered above, including user-defined path groups
  - User Ignored Paths are paths that are ignored during timing analysis
  - Unconstrained Paths contain paths that were not covered by the XDC constraints
  - Violations are displayed in red



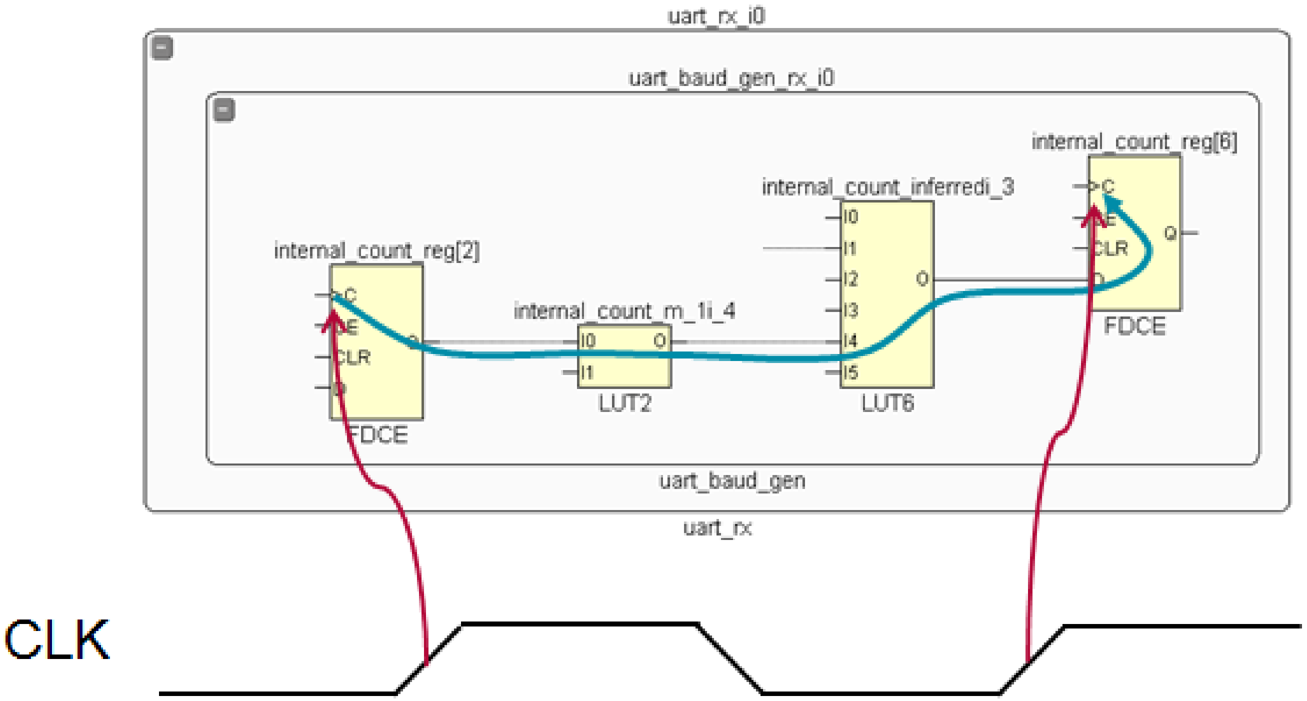
Static Timing Paths

- **A static timing path is a path that**
  - Starts at a clocked element
  - Propagates through any number of combinatorial elements and the nets that interconnect them
  - Ends at a clocked element
- **Clocked elements include flip-flops, block RAMs, DSP cells, …**
- **Combinatorial elements include LUTs, wide MUXes, carry chains, …**



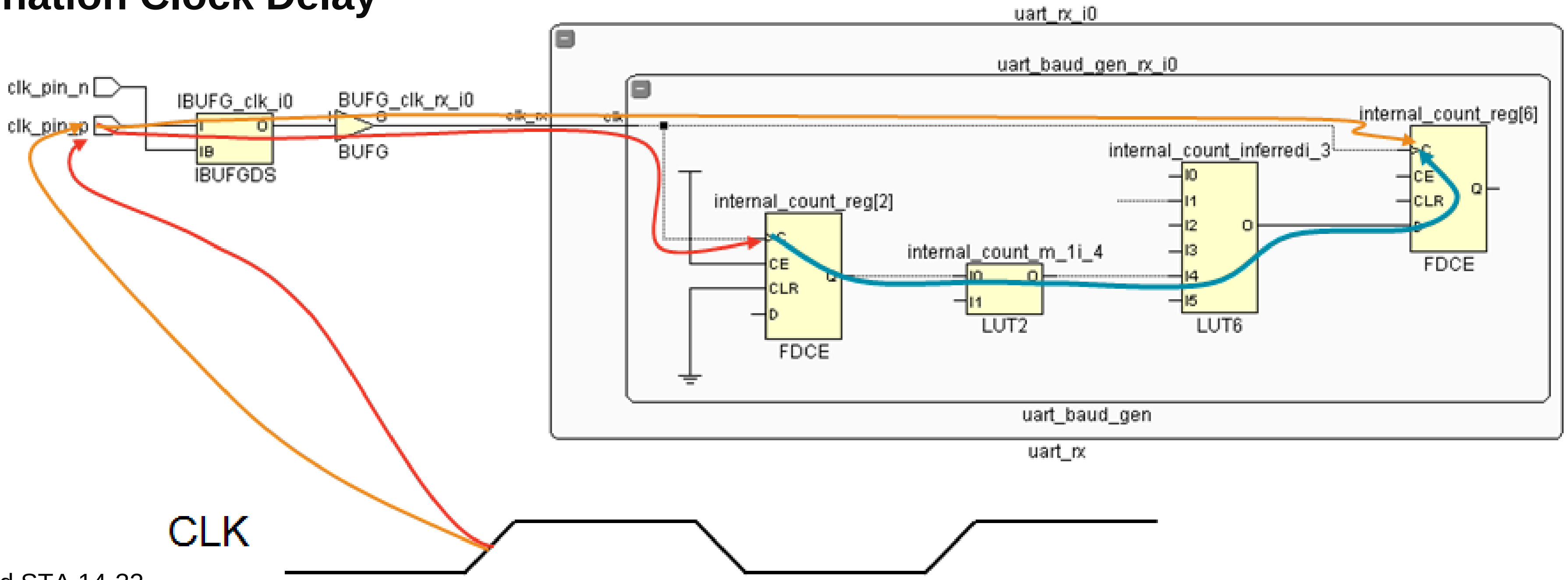
Setup Check

- **Checks that a change in a clocked element has time to propagate to other clocked elements before the next clock event**
  - That is, from the rising edge of the clock to the next rising edge of the clock
  - Checked for all static timing paths



Hold Check

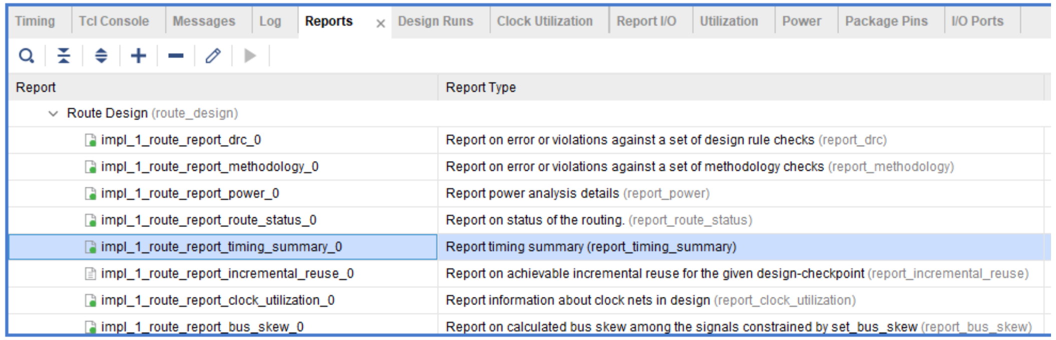
- **Checks that a change in a clocked element caused by a clock event does not propagate to a destination clocked element before the same clock event arrives at the destination element**
  - Usually from the rising edge of the clock to the same edge of the clock
  - Checked for all static timing paths
- **The shortest delay is used for Source Clock and Data Path Delay, and the longest delay is used for Destination Clock Delay**

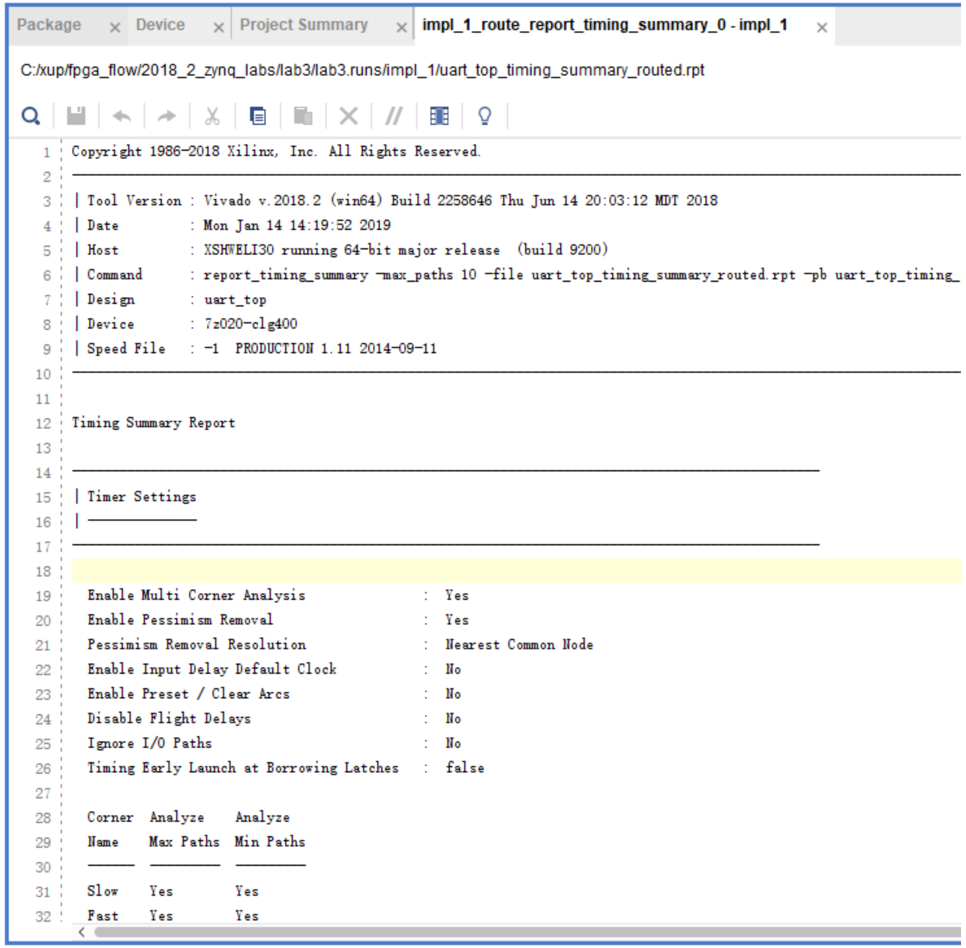


AMD XILINX

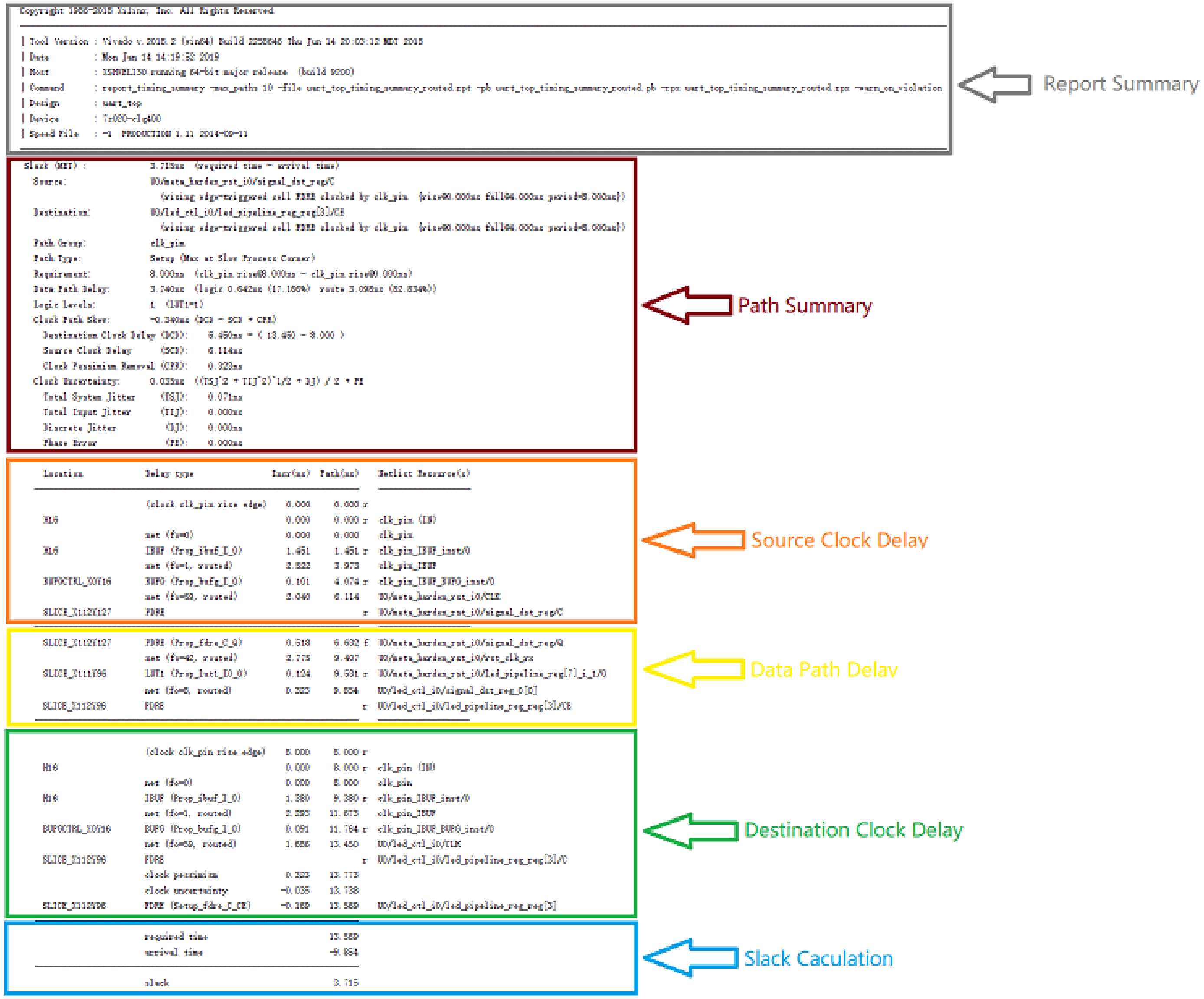
report_timing_summary: TRCE Like Report

- **`report_timing_summary` command produces a comprehensive TRCE-like timing report**
- This report contains these sections
  - Timer Settings
  - check_timing report
  - Design Timing Summary
  - Clock Definitions
  - Intra Clock table
  - Inter Clock table
  - Path Group table





Anatomy of Timing Report



```
Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
------------------------------------------------------------------------------------
| Tool Version : Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
| Date         : Mon Jun 14 14:19:52 2019
| Host         : XSNVELI30 running 64-bit major release  (build 9200)
| Command      : report_timing_summary -max_paths 10 -file uart_top_timing_summary_routed.rpt -pb uart_top_timing_summary_routed.pb -rpx uart_top_timing_summary_routed.rpx -warn_on_violation
| Design       : uart_top
| Device       : 7z020-clg400
| Speed File   : -1  PRODUCTION 1.11 2014-09-11
------------------------------------------------------------------------------------
Slack (MET) :             3.715ns  (required time - arrival time)
  Source:                 U0/meta_harden_rst_i0/signal_dst_reg/C
                            (rising edge-triggered cell FDRE clocked by clk_pin  {rise@0.000ns fall@4.000ns period=8.000ns})
  Destination:            U0/led_ctl_i0/led_pipeline_reg_reg[3]/CE
                            (rising edge-triggered cell FDRE clocked by clk_pin  {rise@0.000ns fall@4.000ns period=8.000ns})
  Path Group:             clk_pin
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            8.000ns  (clk_pin rise@8.000ns - clk_pin rise@0.000ns)
  Data Path Delay:        3.740ns  (logic 0.642ns (17.166%)  route 3.098ns (82.834%))
  Logic Levels:           1  (LUT1=1)
  Clock Path Skew:        -0.340ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    5.450ns = ( 13.450 - 8.000 )
    Source Clock Delay      (SCD):    6.114ns
    Clock Pessimism Removal (CPR):    0.323ns
  Clock Uncertainty:      0.035ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.000ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock clk_pin rise edge)    0.000     0.000 r
    H16                                               0.000     0.000 r  clk_pin (IN)
                         net (fo=0)                   0.000     0.000    clk_pin
    H16                  IBUF (Prop_ibuf_I_O)         1.451     1.451 r  clk_pin_IBUF_inst/O
                         net (fo=1, routed)           2.522     3.973    clk_pin_IBUF
    BUFGCTRL_X0Y16       BUFG (Prop_bufg_I_O)         0.101     4.074 r  clk_pin_IBUF_BUFG_inst/O
                         net (fo=59, routed)          2.040     6.114    U0/meta_harden_rst_i0/CLK
    SLICE_X112Y127       FDRE                                         r  U0/meta_harden_rst_i0/signal_dst_reg/C
  -------------------------------------------------------------------    -------------------
    SLICE_X112Y127       FDRE (Prop_fdre_C_Q)         0.518     6.632 f  U0/meta_harden_rst_i0/signal_dst_reg/Q
                         net (fo=42, routed)          2.775     9.407    U0/meta_harden_rst_i0/rst_clk_rx
    SLICE_X111Y96        LUT1 (Prop_lut1_I0_O)        0.124     9.531 r  U0/meta_harden_rst_i0/led_pipeline_reg[7]_i_1/O
                         net (fo=8, routed)           0.323     9.854    U0/led_ctl_i0/signal_dst_reg_0[0]
    SLICE_X112Y96        FDRE                                         r  U0/led_ctl_i0/led_pipeline_reg_reg[3]/CE
  -------------------------------------------------------------------    -------------------

                         (clock clk_pin rise edge)    8.000     8.000 r
    H16                                               0.000     8.000 r  clk_pin (IN)
                         net (fo=0)                   0.000     8.000    clk_pin
    H16                  IBUF (Prop_ibuf_I_O)         1.380     9.380 r  clk_pin_IBUF_inst/O
                         net (fo=1, routed)           2.293    11.673    clk_pin_IBUF
    BUFGCTRL_X0Y16       BUFG (Prop_bufg_I_O)         0.091    11.764 r  clk_pin_IBUF_BUFG_inst/O
                         net (fo=59, routed)          1.686    13.450    U0/led_ctl_i0/CLK
    SLICE_X112Y96        FDRE                                         r  U0/led_ctl_i0/led_pipeline_reg_reg[3]/C
                         clock pessimism              0.323    13.773
                         clock uncertainty           -0.035    13.738
    SLICE_X112Y96        FDRE (Setup_fdre_C_CE)      -0.169    13.569    U0/led_ctl_i0/led_pipeline_reg_reg[3]
  -------------------------------------------------------------------
                         required time                         13.569
                         arrival time                          -9.854
  -------------------------------------------------------------------
                         slack                                  3.715
```


AMD
XILINX

Report Sections

- Report summary

```
Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
------------------------------------------------------------------------------------------------------------------------
| Tool Version : Vivado v.2018.2 (win64) Build 2258646 Thu Jun 14 20:03:12 MDT 2018
| Date         : Mon Jan 14 14:19:52 2019
| Host         : XSHWELI30 running 64-bit major release  (build 9200)
| Command      : report_timing_summary -max_paths 10 -file uart_top_timing_summary_routed.rpt -pb uart_top_timing_summary_routed.pb -rpx uart_top_timing_summary_routed.rpx -warn_on_violation
| Design       : uart_top
| Device       : 7z020-clg400
| Speed File   : -1  PRODUCTION 1.11 2014-09-11
------------------------------------------------------------------------------------------------------------------------
```

- Design timing summary

```
------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints     WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  -------------------     --------     --------  ----------------------  --------------------
      0.282        0.000                      0                  122        0.064        0.000                      0                  122        3.500        0.000                       0                    60
```

- Clock summary

```
------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------

Clock          Waveform(ns)       Period(ns)      Frequency(MHz)
-----          ------------       ----------      --------------
clk_pin        {0.000 4.000}      8.000           125.000
virtual_clock  {0.000 4.000}      8.000           125.000
```

Report Sections

- Intra clock table

```
--------------------------------------------------------------------------------
| Intra Clock Table
| -----------------
--------------------------------------------------------------------------------

Clock       WNS(ns)   TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints  WHS(ns)   THS(ns)  THS Failing Endpoints  THS Total Endpoints  WPWS(ns)  TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
-----       -------   -------  ---------------------  -------------------  -------   -------  ---------------------  -------------------  --------  --------  ----------------------  --------------------
clk_pin       3.715     0.000                      0                  112    0.064     0.000                      0                  112     3.500     0.000                       0                    60
```

- Maximum delay path

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             3.715ns  (required time - arrival time)
  Source:                 U0/meta_harden_rst_i0/signal_dst_reg/C
                            (rising edge-triggered cell FDRE clocked by clk_pin  {rise@0.000ns fall@4.000ns period=8.000ns})
  Destination:            U0/led_ctl_i0/led_pipeline_reg_reg[0]/CE
                            (rising edge-triggered cell FDRE clocked by clk_pin  {rise@0.000ns fall@4.000ns period=8.000ns})
  Path Group:             clk_pin
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            8.000ns  (clk_pin rise@8.000ns - clk_pin rise@0.000ns)
  Data Path Delay:        3.740ns  (logic 0.642ns (17.166%)  route 3.098ns (82.834%))
  Logic Levels:           1  (LUT1=1)
  Clock Path Skew:        -0.340ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    5.450ns = ( 13.450 - 8.000 )
    Source Clock Delay      (SCD):    6.114ns
    Clock Pessimism Removal (CPR):    0.323ns
  Clock Uncertainty:      0.035ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.000ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock clk_pin rise edge)    0.000     0.000 r
    H16                                               0.000     0.000 r  clk_pin (IN)
                         net (fo=0)                   0.000     0.000    clk_pin
    H16                  IBUF (Prop_ibuf_I_O)         1.451     1.451 r  clk_pin_IBUF_inst/O
```

Report Sections

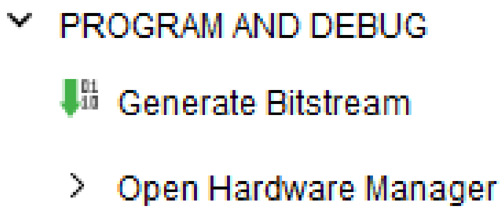
- **Delay path**

```
 Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
 -------------------------------------------------------------------  -------------------
                      (clock clk_pin rise edge)    0.000     0.000 r
 H16                                               0.000     0.000 r  clk_pin (IN)
                      net (fo=0)                   0.000     0.000    clk_pin
 H16                  IBUF (Prop_ibuf_I_O)         1.451     1.451 r  clk_pin_IBUF_inst/O
                      net (fo=1, routed)           2.522     3.973    clk_pin_IBUF
 BUFGCTRL_X0Y16       BUFG (Prop_bufg_I_O)         0.101     4.074 r  clk_pin_IBUF_BUFG_inst/O
                      net (fo=59, routed)          2.057     6.131    U0/led_ctl_i0/CLK
 SLICE_X113Y104       FDRE                                         r  U0/led_ctl_i0/led_o_reg[3]/C
 -------------------------------------------------------------------  -------------------
 SLICE_X113Y104       FDRE (Prop_fdre_C_Q)         0.456     6.587 r  U0/led_ctl_i0/led_o_reg[3]/Q
                      net (fo=1, routed)           1.526     8.112    led_pins_OBUF[3]
 M14                  OBUF (Prop_obuf_I_O)         3.581    11.693 r  led_pins_OBUF[3]_inst/O
                      net (fo=0)                   0.000    11.693    led_pins[3]
 M14                                                               r  led_pins[3] (OUT)
 -------------------------------------------------------------------  -------------------


                      (clock virtual_clock rise edge)
                                                   8.000     8.000 r
                      ideal clock network latency
                                                   0.000     8.000
                      clock pessimism              0.000     8.000
                      clock uncertainty           -0.025     7.975
                      output delay                 4.000    11.975
 -------------------------------------------------------------------
                      required time                         11.975
                      arrival time                         -11.693
 -------------------------------------------------------------------
                      slack                                  0.282
```

Bitstream Generation and Verification in Hardware

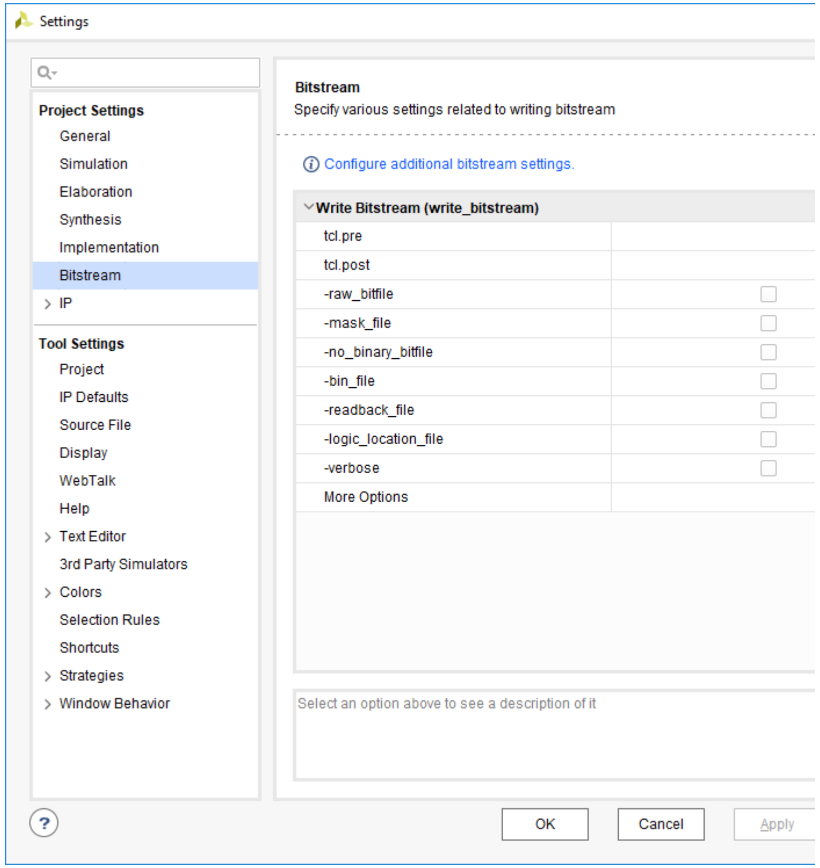
Bitstream Generation

- **Generates bitstream for the device chosen for the current project**
- **Runs on an implemented design**
- **Uses the pull model to regenerate implemented design if design is out of date**
- **Project-based flow**
  - IDE: Generate bitstream
  - Tcl: `launch_runs impl_1 -to_step write_bitstream`



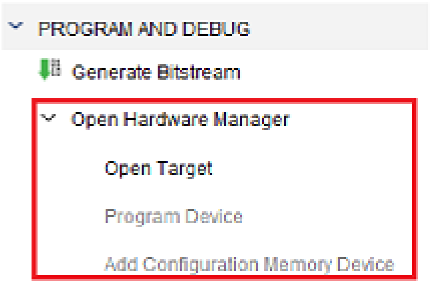
Bitstream Generation Settings

- By default, binary bitstream format is used
- `-raw_bitfile`: Causes write_bitstream to write a raw bit file (.rbt) containing the same information in ASCII format
- `-mask_file`: Generates a mask file that masks out dynamic bits in the device fabric
- `-no_binary_bitfile`: Do not write the binary bitstream file (.bit)
  - Use this command to generate the ASCII bitstream or mask file, or to generate a bitstream report, without generating the binary bitstream file
- `-bin_file`: Creates a binary file (.bin) containing only device programming data, without the header information
- `-logic_location_file`: Generates a (.ll) file that contains the location of LUTs, BRAM, flip-flops, latches, I/O block inputs and outputs



Hardware Manager

- **The steps to connect to hardware and programing the target FPGA device**
  - Open a hardware manager
    - Uses Target Communication Framework (TCF) Agent, `hw_server`
  - Open a hardware target that is managed by a hardware server running on a host computer
  - Associate the bitstream data programming file with the appropriate FPGA device
  - Program or download the programming file into the hardware device
  - Opens the hardware analyzer view for debugging



Summary

Summary

- Implementation is made up of the programs link, opt_design, power_opt, place_design, phys_opt_design, and route_design
- There are several implementation reports available to help designers better manage their FPGA designs
- Static timing paths start at clocked elements and end at clocked elements
- Static timing paths are analyzed for setup and hold violations at both fastest and slowest process corners
- Setup and hold checks include the analysis of the clock propagation paths
- report_timing_summary is used as post-implementation sign-off
- report_timing is used for interactive and detailed timing analysis after synthesis or implementation

Thank You

Disclaimer and Attribution

AMD
XILINX