



Department of Electronic & Telecommunication Engineering  
University of Moratuwa

EN4604 - DIGITAL IC DESIGN

LABORATORY EXPERIMENT 1: RTL SYNTHESIS

Supervisors:

*Dr. Thayaparan Subramaniam*

Group No :- 07

*MANIMOHAN T.*

*200377M*

*MARASINGHE M.M.H.N.B.*

*200381U*

*METHSARANI H.E.N.*

*200395P*

This report is submitted as the fulfillment of Lab - 01 of module EN4604

## **ABSTRACT**

The work performed in this lab consisted of RTL synthesis using Cadence Genus in order to understand the ASIC synthesis flow. The experiment involves synthesizing a transceiver design through Verilog RTL code. We feed source Verilog files, technology libraries, and timing constraints as inputs to Cadence Genus. Cadence Genus synthesizes the design and provides netlist and timing constraints as output. First, we created the project directories and imported the RTL design files into Cadus Genus. Then, elaboration on the design was run, followed by the application of timing constraints. Further, synthesis was performed, mapping the RTL design onto standard cells provided by the technology libraries. In this lab, we modified some of these design parameters-such as the length of the words to be transmitted and received-and then re-compiled the design in attempts to measure the resulting area and timing. We then modified the timing constraint file by changing clock frequencies and duty cycles and modifying pin loads; and finally, induce a phase shift in one of the clocks going into the transceivers. These modifications were evident in the synthesis results, revealing timing violations that necessitate a performance impact analysis on our design.

## 1 INTRODUCTION

### UART Transceivers Reference Design

The following design in **Figure 1** shows two UART Transceivers connected together. The entire design has been implemented in **Verilog HDL**, adhering to the RS232 communication standard. This system contains **two independent clock signals**, one for each transceiver. Transceiver A is clocked with **clk\_a**, while Transceiver B is clocked with **clk\_b**. Additionally, there are two reset signals: **reset\_a** is used to reset Transceiver A, and **reset\_b** is used to reset Transceiver B.

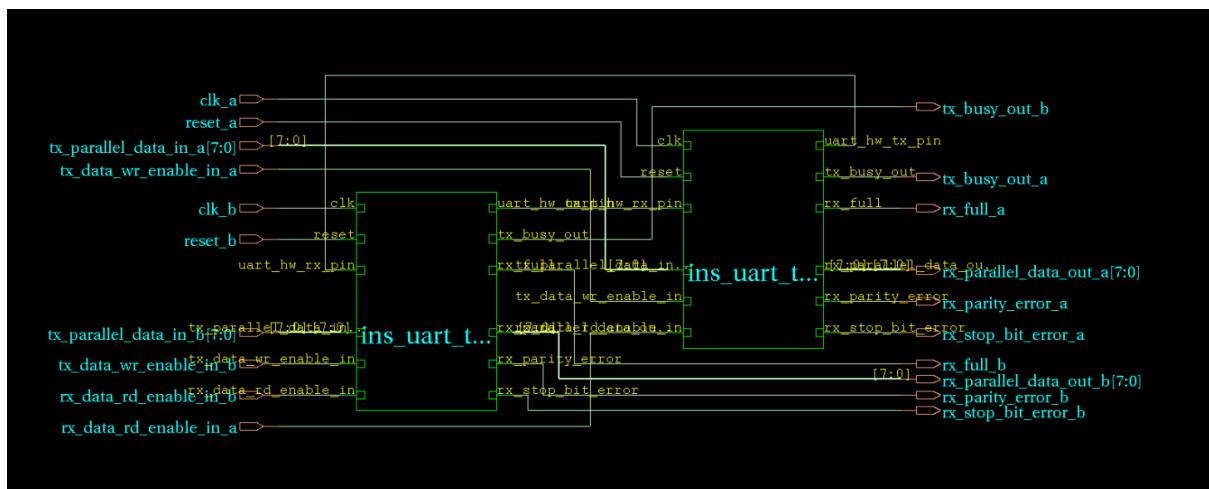


Figure 1 — `UART_Transceiver_Top.v`

Each one of the UART transceiver modules controls the data transmission and reception. The user interface on both transmitter and receiver is parallel in nature.

#### Transmitter Interface:

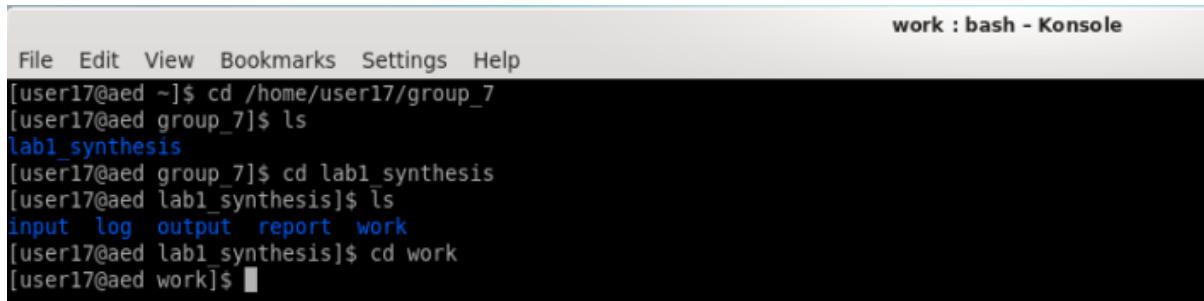
- \* `tx_data_wr_enable_in_x` – Enables the writing of data to be transmitted.
- \* `tx_parallel_data_in_x` – Inputs the parallel data for transmission.
- \* `tx_busy_out_x` – Indicates when the transmitter is busy.

#### Receiver Interface:

- \* `rx_full_x` – Indicates that the receive buffer is full.
- \* `rx_parallel_data_out_x` – Outputs the received parallel data.
- \* `rx_parity_error_x` – Flags a parity error if detected.
- \* `rx_stop_bit_error_x` – Flags a stop bit error if detected.
- \* `rx_data_rd_enable_in_x` – Enables the reading of received data.

## 2 SETTING UP AND SYNTHESIZING A DIGITAL DESIGN PROJECT USING GENUS

### Navigating to Directory

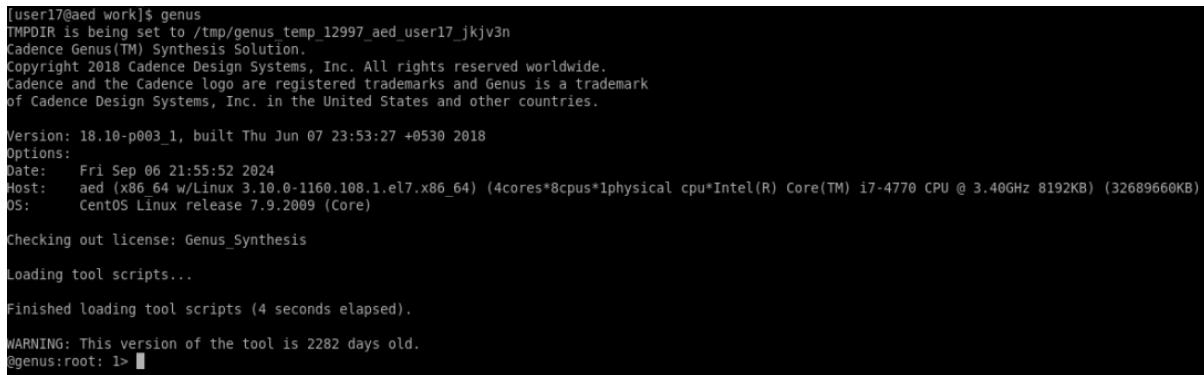


```
File Edit View Bookmarks Settings Help
work : bash - Konsole
[user17@aed ~]$ cd /home/user17/group_7
[user17@aed group_7]$ ls
lab1_synthesis
[user17@aed group_7]$ cd lab1_synthesis
[user17@aed lab1_synthesis]$ ls
input log output report work
[user17@aed lab1_synthesis]$ cd work
[user17@aed work]$
```

Figure 2 — Set Up the Project Directories

### Using Genus

#### Start Genus



```
[user17@aed work]$ genus
TMPDIR is being set to /tmp/genus_temp_12997_aed_user17_jkjv3n
Cadence Genus(TM) Synthesis Solution.
Copyright 2018 Cadence Design Systems, Inc. All rights reserved worldwide.
Cadence and the Cadence logo are registered trademarks and Genus is a trademark
of Cadence Design Systems, Inc. in the United States and other countries.

Version: 18.10-p003_1, built Thu Jun 07 23:53:27 +0530 2018
Options:
Date:   Fri Sep 06 21:55:52 2024
Host:   aed (x86_64 w/Linux 3.10.0-1160.108.1.el7.x86_64) (4cores*8cpus*1physical cpu*Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz 8192KB) (32689660KB)
OS:    CentOS Linux release 7.9.2009 (Core)

Checking out license: Genus_Synthesis
Loading tool scripts...
Finished loading tool scripts (4 seconds elapsed).

WARNING: This version of the tool is 2282 days old.
@genus:root: 1> ■
```

Figure 3 — Start the Genus

### Set the Library Search Path

This ensures that at setup, Genus can reach out to necessary technology libraries and files for proper design synthesis and analysis.

## ENEN4604: LABORATORY EXPERIMENT 1 - RTL SYNTHESIS

```
@genus:root: 1> set_db init_lib search_path [list ..../input/libs/gsclib045/lef ..../input/libs/gsclib045/timing ..../input/libs/gsclib045/qrc/qx]
Setting attribute of root '/': 'init lib search path' = ..../input/libs/gsclib045/lef ..../input/libs/gsclib045/timing ..../input/libs/gsclib045/qrc/qx
1 {..../input/libs/gsclib045/lef ..../input/libs/gsclib045/timing ..../input/libs/gsclib045/qrc/qx}
@genus:root: 2> █
```

Figure 4 — Setting up the Library Search Path

## Including Technology Libraries

```
@genus:root: 2> set_db library {slow_vdd1v0_basicCells.lib fast_vdd1v0_basicCells.lib}
Threads Configured:6

Message Summary for Library both libraries:
*****
Could not find an attribute in the library. [LBR-436]: 1148
Missing a function attribute in the output pin definition. [LBR-518]: 2
*****


Warning : Libraries have inconsistent nominal operating conditions. [LBR-38]
: The libraries are 'slow_vdd1v0' and 'fast_vdd1v0'.
: This is a common source of delay calculation confusion and is strongly discouraged.
Info : Created nominal operating condition. [LBR-412]
: Operating condition '_nominal_' was created for the PVT values (1.000000, 0.900000, 125.000000) in library 'slow_vdd1v0_basicCells.lib'.
: The nominal operating condition represents either the nominal PVT values if specified in the library source, or the default PVT values (1.0, 1.0, 1
.0).
Info : Created nominal operating condition. [LBR-412]
: Operating condition '_nominal_' was created for the PVT values (1.000000, 1.100000, 0.000000) in library 'fast_vdd1v0_basicCells.lib'.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'ANTENNA' must have an output pin.
: Add the missing output pin(s), then reload the library. Otherwise, the library cell will be marked as unusable and as timing model.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'ANTENNA' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP10' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP10' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP2' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP2' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP3' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP3' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP4' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP4' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP5' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP5' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP6' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP6' must have an output pin.
Warning : Library cell has no output pins defined. [LBR-9]
: Library cell 'DECAP7' must have an output pin.
```

Figure 5 — Including Timing Libraries

- \* Library Timing (.lib) Files: This defines the timing features for all standard cells, inclusive of cell delay, transition times, setup, and hold times. These library files include slow and fast versions to account for process variations that may affect cell performance.

```
@genus:root: 4> set_db qrc_tech_file gpdk045.tch
According to qrc_tech_file, there are total 11 routing layers [ V(5) / H(6) ]

Setting attribute of root '/': 'qrc_tech_file' = /home/user17/group_7/lab1_synthesis/work/../input/libs/gsclib045/qrc/qx/gpdk045.tch
1 /home/user17/group_7/lab1_synthesis/work/../input/libs/gsclib045/qrc/qx/gpdk045.tch
@genus:root: 5> █
```

Figure 6 — Including Tch Files

- \* TCH Files: These are binary files that characterize library elements with detailed capacitance and resistance information essential for accurate simulation and analysis of the design.

```
File Edit View Bookmarks Settings Help
@genus:root: 3> set db lef library {gsclib045_tech.lef gsclib045_macro.lef gsclib045_multibitsDFF.lef}
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_HV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_VV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_WH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_HH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_2x1_HV_E' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_2x1_HV_W' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_1x2_HV_N' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M2_M1_1x2_HV_S' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_VH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_WH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_HV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_VV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_M_NH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_M_SH' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_2x1_VH_E' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_2x1_VH_W' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_1x2_VH_N' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M3_M2_1x2_VH_S' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M4_M3_HV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
        : Via 'M4_M3_VV' has no resistance value.
Info   : Via with no resistance will have a value of '0.0' assigned. [PHYS-129]
```

Figure 7 — Including Library Exchange Format (LEF) Files

- \* LEF files (Library Exchange Format): These will define the design rules, metal layer capacitances, and other information particular to the layers of interest. These supply the details necessary in doing a correct layout and physical design implementation.

### Read the RTL design files into Genus

This command reads all the Verilog RTL design files from the targeted directory ('./input/rtl/') into the Genus.

```
@genus:root: 5> read hdl [glob ..../input/rtl/*.v]
```

Figure 8 — Read the RTL design files

## Elaborate the Design

```
@genus:root: 6> elaborate uart_top
Info    : Elaborating Design. [ELAB-1]
          : Elaborating top-level block 'uart_top' from file '../input/rtl/uart_top.v'.
Warning : Using default parameter value for module elaboration. [CDFG-818]
          : Elaborating block 'uart_top' with default parameters value.
Info    : Done Elaborating Design. [ELAB-3]
          : Done elaborating 'uart_top'.
Checking for analog nets...
Check completed for analog nets.
Checking for source RTL...
Check completed for source RTL.
UM: flow.cputime   flow.realtime   timing.setup.tns   timing.setup.wns   snapshot
UM:           70           3338                               elaborate
design:uart top
```

Figure 9 — Elaboration of the design after the RTL files are read into genus.

At this stage, Genus synthesizes the RTL design. It also executes syntax checking, builds up the design hierarchy, applies any parameters, and drives signals. We need to specify the top module of our design (uart\_top) that we want to start with to initiate the synthesis process.

## Check the Design

```
@genus:root: 7> check_design > ../log/check_design.log
Checking the design.

Done Checking the design.
```

Figure 10 — Check bugs, undriven pins, etc. in the design

This command checks for problems such as undriven pins, unloaded ports, unresolved references, and empty modules. The results will be written to a log file (check\_design.log) for review.

## Uniquify the top module.

```
@genus:root: 8> uniquify uart_top
```

Figure 11 — Uniquify uart\_top module

This marks each instantiation of the top entity (uart\_top) to be unique, and thus not to reuse any sub-blocks across instantiated entities. This will be quite relevant for proper synthesis and optimization.

## Timing Constraints Setup

```
@genus:root: 9> source ../input/constraints.tcl
Sourcing '../input/constraints.tcl' (Fri Sep 06 22:53:06 +0530 2024)...
#@ Begin verbose source input/constraints.tcl
@file(constraints.tcl) 5: create_clock -name clk_a -period 10 [get_ports clk_a] -waveform {0 5}
@file(constraints.tcl) 6: create_clock -name clk_b -period 10 [get_ports clk_b] -waveform {0 5}
@file(constraints.tcl) 8: set design_inputs [get_ports {tx_parallel_data_in* tx_data_wr_enable_in* rx_data_rd_enable_in*}]
@file(constraints.tcl) 9: set design_outputs [get_ports {tx_busy_out_* rx_full_* rx_parallel_data_out_* rx_parity_error_* rx_stop_bit_error_*}]
@file(constraints.tcl) 11: set design_inputs_a [get_ports {tx_parallel_data_in a tx_data_wr_enable_in a rx_data_rd_enable_in a}]
@file(constraints.tcl) 12: set design_outputs_a [get_ports {tx_busy_out_a rx_full_a rx_parallel_data_out_a rx_parity_error_a rx_stop_bit_error_a}]
@file(constraints.tcl) 14: set design_inputs_b [get_ports {tx_parallel_data_in b tx_data_wr_enable_in b rx_data_rd_enable_in b}]
@file(constraints.tcl) 15: set design_outputs_b [get_ports {tx_busy_out_b rx_full_b rx_parallel_data_out_b rx_parity_error_b rx_stop_bit_error_b}]
@file(constraints.tcl) 17: set_clock_uncertainty 0.5 [all_clocks]
@file(constraints.tcl) 18: set_dont_touch_network [all_clocks]
@file(constraints.tcl) 19: set_dont_touch_network [get_ports {reset_a reset_b}]
@file(constraints.tcl) 21: set_input_delay 6 -clock clk_a $design_inputs_a
@file(constraints.tcl) 22: set_input_delay 6 -clock clk_b $design_inputs_b
@file(constraints.tcl) 23: set_output_delay 6 -clock clk_a $design_outputs_a
@file(constraints.tcl) 24: set_output_delay 6 -clock clk_b $design_outputs_b
@file(constraints.tcl) 26: set_load 0.2 $design_outputs
@file(constraints.tcl) 27: set_max_fanout 20 $design_inputs
@file(constraints.tcl) 28: set_fanout_load 2 $design_outputs
@file(constraints.tcl) 30: set_false_path -from [get_clocks clk_a] -to [get_clocks clk_b]
@file(constraints.tcl) 31: set_false_path -from [get_clocks clk_b] -to [get_clocks clk_a]
#@ End verbose source input/constraints.tcl
exception:uart top/dis 2
```

Figure 12 — Create timing constraints using genus

This **source** command will load these constraints into Genus to apply them during synthesis.

```
#####
## constraints.tcl
#####

create_clock -name clk_a -period 10 [get_ports clk_a] -waveform {0 5}
create_clock -name clk_b -period 10 [get_ports clk_b] -waveform {0 5}

set design_inputs [get_ports {tx_parallel_data_in* tx_data_wr_enable_in* rx_data_rd_enable_in*}]
set design_outputs [get_ports {tx_busy_out_* rx_full_* rx_parallel_data_out_* rx_parity_error_* rx_stop_bit_error_*}]

set design_inputs_a [get_ports {tx_parallel_data_in a tx_data_wr_enable_in a rx_data_rd_enable_in a}]
set design_outputs_a [get_ports {tx_busy_out_a rx_full_a rx_parallel_data_out_a rx_parity_error_a rx_stop_bit_error_a}]

set design_inputs_b [get_ports {tx_parallel_data_in b tx_data_wr_enable_in b rx_data_rd_enable_in b}]
set design_outputs_b [get_ports {tx_busy_out_b rx_full_b rx_parallel_data_out_b rx_parity_error_b rx_stop_bit_error_b}]

set_clock_uncertainty 0.5 [all_clocks]
set_dont_touch_network [all_clocks]
set_dont_touch_network [get_ports {reset_a reset_b}]

set_input_delay 6 -clock clk_a $design_inputs_a
set_input_delay 6 -clock clk_b $design_inputs_b
set_output_delay 6 -clock clk_a $design_outputs_a
set_output_delay 6 -clock clk_b $design_outputs_b

set_load 0.2 $design_outputs
set_max_fanout 20 $design_inputs
set_fanout_load 2 $design_outputs

set_false_path -from [get_clocks clk_a] -to [get_clocks clk_b]
set_false_path -from [get_clocks clk_b] -to [get_clocks clk_a]
```

Figure 13 — constraints.tcl

The **constraints.tcl** file is used to describe the timing and design constraints of the digital IC project. This defines two clocks, ‘clk\_a’ and ‘clk\_b’, with specified periods

and waveforms; assigns input and output port groups for different data signals. Further, it applies different sets of constraints responsible for input/output delays, clock uncertainty, maximum fanout, and load limits so as to ensure that the proper timing behavior of the functional blocks is realized. It also includes "don't touch" commands to preserve some of the networks and defines false paths between the clocks to improve synthesis performance without superfluous timing checks across some clock domains.

## Synthesizing the design

```
@genus:root: 10> synthesize -to_mapped -effort m
Warning : This command will be obsolete in a next major release. [TUI-37]
      : command: 'synthesize'
      : The synthesize command is obsolete. Use the syn_gen, syn_map or syn_opt commands instead.
Current PLE settings:

Aspect ratio      : 1.00
Shrink factor    : 1.00
Scale of res/length : 1.00
Scale of cap/length : 1.00
Net derating factor : 1.00
Thermal factor   : 1.00
Via Resistance    : 6.60 ohm (from qrc_tech_file)
Site size         : 1.91 um (from lef [tech+cell])

                                         Capacitance
Layer          / Length           Data source:
Name           Direction Utilization (pF/micron)  qrc_tech_file
-----
METAL_1        H     0.00    0.000210
METAL_2        V     1.00    0.000174
METAL_3        H     1.00    0.000174
METAL_4        V     1.00    0.000174
METAL_5        H     1.00    0.000174
METAL_6        V     1.00    0.000174
METAL_7        H     1.00    0.000175
METAL_8        V     1.00    0.000176
METAL_9        H     1.00    0.001240
METAL_10       V     1.00    0.000529
METAL_11       H     1.00    0.000686

                                         Resistance
Layer          / Length           Data source:
Name           Direction Utilization (ohm/micron)  qrc_tech_file
-----
METAL_1        H     0.00    1.226667
METAL_2        V     1.00    0.755000
METAL_3        H     1.00    0.755000
METAL_4        V     1.00    0.755000
METAL_5        H     1.00    0.755000
METAL_6        V     1.00    0.755000
METAL_7        H     1.00    0.755000
METAL_8        V     1.00    0.267500
METAL_9        H     1.00    0.267500
METAL_10       V     1.00    0.097273
METAL_11       H     1.00    0.095455

                                         Area
Layer          / Length           Data source:
Name           Direction Utilization (micron)  lef_library
-----
Metal1        H     0.00    0.060000
Metal2        V     1.00    0.080000
Metal3        H     1.00    0.080000
Metal4        V     1.00    0.080000
```

Figure 14 — Synthesis of the design

Above command synthesizes a design to the RTL and mapping stage for some technology library, using medium effort: **genus: synthesize to \_mapped -effort m**. That means the synthesis process is in balance between runtime and result quality, producing reasonably optimized results without excessively long runtimes.

## Writing Netlist and Constraints

```
@genus:root: 11> write -mapped > ../output/uart_top.v
@genus:root: 12> write sdc > ..output/uart_top.sdc
Finished SDC export (command execution time mm:ss (real) = 00:00).
@genus:root: 13> 
```

Figure 15 — Netlist and Constraints

Once synthesis is complete, the next step is to generate a netlist and constraints for the place and route process. The netlist is a Verilog file where source code has been mapped into standard cells provided by the foundry, detailing how the design is implemented with specific hardware components. The SDC file - Synopsys Design Constraints - includes constraints that the place and route tools will use, including clock definitions, timing constraints, and I/O delays.

```
#####
# Created by Genus(TM) Synthesis Solution 18.10-p003_1 on Wed Sep 04 22:32:18 +0530 2024
#
set sdc_version 2.0
set_units -capacitance 1000.0fF
set_units -time 1000.0ps
# Set the current design
current_design uart_top
create_clock -name "clk a" -period 10.0 -waveform {0.0 5.0} [get_ports clk_a]
set dont_touch network [get_clocks clk_a]
create_clock -name "clk b" -period 10.0 -waveform {0.0 5.0} [get_ports clk_b]
set dont_touch network [get_clocks clk_b]
set_load -pin load 0.2 [get_ports tx_busy_out_a]
set_fanout_load 2.000 [get_ports tx_busy_out_a]
set_load -pin load 0.2 [get_ports rx_full_a]
set_fanout_load 2.000 [get_ports rx_full_a]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[7]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[7]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[6]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[6]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[5]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[5]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[4]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[4]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[3]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[3]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[2]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[2]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[1]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[1]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_a[0]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_a[0]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_b[7]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_b[7]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_b[6]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_b[6]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_b[5]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_b[5]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_b[4]]
set_fanout_load 2.000 [get_ports rx_parallel_data_out_b[4]]
set_load -pin load 0.2 [get_ports rx_parallel_data_out_b[3]]
```

Figure 16 — uart\_top.sdc

```
// Generated by Cadence Genus(TM) Synthesis Solution 18.10-p003_1
// Generated on: Sep 7 2024 22:33:42 +0530 (Sep 7 2024 17:03:42 UTC)

// Verification Directory fv/uart_top

module rx_buffer_WORD_SIZE8_NO_OF_WORDS1(clk, reset, data_serial_wr_en,
    data_serial_in, data_parallel_rd_enable, data_parallel_out,
    buffer_full);
    input clk, reset, data_serial_wr_en, data_serial_in,
        data_parallel_rd_enable;
    output [7:0] data_parallel_out;
    output buffer_full;
    wire clk, reset, data_serial_wr_en, data_serial_in,
        data_parallel_rd_enable;
    wire [7:0] data_parallel_out;
    wire buffer_full;
    wire [4:0] buffer_full_counter;
    wire [7:0] memory;
    wire n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7;
    wire n_8, n_9, n_10, n_11, n_12, n_13, n_14, n_15;
    wire n_16, n_17, n_18, n_19, n_20, n_21, n_22, n_23;
    wire n_24, n_25, n_26, n_27, n_28, n_29, n_30, n_31;
    wire n_32, n_33, n_34, n_35, n_36, n_37, n_39, n_40, n_41;
    wire n_42, n_43, n_44, n_45, n_46, n_47;
    AND2X6 g236(.A (n_45), .B (n_46), .Y (n_47));
    NOR2BX1 g237(.AN (buffer_full_counter[3]), .B
        (buffer_full_counter[2]), .Y (n_46));
    NOR2XL g238(.A (buffer_full_counter[0]), .B (buffer_full_counter[1]),
        .Y (n_45));
    CLKBUFX20 drc_bufs(.A (n_47), .Y (buffer_full));
    DFFHDX1 \buffer_full_counter_reg[2] (.CK (clk), .D (n_34), .Q
        (buffer_full_counter[2]));
    DFFHDX1 \buffer_full_counter_reg[1] (.CK (clk), .D (n_32), .Q
        (buffer_full_counter[1]));
    NOR2X1 g555(.A (reset), .B (n_31), .Y (n_34));
    DFFHDX1 \buffer_full_counter_reg[3] (.CK (clk), .D (n_33), .Q
        (buffer_full_counter[3]));
    NOR2X1 g557(.A (reset), .B (n_29), .Y (n_33));
    DFFHDX1 \buffer_full_counter_reg[0] (.CK (clk), .D (n_30), .Q
        (buffer_full_counter[0]));
    NOR2X1 g559(.A (reset), .B (n_28), .Y (n_32));
    AOI22X1 g560(.A0 (buffer_full_counter[2]), .A1 (n_26), .B0
        (buffer_full_counter[1]), .B1 (n_25), .Y (n_31));
    NOR2X1 g561(.A (reset), .B (n_27), .Y (n_30));
    AOI22X1 g562(.A0 (buffer_full_counter[1]), .A1 (n_23), .B0
        (buffer_full_counter[3]), .B1 (n_12), .Y (n_29));
    MX2X1 g563(.A (n_11), .B (n_19), .S0 (buffer_full_counter[1]), .Y
        (n_28));
    DFFHDX8 \data_parallel_out_reg[7] (.CK (clk), .D (n_14), .Q (n_44));
    DFFHDX8 \data_parallel_out_reg[0] (.CK (clk), .D (n_22), .Q (n_36));
    DFFHDX8 \data_parallel_out_reg[1] (.CK (clk), .D (n_21), .Q (n_37));
    DFFHDX8 \data_parallel_out_reg[2] (.CK (clk), .D (n_20), .Q (n_39));
    DFFHDX8 \data_parallel_out_reg[3] (.CK (clk), .D (n_18), .Q (n_40));
    DFFHDX8 \data_parallel_out_reg[4] (.CK (clk), .D (n_17), .Q (n_41));
```

Figure 17 — uart\_top.v

## Generating and Analyzing Synthesis Reports

### Area Report

genus: report\_area > ../report/area.log: Provides an estimate of the design's area usage in square millimeters ( $\text{mm}^2$ ), helping assess the design's physical size and resource utilization.

### Timing Report

genus: report\_timing -nworst 10 > ../report/timing.log: Displays the worst 10 timing paths and highlights any timing violations, crucial for ensuring the design meets performance requirements.

### Port Report

genus: report\_port \* > ../report/ports\_final.log: Lists all input and output ports of the design, offering a detailed overview of the design's interface.

## Power Report

genus: report\_power > ./report/power.log: Estimates the power consumption of the design, allowing power optimization and efficiency evaluation.

```
@genus:root: 13> report_area > ./report/area.log
@genus:root: 14> report_timing -nworst 10 > ./report/timing.log
Warning : Possible timing problems have been detected in this design. [TIM-11]
  : The design is 'uart_top'.
  : Use 'report timing -lint' for more information.
@genus:root: 15> report_port * > ./report/ports_final.log
@genus:root: 16> report_power > ./report/power.log
@genus:root: 17> 
```

Figure 18 — Generating and Analyzing Synthesis Reports

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
uart_top		772	2220.948	946.493	3167.441
ins_uart_transceiver_B	uart_transceiver_CLOCK_IN_MHZ100_TX_WORD_LENGTH8_T	386	1110.474	423.318	1533.792
ins_rx_wrapper	rx_wrapper_NO_OF_WORS_IN_BUFFER1_NO_OF_DATA_BITS8_	197	621.072	209.183	830.175
ins_rx_buffer	rx_buffer_WORD_SIZE8_NO_OF_WORDS1_	67	292.066	61.047	353.115
ins_rx_fsm	rx_fsm_NO_OF_DATA_BITS8_PARITY_ENABLED32h54525545	74	180.918	77.867	258.785
ins_sampling_tick_generator	sampling_tick_generator_BAUD115200_CLOCK_IN_MHZ100	56	148.086	62.149	210.235
ins_tx_wrapper	tx_wrapper_NO_OF_WORDS_IN_BUFFER1_NO_OF_DATA_BITS8	189	489.402	214.215	703.617
ins_tx_fsm	tx_fsm_NO_OF_DATA_BITS8_PARITY_ENABLED32h54525545_	94	210.330	103.544	313.874
ins_tx_buffer	tx_buffer_WORD_SIZE8_NO_OF_WORDS1_	53	146.718	52.247	198.965
ins_sampling_tick_generator	sampling_tick_generator_BAUD115200_CLOCK_IN_MHZ100	42	132.354	46.412	178.766
ins_uart_transceiver_A	uart_transceiver_CLOCK_IN_MHZ100_RX_WORD_LENGTH8_T	386	1110.474	423.318	1533.792
ins_rx_wrapper	rx_wrapper_NO_OF_WORS_IN_BUFFER1_NO_OF_DATA_BITS8_	197	621.072	209.183	830.175
ins_rx_buffer	rx_buffer_WORD_SIZE8_NO_OF_WORDS1_	67	292.066	61.047	353.115
ins_rx_fsm	rx_fsm_NO_OF_DATA_BITS8_PARITY_ENABLED32h54525545	74	180.918	77.867	258.785
ins_sampling_tick_generator	sampling_tick_generator_BAUD115200_CLOCK_IN_MHZ100	56	148.086	62.149	210.235
ins_tx_wrapper	tx_wrapper_NO_OF_WORDS_IN_BUFFER1_NO_OF_DATA_BITS8	189	489.402	214.215	703.617
ins_tx_fsm	tx_fsm_NO_OF_DATA_BITS8_PARITY_ENABLED32h54525545_	94	210.330	103.544	313.874
ins_tx_buffer	tx_buffer_WORD_SIZE8_NO_OF_WORDS1_	53	146.718	52.247	198.965
ins_sampling_tick_generator	sampling_tick_generator_BAUD115200_CLOCK_IN_MHZ100	42	132.354	46.412	178.766

Figure 19 — Area Report

Instance	Leakage Cells Power(nW)	Dynamic Power(nW)	Total Power(nW)
uart_top	772 60.819	172490.052	172550.870
ins_uart_transceiver_B	386 30.415	87250.682	87281.098
ins_rx_wrapper	197 19.437	60908.810	60920.247
ins_rx_buffer	67 11.235	44449.732	44460.967
ins_rx_fsm	74 5.135	11368.789	11373.924
ins_sampling_tick_generator	56 3.067	5082.288	5085.355
ins_tx_wrapper	189 10.979	26349.873	26360.851
ins_tx_fsm	94 5.124	14188.375	14193.500
ins_tx_buffer	53 3.228	7822.657	7825.885
ins_sampling_tick_generator	42 2.626	4338.840	4341.466
ins_uart_transceiver_A	386 30.403	80046.763	80077.167
ins_rx_wrapper	197 19.422	54641.240	54666.663
ins_rx_buffer	67 11.223	37609.104	37620.327
ins_rx_fsm	74 5.146	12300.830	12305.977
ins_sampling_tick_generator	56 3.053	4131.306	4134.359
ins_tx_wrapper	189 10.981	26005.523	26016.594
ins_tx_fsm	94 5.117	13676.284	13681.321
ins_tx_buffer	53 3.232	8467.156	8470.388
ins_sampling_tick_generator	42 2.632	3862.163	3864.795

Figure 20 — Power Report

# ENEN4604: LABORATORY EXPERIMENT 1 - RTL SYNTHESIS

External Delays & Exceptions						
Port	Dir	Clock	Rise Delay	Fall Delay	Ext Delay Object	Exception Object/Type
clk_a	in	clk a	0.0	n_value	create_clock_delay_domain_1_clk_a_R_0	N/A
			clk a	no_value	0.0	create_clock_delay_domain_1_clk_a_F_0
reset_a	in	N/A	N/A	N/A	N/A	N/A
tx_parallel_data_in_a[7]	in	clk a	6000.0	6000.0	in del 1	N/A
tx_parallel_data_in_a[6]	in	clk a	6000.0	6000.0	in del 2	N/A
tx_parallel_data_in_a[5]	in	clk a	6000.0	6000.0	in del 3	N/A
tx_parallel_data_in_a[4]	in	clk a	6000.0	6000.0	in del 4	N/A
tx_parallel_data_in_a[3]	in	clk a	6000.0	6000.0	in del 5	N/A
tx_parallel_data_in_a[2]	in	clk a	6000.0	6000.0	in del 6	N/A
tx_parallel_data_in_a[1]	in	clk a	6000.0	6000.0	in del 7	N/A
tx_parallel_data_in_a[0]	in	clk a	6000.0	6000.0	in del 8	N/A
tx_data_wr_enable_in_a	in	clk a	6000.0	6000.0	in del 9	N/A
rx_data_rd_enable_in_a	in	clk a	6000.0	6000.0	in del 9	N/A
clk_b	in	clk b	0.0	n_value	create_clock_delay_domain_1_clk_b_R_0	N/A
			clk b	no_value	0.0	create_clock_delay_domain_1_clk_b_F_0
reset_b	in	N/A	N/A	N/A	N/A	N/A
tx_parallel_data_in_b[7]	in	clk b	6000.0	6000.0	in del 10	N/A
tx_parallel_data_in_b[6]	in	clk b	6000.0	6000.0	in del 11	N/A
tx_parallel_data_in_b[5]	in	clk b	6000.0	6000.0	in del 12	N/A
tx_parallel_data_in_b[4]	in	clk b	6000.0	6000.0	in del 13	N/A
tx_parallel_data_in_b[3]	in	clk b	6000.0	6000.0	in del 14	N/A
tx_parallel_data_in_b[2]	in	clk b	6000.0	6000.0	in del 15	N/A
tx_parallel_data_in_b[1]	in	clk b	6000.0	6000.0	in del 16	N/A
tx_parallel_data_in_b[0]	in	clk b	6000.0	6000.0	in del 17	N/A
tx_data_wr_enable_in_b	in	clk b	6000.0	6000.0	in del 18	N/A
rx_data_rd_enable_in_b	in	clk b	6000.0	6000.0	in del 19	N/A
tx_busy_out_a	out	clk a	6000.0	6000.0	ou del	N/A
rx_full_a	out	clk b	6000.0	6000.0	ou del 32	N/A
rx_parallel_data_out_a[7]	out	clk a	6000.0	6000.0	ou del 21	N/A
rx_parallel_data_out_a[6]	out	clk a	6000.0	6000.0	ou del 22	N/A
rx_parallel_data_out_a[5]	out	clk a	6000.0	6000.0	ou del 23	N/A
rx_parallel_data_out_a[4]	out	clk a	6000.0	6000.0	ou del 24	N/A
rx_parallel_data_out_a[3]	out	clk a	6000.0	6000.0	ou del 25	N/A
rx_parallel_data_out_a[2]	out	clk a	6000.0	6000.0	ou del 26	N/A
rx_parallel_data_out_a[1]	out	clk a	6000.0	6000.0	ou del 27	N/A
rx_parallel_data_out_a[0]	out	clk a	6000.0	6000.0	ou del 28	N/A
rx_parity_error_a	out	clk a	6000.0	6000.0	ou del 29	N/A
rx_stop_bit_error_a	out	clk a	6000.0	6000.0	ou del 30	N/A
tx_busy_out_b	out	clk b	6000.0	6000.0	ou del 31	N/A
rx_full_b	out	N/A	N/A	N/A	N/A	N/A

Figure 21 — Final Port Report

Path 1: MET (2539 ps) Setup Check with Pin ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/buffer_full_counter_reg[2]/CK->D						
Group: clk_b						
Startpoint: (F) rx_data_rd_enable_in_b						
Clock: (R) clk_b						
Endpoint: (R) ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/buffer_full_counter_reg[2]/D						
Clock: (R) clk_b						
Capture						
Clock Edge:+	10000		0			
Drv Adjust:+	0		0			
Src Latency:+	0		0			
Net Latency:+	0 (I)		0 (I)			
Arrival:=	10000		0			
Setup:-	168					
Uncertainty:-	500					
Required Time=:	9332					
Launch Clock:-	0					
Input Delay:-	6000					
Data Path:-	793					
Slack=:	2539					
Exceptions/Constraints:						
input_delay	6000		in_del_19_1			
#----- Timing Point Flags Arc Edge Cell Fanout Load Trans Delay Arrival Instance						
#-----			(arrival)			
#-----			9 11.4	0	0	6000 (.,.)
#-----			- F			
rx_data_rd_enable_in_b			- B->Y R NOR2X1	3	4.4	216 6132 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/q598/Y			- B->Y F NOR2X1	2	2.9	123 181 6312 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/q593/Y			- B->Y R 0A121X1	1	2.0	144 104 6416 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/q574/Y			- B0->Y R A0122X1	1	1.9	198 266 6623 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/q560/Y			- A1->Y F NOR2X1	1	1.9	198 266 6623 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/q555/Y			- B->Y R NOR2X1	1	1.9	127 171 6793 (.,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/buffer_full_counter_reg[2]/D -			- R DFFH0X1	1	-	- 0 6793 (.,-)
Path 2: MET (2539 ps) Setup Check with Pin ins_uart_transceiver_A/ins_rx_wrapper/ins_rx_buffer/buffer_full_counter_reg[2]/CK->D						
Group: clk_a						
Startpoint: (F) rx_data_rd_enable_in_a						
Clock: (R) clk_a						
Endpoint: (R) ins_uart_transceiver_A/ins_rx_wrapper/ins_rx_buffer/buffer_full_counter_reg[2]/D						

Figure 22 — Timing Report

## Viewing Synthesized Design in GUI

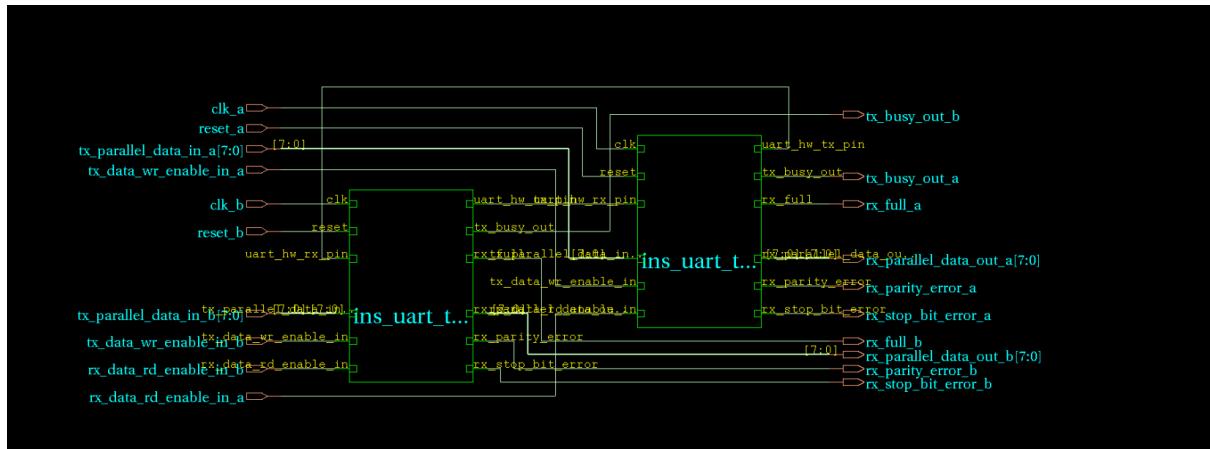


Figure 23 — Top-Level Design Schematic

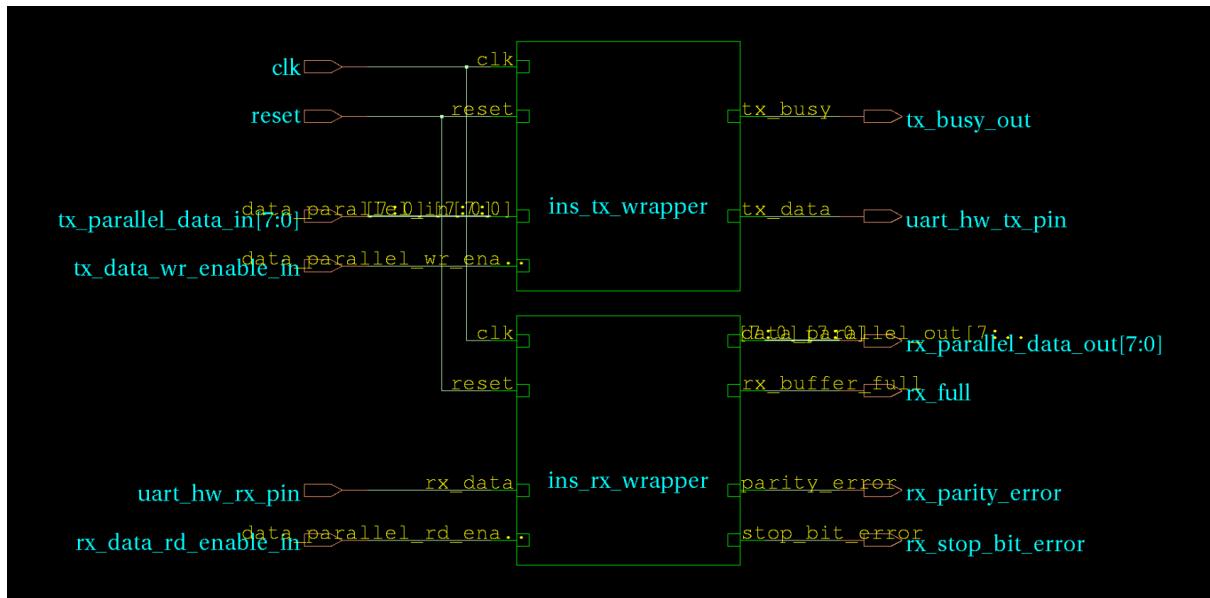


Figure 24 — ins\_uart\_transceiver<sub>B</sub>

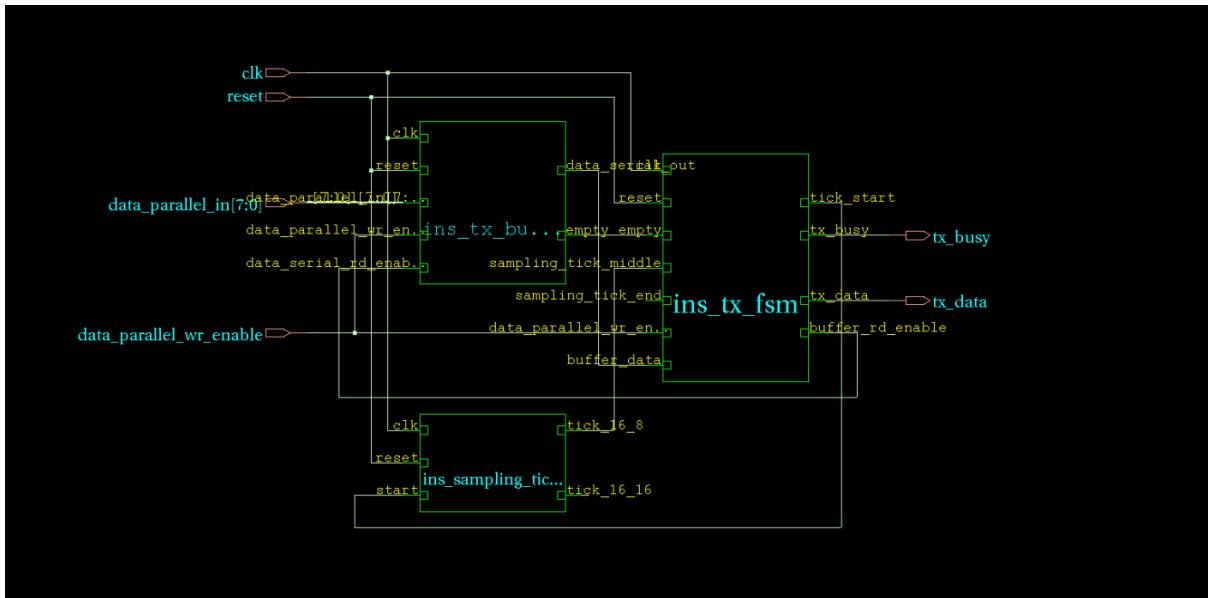


Figure 25 — `ins_tx_wrapper`

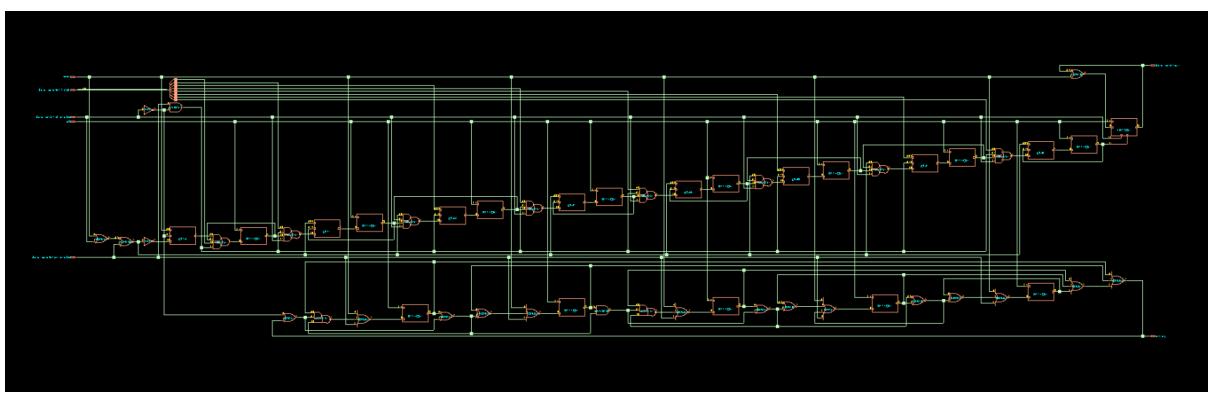


Figure 26 — `ins_tx_buffer`

### 3 EXERCISE

#### 1. Identify the system clocks, system resets and their properties.

```
#####
## constraints.tcl
#####

create_clock -name clk_a -period 10 [get_ports clk_a] -waveform {0 5}
create_clock -name clk_b -period 10 [get_ports clk_b] -waveform {0 5}

set design_inputs [get_ports {tx_parallel_data_in* tx_data_wr_enable_in* rx_data_rd_enable_in*}]
set design_outputs [get_ports {tx_busy_out_* rx_full_* rx_parallel_data_out_* rx_parity_error_* rx_stop_bit_error_*}]

set design_inputs_a [get_ports {tx_parallel_data_in_a tx_data_wr_enable_in_a rx_data_rd_enable_in_a}]
set design_outputs_a [get_ports {tx_busy_out_a rx_full_a rx_parallel_data_out_a rx_parity_error_a rx_stop_bit_error_a}]

set design_inputs_b [get_ports {tx_parallel_data_in_b tx_data_wr_enable_in_b rx_data_rd_enable_in_b}]
set design_outputs_b [get_ports {tx_busy_out_b rx_full_b rx_parallel_data_out_b rx_parity_error_b rx_stop_bit_error_b}]

set clock uncertainty 0.5 [all_clocks]
set dont_touch_network [all_clocks]
set dont touch network [get_ports {reset_a reset_b}]

set input_delay 6 -clock clk_a $design_inputs_a
set input_delay 6 -clock clk_b $design_inputs_b
set output_delay 6 -clock clk_a $design_outputs_a
set output_delay 6 -clock clk_b $design_outputs_b

set_load 0.2 $design_outputs
set_max_fanout 20 $design_inputs
set_fanout_load 2 $design_outputs

set_false_path -from [get_clocks clk_a] -to [get_clocks clk_b]
set_false_path -from [get_clocks clk_b] -to [get_clocks clk_a]
```

Figure 27 — Identification of System Clocks using constraints.tcl

	System clk_a	System clk_b
Clock Period	10 ns (0.1 GHz)	10 ns (0.1 GHz)
Clock Duty Cycle	50%	50%
Clock Uncertainty	0.5 ns	0.5 ns
Input Delay	6 ns	6 ns
Output Delay	6 ns	6 ns

Table 1 — Clock specifications and timing delays for System clk\_a and System clk\_b.

```
// Generated by Cadence Genus(TM) Synthesis Solution 18.10-p003_1
// Generated on: Sep 7 2024 22:33:42 +0530 (Sep 7 2024 17:03:42 UTC)

// Verification Directory fv/uart_top

module rx_buffer WORD_SIZE8_NO_OF_WORDS1(clk, reset, data_serial_wr_en,
    data_serial_in, data_parallel_rd_enable, data_parallel_out,
    buffer_full);
    input clk, reset, data_serial_wr_en, data_serial_in,
        data_parallel_rd_enable;
    output [7:0] data_parallel_out;
    output buffer_full;
    wire clk, reset, data_serial_wr_en, data_serial_in,
        data_parallel_rd_enable;
    wire [7:0] data_parallel_out;
    wire buffer_full;
    wire [4:0] buffer_full_counter;
    wire [7:0] memory;
    wire n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7;
    wire n_8, n_9, n_10, n_11, n_12, n_13, n_14, n_15;
    wire n_16, n_17, n_18, n_19, n_20, n_21, n_22, n_23;
    wire n_24, n_25, n_26, n_27, n_28, n_29, n_30, n_31;
    wire n_32, n_33, n_34, n_36, n_37, n_39, n_40, n_41;
    wire n_42, n_43, n_44, n_45, n_46, n_47;
    AND2X6 g236(.A (n_45), .B (n_46), .Y (n_47));
    NOR2BX1 g237(.AN (buffer_full_counter[3]), .B
        (buffer_full_counter[2]), .Y (n_46));
    NOR2XL g238(.A (buffer_full_counter[0]), .B (buffer_full_counter[1]),
        .Y (n_45));
    CLKBUFX20 drc_bufs(.A (n_47), .Y (buffer_full));
    DFFHQX1 \buffer_full_counter_reg[2] (.CK (clk), .D (n_34), .Q
        (buffer_full_counter[2]));
    DFFHQX1 \buffer_full_counter_reg[1] (.CK (clk), .D (n_32), .Q
        (buffer_full_counter[1]));
    NOR2X1 g555(.A (reset), .B (n_31), .Y (n_34));
    DFFHQX1 \buffer_full_counter_reg[3] (.CK (clk), .D (n_33), .Q
        (buffer_full_counter[3]));
    NOR2X1 g557(.A (reset), .B (n_29), .Y (n_33));
    DFFHQX1 \buffer_full_counter_reg[0] (.CK (clk), .D (n_30), .Q
        (buffer_full_counter[0]));
    NOR2X1 g559(.A (reset), .B (n_28), .Y (n_32));
    AOI22X1 g560(.A0 (buffer_full_counter[2]), .A1 (n_26), .B0
        (buffer_full_counter[1]), .B1 (n_25), .Y (n_31));
    NOR2X1 g561(.A (reset), .B (n_27), .Y (n_30));
    AOI22X1 g562(.A0 (buffer_full_counter[1]), .A1 (n_23), .B0
        (buffer_full_counter[3]), .B1 (n_12), .Y (n_29));
    MX2X1 g563(.A (n_11), .B (n_19), .S0 (buffer_full_counter[1]), .Y
        (n_28));
    DFFHQX8 \data_parallel_out_reg[7] (.CK (clk), .D (n_14), .Q (n_44));
    DFFHQX8 \data_parallel_out_reg[0] (.CK (clk), .D (n_22), .Q (n_36));
    DFFHQX8 \data_parallel_out_reg[1] (.CK (clk), .D (n_21), .Q (n_37));
    DFFHQX8 \data_parallel_out_reg[2] (.CK (clk), .D (n_20), .Q (n_39));
    DFFHQX8 \data_parallel_out_reg[3] (.CK (clk), .D (n_18), .Q (n_40));
    DFFHQX8 \data_parallel_out_reg[4] (.CK (clk), .D (n_17), .Q (n_41));
```

Figure 28 — Identification of System Resets using uart\_top.v

We identified two system resets by referring the verilog code of the uart\_top module.

```
* reset_a
* reset_b
```

There are two kinds of system resets in common use: **asynchronous active-low resets** and **synchronous active-high resets**.

**Synchronous Reset**

```
always @(posedge CLK)
begin
    if (!RSTn)
        .....
    else
        .....
end
```

**Asynchronous Reset**

```
always @(posedge CLK or negedge RSTn)
begin
    if (!RSTn)
        .....
    else
        .....
end
```

Figure 29 — Comparison between synchronous and asynchronous resets

We determined that both reset signals were synchronous and active high in the provided verilog code. Both resets occur on the rising edge of the clock because both are enclosed within an ‘always @ (posedge clk)‘ block. Synchronous resets rely on the clock for operation and are asserted when the reset signal is high.

But in general, system reset is asynchronous and active low. The asynchronous reset does not depend on the clock; hence, this is more reliable during system power-up.

**2. If there are any derived clocks, identify their properties.**

No derived clocks are present in this design, as the `create_generated_clock` command is absent from the constraints file.

**3. Observe the `area.log` and `constraints.log` on the report directory.**

The `area.log` file, shown in Figure 19, summarizes the area utilization for each module in the design. It provides details on cell count, cell area, net area, and total area, but does not include gate-level area utilization. The report aggregates the areas of cells by module, offering an abstract view of the design’s physical footprint.

**Key details:**

- \* **Total Area:** 3167.441
- \* **Breakdown:** Detailed per module with respective cell counts and area metrics.

4.Change the top-level design parameters: TX\_WORD\_LENGTH = 32 and RX\_WORD\_LENGTH = 32. Next, elaborate and synthesize the design again.

```

File Edit View Bookmarks Tools Settings Help
New Open Save Save As Close Undo Redo
uart_top.v - KWrite <2>
module uart_top #(
    parameter CLOCK_IN_MHZ = 100,
    parameter TX_WORD_LENGTH=32, // # of UART-transmit data bits ; 6,7,8
    parameter TX_NO_OF_WORDS =1, // Transmitter buffer size in words.
    > > > > > > // Actual buffer size= 'TX_WORD_LENGTH'*'TX_WORD_LENGTH' ; 1,2...
    parameter RX_WORD_LENGTH=32, // # of UART-receive data bits ; 6,7,8
    parameter RX_NO_OF_WORDS =1 // Receiver buffer size in words.
    > > > > > > // Actual buffer size= 'RX_WORD_LENGTH'*'RX_NO_OF_WORDS' ; 1,2....
)
(
    > // Interface of 'A'
    input clk_a,
    > input reset_a,
    >
    input [TX_WORD_LENGTH*TX_NO_OF_WORDS-1:0] tx_parallel_data_in_a,
    input tx_data_wr_enable_in_a,
    output tx_busy_out_a,
    >
    output rx_full_a,
    output [RX_WORD_LENGTH*RX_NO_OF_WORDS-1:0] rx_parallel_data_out_a,
    input rx_data_rd_enable_in_a,
    output rx_parity_error_a,
    output rx_stop_bit_error_a,
    >
    // Interface of 'B'
    input clk_b,
    > input reset_b,
    >
    input [TX_WORD_LENGTH*TX_NO_OF_WORDS-1:0] tx_parallel_data_in_b,
    input tx_data_wr_enable_in_b,
    output tx_busy_out_b,
    >
    output rx_full_b,
    output [RX_WORD_LENGTH*RX_NO_OF_WORDS-1:0] rx_parallel_data_out_b,
    input rx_data_rd_enable_in_b,
    output rx_parity_error_b,
    output rx_stop_bit_error_b
);
    wire w_a_tx_b_rx;
    wire w_b_tx_a_rx;
    uart_transceiver#(
        .CLOCK_IN_MHZ(CLOCK_IN_MHZ),
        .TX_WORD_LENGTH(TX_WORD_LENGTH),
        .TX_NO_OF_WORDS(TX_NO_OF_WORDS),
        .RX_WORD_LENGTH(RX_WORD_LENGTH),
        .RX_NO_OF_WORDS(RX_NO_OF_WORDS)
    ) ins_uart_transceiver_A
    (
        .clk(clk_a),
        .reset(reset_a),
        .tx_parallel_data_in(tx_parallel_data_in_a),
        .tx_data_wr_enable(tx_data_wr_enable_in_a),
        .tx_busy_out(tx_busy_out_a),
        .rx_parallel_data_out(rx_parallel_data_out_a),
        .rx_data_rd_enable(rx_data_rd_enable_in_a),
        .rx_parity_error(rx_parity_error_a),
        .rx_stop_bit_error(rx_stop_bit_error_a),
        .tx_parallel_data_in(tx_parallel_data_in_b),
        .tx_data_wr_enable(tx_data_wr_enable_in_b),
        .tx_busy_out(tx_busy_out_b),
        .rx_parallel_data_out(rx_parallel_data_out_b),
        .rx_data_rd_enable(rx_data_rd_enable_in_b),
        .rx_parity_error(rx_parity_error_b),
        .rx_stop_bit_error(rx_stop_bit_error_b)
    );
endmodule

```

Figure 30 — Modification of the bit length value to 32 in the ‘uart\_top.v’

### Comparision of Area Report

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
uart_top		272	223.948	946.493	3167.441
ins_tx_transceiver_B	uart transceiver#(CLOCK_IN_MHZ:100,TX_WORD_LENGTH:32,TX_NO_OF_WORDS:1,RX_WORD_LENGTH:32,RX_NO_OF_WORDS:1)	397	113.000	423.428	1333.428
ins_rx_wrapper	rx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	197	621.072	209.103	830.175
ins_rx_buffer	rx buffer#(WORD_SIZE:8)	67	292.068	61.047	353.115
ins_tx_wrapper	tx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	197	621.072	209.103	830.175
ins_tx_buffer	tx buffer#(WORD_SIZE:8)	56	148.086	62.149	210.235
ins_sampling_tick_generator	sampling tick generator#(BAUD115200,CLOCK_IN_MHZ:100)	189	489.402	214.215	783.617
ins_rx_fsm	rx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	44	132.354	303.103	435.447
ins_tx_fsm	tx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	53	146.718	52.247	198.965
ins_tx_wrapper_A	tx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	42	132.354	46.412	178.766
ins_rx_wrapper_A	rx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	197	621.072	209.103	830.175
ins_rx_buffer_A	rx buffer#(WORD_SIZE:8)	67	292.068	61.047	353.115
ins_tx_wrapper_A	tx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	197	621.072	209.103	830.175
ins_tx_buffer_A	tx buffer#(WORD_SIZE:8)	56	148.086	62.149	210.235
ins_sampling_tick_generator_A	sampling tick generator#(BAUD115200,CLOCK_IN_MHZ:100)	189	489.402	214.215	783.617
ins_rx_fsm_A	rx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	44	132.354	303.103	435.447
ins_tx_fsm_A	tx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:8)	53	146.718	52.247	198.965
ins_sampling_tick_generator_A	sampling tick generator#(BAUD115200,CLOCK_IN_MHZ:100)	42	132.354	46.412	178.766

Instance	Module	Cell Count	Cell Area	Net Area	Total Area
uart_top		1140	4917.136	1407.950	5425.887
ins_tx_transceiver_B	uart transceiver#(CLOCK_IN_MHZ:100,TX_WORD_LENGTH:32,TX_NO_OF_WORDS:1,RX_WORD_LENGTH:32,RX_NO_OF_WORDS:1)	313	105.000	320.000	1662.276
ins_rx_wrapper	rx wrapper#(WORD_SIZE:32)	199	1807.874	192.739	1990.613
ins_rx_buffer	rx buffer#(WORD_SIZE:32)	57	140.428	62.890	211.318
ins_tx_wrapper	tx wrapper#(WORD_SIZE:32)	54	140.428	62.890	211.318
ins_tx_buffer	tx buffer#(WORD_SIZE:32)	258	704.178	295.332	999.510
ins_tx_fsm	tx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	142	401.166	152.161	553.327
ins_rx_fsm	rx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	74	132.354	46.412	178.766
ins_tx_wrapper_A	tx wrapper#(WORD_SIZE:32)	570	132.354	46.412	178.766
ins_rx_wrapper_A	rx wrapper#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	312	1394.388	319.888	1624.276
ins_rx_buffer_A	rx buffer#(WORD_SIZE:32)	199	1807.874	192.739	1990.613
ins_tx_fsm_A	tx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	57	140.428	62.890	211.318
ins_rx_fsm_A	rx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	56	140.428	62.890	211.318
ins_tx_buffer_A	tx buffer#(WORD_SIZE:32)	142	401.166	152.161	553.327
ins_tx_fsm_A	tx fsm#(NO_OF_WORFS:1,NO_OF_DATA_BITS:32)	74	170.058	76.883	247.461
ins_sampling_tick_generator_A	sampling tick generator#(BAUD115200,CLOCK_IN_MHZ:100)	42	132.354	46.412	178.766

Figure 31 — Area Report For 8bits Word Length

Figure 32 — Area Report For 32bits Word Length

Instance	Total Area (8bits Word Length)	Total Area (32bits Word Length)
uart_top	3167.441	5425.087
ins_uart_transceiver_B	1533.792	2623.786
ins_rx_wrapper	830.175	1624.276
ins_rx_buffer	353.115	1200.613
ins_rx_fsm	258.785	211.318
ins_sampling_tick_generator	210.235	204.666
ins_tx_wrapper	703.617	999.510
ins_tx_fsm	313.874	553.327
ins_tx_buffer	198.965	247.461
ins_sampling_tick_generator	178.766	178.766
ins_uart_transceiver_A	1533.792	2623.786
ins_rx_wrapper	830.175	1624.276
ins_rx_buffer	353.115	1200.613
ins_rx_fsm	258.785	211.318
ins_sampling_tick_generator	210.235	204.666
ins_tx_wrapper	703.617	999.510
ins_tx_fsm	313.874	553.327
ins_tx_buffer	198.965	247.461
ins_sampling_tick_generator	178.766	178.766

Table 2 — The table compares the total area of each instance using 8-bit and 32-bit word lengths.

- **Increased Area:** Most instances show a significant increase in area when transitioning from 8-bit to 32-bit word lengths. For example, `uart_top`, `ins_uart_transceiver_B`, `ins_uart_transceiver_A` exhibit considerable area growth.
- **Decreased Area:** A few instances experience a reduction in area with 32-bit word lengths. Notably, `ins_rx_fsm` and `ins_sampling_tick_generator` show reduced area utilization.
- **No Change:** The area for `ins_sampling_tick_generator` remains unchanged across both word lengths.

## Comparision of Timing Report

Figure 33 — Timing Report For 8bits Word Length

## 8-bit Case

## Worst-case Path Details:

- \* Path: rx\_data\_rd\_enable\_in\_b to buffer\_full\_counter\_reg[2]/D
  - \* Timing Parameters:
    - Setup Time: 168 ps
    - Uncertainty: 500 ps
    - Required Time: 9332 ps
    - Data Path Delay: 793 ps
    - Arrival Time: 10000 ps
    - Input Delay: 6000 ps
    - Clock Edge: +10000 ps for Capture, 0 ps for Launch

## Analysis:

- \* The slack: 2539 ps.  
 \* No timing violations were detected as the slack is positive.

## 32-bit Case

## Worst-case Path Details:

- \* **Path:** rx\_data\_rd\_enable\_in\_b to buffer\_full\_counter\_reg[4]/D
  - \* **Worst-case Slack:** 2031 ps
  - \* **Timing Parameters:**
    - **Setup Time:** 168 ps
    - **Uncertainty:** 500 ps
    - **Required Time:** 9332 ps

Figure 34 — Timing Report For 32bits Word Length

- **Data Path Delay:** 1301 ps
- **Arrival Time:** 10000 ps
- **Input Delay:** 6000 ps
- **Clock Edge:** +10000 ps for Capture, 0 ps for Launch

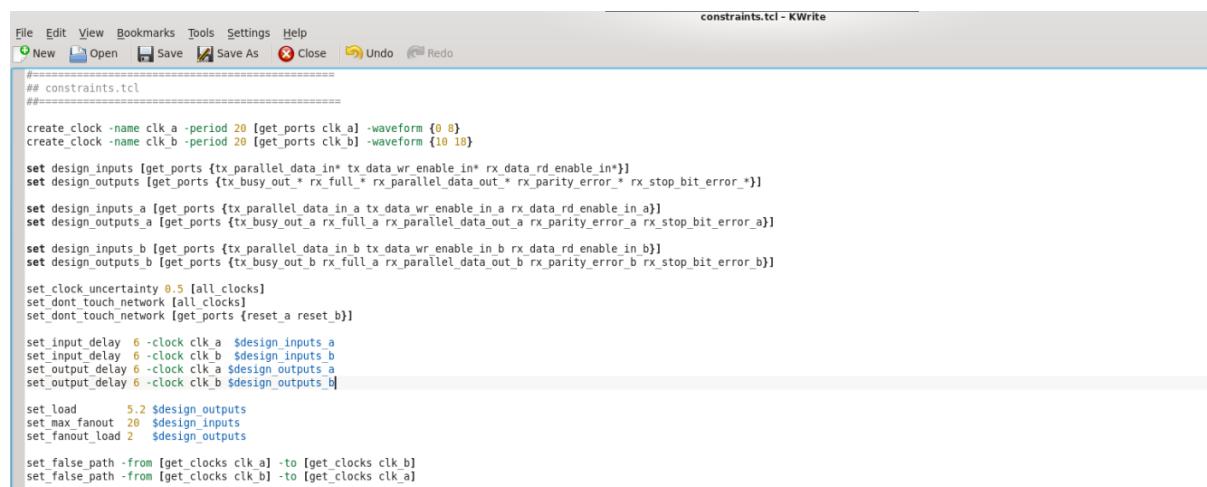
#### Analysis:

- \* The slack: 2031 ps.
- \* This indicates that even with the increase in word length from 8 bits to 32 bits, the timing requirements are still met without any violations.

#### Timing Violations

1. Given that the slack of the worst-case timing path remains positive in both the 8-bit and 32-bit cases, there are no timing violations in the design.
2. The increase in word length from 8-bits to 32-bits slightly reduced the slack but did not introduce any timing violations. This suggests that the design is robust and the timing constraints are adequately satisfied across different word lengths.

**5. Without modifying the HDL source files, do the necessary changes to constraints.tcl and synthesize the design to meet the below criteria.**



The screenshot shows a KWrite text editor window titled "constraints.tcl - KWrite". The menu bar includes File, Edit, View, Bookmarks, Tools, Settings, Help, New, Open, Save, Save As, Close, Undo, and Redo. The code in the editor is as follows:

```
File Edit View Bookmarks Tools Settings Help
constraints.tcl - KWrite

## constraints.tcl
#####
create_clock -name clk_a -period 20 [get_ports clk_a] -waveform {0 8}
create_clock -name clk_b -period 20 [get_ports clk_b] -waveform {10 18}

set design_inputs [get_ports {tx_parallel_data_in* tx_data_wr_enable_in* rx_data_rd_enable_in*}]
set design_outputs [get_ports {tx_busy_out* rx_full* rx_parallel_data_out_* rx_parity_error_* rx_stop_bit_error_*}]

set design_inputs_a [get_ports {tx_parallel_data_in_a tx_data_wr_enable_in_a rx_data_rd_enable_in_a}]
set design_outputs_a [get_ports {tx_busy_out_a rx_full_a rx_parallel_data_out_a rx_parity_error_a rx_stop_bit_error_a}]

set design_inputs_b [get_ports {tx_parallel_data_in_b tx_data_wr_enable_in_b rx_data_rd_enable_in_b}]
set design_outputs_b [get_ports {tx_busy_out_b rx_full_b rx_parallel_data_out_b rx_parity_error_b rx_stop_bit_error_b}]

set_clock_uncertainty 0.5 [all_clocks]
set_dont_touch_network [all_clocks]
set_dont_touch_network [get_ports {reset_a reset_b}]

set_input_delay 6 -clock clk_a $design_inputs_a
set_input_delay 6 -clock clk_b $design_inputs_b
set_output_delay 6 -clock clk_a $design_outputs_a
set_output_delay 6 -clock clk_b $design_outputs_b

set_load 5.2 $design_outputs
set_max_fanout 20 $design_inputs
set_fanout_load 2 $design_outputs

set_false_path -from [get_clocks clk_a] -to [get_clocks clk_b]
set_false_path -from [get_clocks clk_b] -to [get_clocks clk_a]
```

Figure 35 — Synthesis and Timing Analysis with Modified Constraints

**Change the system clocks of both A and B transceivers to operate at 50 MHz.**

To change the system clocks of transceivers A and B to 50 MHz, the clock period is set to 20 ns.

**Change the duty cycle of both clocks to 40%.**

The duty cycle of both clocks is set to 40%, meaning the clock signal is high for 8 ns and low for 12 ns for total period of 20ns.

**Introduce a 180° phase shift to clock B.**

We delayed the waveform of tranceiver B by 10ns to introduce 180° phase shift.

```
create_clock -name clk_a -period 20 [get_ports clk_a] -waveform {0 8}
create_clock -name clk_b -period 20 [get_ports clk_b] -waveform {10 18}
```

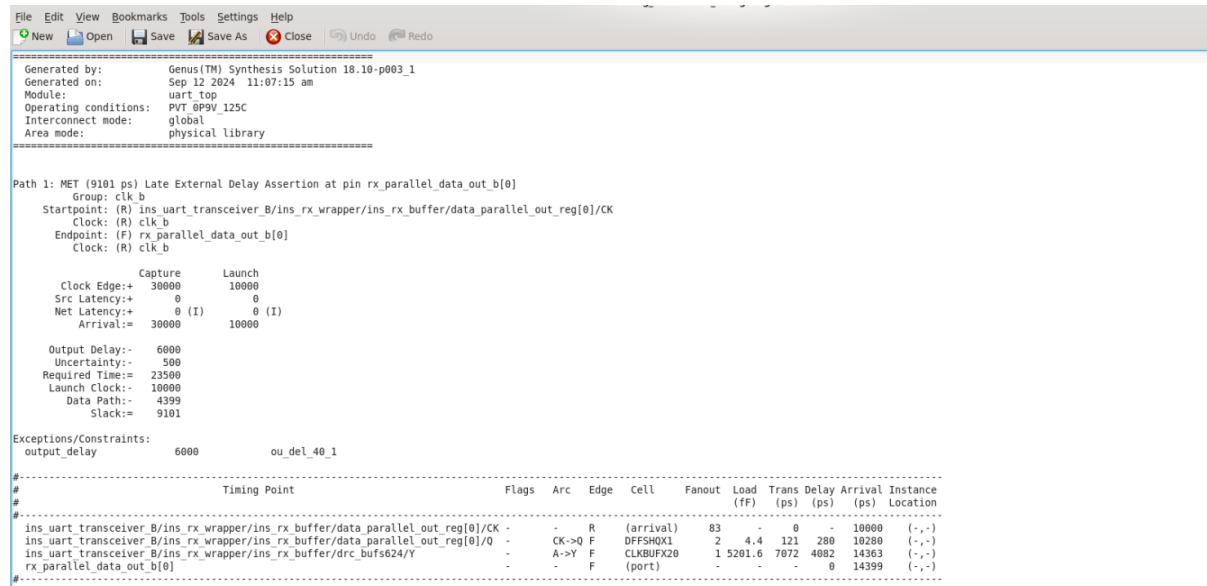
Above code results in half-period shift between clock A and clock B.

**Increase the pin load by 5 pF and check for timing violations.**

To increased pin load by 5 pF, the `set_load` command is used.

```
set_load 5.2 $design_outputs
```

## Identification of timing violations for constraints changes



A screenshot of a software interface showing a timing report. The menu bar includes File, Edit, View, Bookmarks, Tools, Settings, Help, New, Open, Save, Save As, Close, Undo, and Redo. The report header indicates it was generated by Genus(TM) Synthesis Solution 18.10-p003\_1 on Sep 12 2024 at 11:07:15 am, for uart\_top module operating at PVT 0P9V\_125C with global interconnect mode and physical library area mode. It details a single timing path (Path 1) from rx\_parallel\_data\_out\_b[0] to clk\_b, showing capture and launch times, clock edges, latency, and arrival times. It also lists exceptions/constraints like output\_delay and provides a detailed timing point table.

```
File Edit View Bookmarks Tools Settings Help
New Open Save Save As Close Undo Redo

Generated by: Genus(TM) Synthesis Solution 18.10-p003_1
Generated on: Sep 12 2024 11:07:15 am
Module: uart_top
Operating conditions: PVT 0P9V_125C
Interconnect mode: global
Area mode: physical library

=====
Path 1: MET (9101 ps) Late External Delay Assertion at pin rx_parallel_data_out_b[0]
  Group: clk_b
  Startpoint: (R) ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/data_parallel_out_reg[0]/CK
    Clock: (R) clk_b
  Endpoint: (F) rx_parallel_data_out_b[0]
    Clock: (R) clk_b

    Capture          Launch
    Clock Edge:+ 300000      100000
    Src Latency:+   0          0
    Net Latency:+   0 (I)      0 (I)
    Arrival:= 300000      100000

  Output Delay: 6000
  Uncertainty: 500
  Required Time:= 23500
  Launch Clock: 100000
  Data Path: 4399
  Slack:= 9101

Exceptions/Constraints:
  output_delay       6000        ou_del_40_1

#-----#
#           Timing Point
#-----#
#-----# Flags Arc Edge Cell Fanout Load Trans Delay Arrival Instance
#-----# (ff) (ps) (ps) (ps) Location
#-----#
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/data_parallel_out_reg[0]/CK - - R (arrival) 83 - 0 10000 (-,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/data_parallel_out_reg[0]/Q - CK->Q DFFSHDX1 2 4.4 121 280 10280 (-,-)
ins_uart_transceiver_B/ins_rx_wrapper/ins_rx_buffer/drc_bufs24/Y - A->Y CLKBUFX20 1 5201.6 7072 4682 14363 (-,-)
rx_parallel_data_out_b[0] - F (port) - - 0 14399 (-,-)
#-----#
```

Figure 36 — Timing Report with Modified Constraints

\* Slack: 9101 ps.

The design has no timing violations and the slack has increased considerably after constraint modification. It has improved due to a reduction of clock frequency. Therefore, if the design is to run at this clock frequency (50MHz), we have the opportunity to optimize it for area because the slack is large and positive.