Adding timing constraint, synthesizing report, writing timing constraint Tutorial

1. First, navigate to the MY_design folder created in Task 2.

```
baris@walle:~$ cd MY_design/
```

2. Use *ls* make sure all the files created in Task 2 are still there. Then, open DC shell.

```
baris@walle:~/MY_design$ dc_shell

                  Design Compiler Graphical
                        DC Ultra (TM)
                         DFTMAX (TM)
                     Power Compiler (TM)
                       DesignWare (R)
                       DC Expert (TM)
                     Design Vision (TM)
                      HDL Compiler (TM)
                     VHDL Compiler (TM)
                        DFT Compiler
                    Library Compiler (TM)
                     Design Compiler(R)

          Version G-2012.06 for RHEL32 -- May 30, 2012
             Copyright (c) 1988-2012 Synopsys, Inc.

This software and the associated documentation are confidential and
proprietary to Synopsys, Inc. Your use or disclosure of this software
is subject to the terms and conditions of a written license agreement
between you, or your company, and Synopsys, Inc.

Initializing...
```

3. As in the previous tutorial, set your target and link library using *target_library* and *link_library* respectively.

```
dc_shell> set target_library u/baris/MY_design/osu05_stdcells.db
u/baris/MY_design/osu05_stdcells.db
dc_shell> set link_library u/baris/MY_design/osu05_stdcells.db
u/baris/MY_design/osu05_stdcells.db
```

4. Use *read_verilog MY_DESIGN.v* to load *MY_DESIGN.v* Verilog file.

```
dc_shell> read_verilog MY_DESIGN.sv
Loading db file '/pkgs/synopsys/G-2012.06/libraries/syn/gtech.db'
Loading db file '/pkgs/synopsys/G-2012.06/libraries/syn/standard.sldb'
  Loading link library 'gtech'
Loading verilog file '/u/baris/MY_design/MY_DESIGN.sv'
Detecting input file type automatically (-rtl or -netlist).
Running DC verilog reader
Reading with Presto HDL Compiler (equivalent to -rtl option).
Running PRESTO HDLC
```

5. Next, type *current_design MY_DESIGN* into the DC shell. This makes *MY_DESIGN* the top design module.

```
dc_shell> current_design MY_DESIGN
Current design is 'MY_DESIGN'.
{MY_DESIGN}
```

6. Then, type *link* into DC shell. The related linking message should be displayed as following:

```
Current design is 'MY_DESIGN'.

  Linking design 'MY_DESIGN'
  Using the following designs and libraries:
  --------------------------------------------------------------------------
  osu05_stdcells (library)    /u/pengao/Downloads/lib/osu05_stdcells.db

Current design is 'MY_DESIGN'.
```

7. Afterwards, type *check_design* into DC shell. You should see a summary like the one provided below. This command will check design violation in Design Compiler.

```
****************************************
check_design summary:
Version:     G-2012.06
Date:        Wed Aug 28 13:46:39 2019
****************************************

                Name                                                Total
--------------------------------------------------------------------------------
Cells                                                                2
    Cells do not drive (LINT-1)                                      2
--------------------------------------------------------------------------------

Warning: In design 'ARITH', cell 'B_2' does not drive any nets. (LINT-1)
Warning: In design 'ARITH', cell 'B_3' does not drive any nets. (LINT-1)
Information: Updating design information... (UID-85)
```

8. Follow the lines of code provided below to determine your timing constraints.

```
dc_shell> create_clock -period 3.0 [get_ports clk]
1
dc_shell> set_clock_latency -source  -max 0.7 [get_clocks clk]
1
dc_shell> set_clock_latency -max 0.3 [get_clocks clk]
1
dc_shell> set_clock_uncertainty -setup 0.15 [get_clocks clk]
1
dc_shell> set_clock_transition 0.12 [get_clocks clk]
1
dc_shell> set_input_delay -max  0.45 -clock clk [get_ports data*]
1
dc_shell> set_output_delay -max  0.5 -clock clk [get_ports out1]
1
dc_shell> set_output_delay -max 2.04  -clock clk [get_ports out2]
1
dc_shell> set_output_delay -max  0.4 -clock clk [get_ports out3]
1
dc_shell> set_input_delay -max 0.3 -clock clk [get_ports Cin*]
1
dc_shell> set_output_delay -max 0.1 -clock clk [get_ports Cout]
1
```

9. Use *report_clock* to check the setup of clock signal in your design. You should get all information of the clock signal like period, duty-cycle.

```
****************************************
Report : clocks
Design : MY_DESIGN
Version: G-2012.06
Date   : Wed Aug 28 13:46:39 2019
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
------------------------------------------------------------------------------
clk              3.00   {0 1.5}                       {clk}
------------------------------------------------------------------------------
```

10. Use *report_timing* to check the delay of any path in this design. You can set start point and end point to get the max/min delay of this path.

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : MY_DESIGN
Version: G-2012.06
Date   : Wed Aug 28 13:46:39 2019
****************************************

Operating Conditions: typical   Library: osu05_stdcells
Wire Load Model Mode: top

  Startpoint: R3_reg[0] (rising edge-triggered flip-flop clocked by clk)
  Endpoint: out2[0] (output port clocked by clk)
  Path Group: clk
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              1.00       1.00
  R3_reg[0]/clocked_on (**SEQGEN**)        0.00       1.00 r
  R3_reg[0]/Q (**SEQGEN**)                 0.00       1.00 r
  C26/Z (GTECH_AND2)                       0.00       1.00 r
  out2[0] (out)                            0.00       1.00 r
  data arrival time                                   1.00

  clock clk (rise edge)                    3.00       3.00
  clock network delay (ideal)              1.00       4.00
  clock uncertainty                       -0.15       3.85
  output external delay                   -2.04       1.81
  data required time                                  1.81
  -----------------------------------------------------------
  data required time                                  1.81
  data arrival time                                  -1.00
  -----------------------------------------------------------
  slack (MET)                                         0.81
```

11. Using *write_script -out myd_design.wscr* to export all your timing constraints to the file" *myd_design.wscr*". Using this file to comparing with your specification document, make sure you didn't miss any constraint.
12. Using *write -format ddc -h -out unmapped_design.ddc* to export all your timing constraints to the file" *unmapped_design.ddc*". This file can be used by other EDA tools for future design check.