EE4415 Integrated Digital Design Project Report

Name: Phang Swee King
Matric Number: U066584J

April 10, 2010

Contents

1 Lab Unit 1

Objectives

This exercise describes the 6 basic steps involved in the synthesis process and how to navigate through a designs hierarchy using Design Vision, the graphical interface to Design Compiler.

Discussions

Question 1

To increase the counter width from three bits to six bits, the *vhdl* file can be modified. `counter.v` is editted and the following line of code was modified.

```
output [5:0] Counter_Out;
reg  [5:0] Counter_Out;
```

Question 2

To increase the clock frequency to 1 GHz, the config file, `counter6.con` can be modified. The following line of code is used to replaced the original code on clock initialization.

```
create_clock -period 1 -name my_clock [get_ports Clock_In]
```

Note that it is refering to create a clock with period of 1 nanoseconds.

Question 3

To switch to a different silicon vendor, a new library file should be given by the new vendor. Then, the link to the target library file can be changed in `.synopsys_dc.setup` file. Or else, change the target library variable by typing the command set target_library <another library name>.db and change the design references by typing set link_library "* <another library name>.db"

2 Lab Unit 2

Objectives

1. Use the basic features of Design Vision
2. Use the Designs, Symbol and Schematic Views of Design Vision, and select menu, and mouse functions
3. Take a design through the basic synthesis steps

Discussions

Task 2

3. (a) Link Library is *** core_slow.db**
   (b) Target Library is **core_slow.db**
   (c) Symbol Library is **core.sdb**

Task 8

2. Max Delay: Largest Violation (Slack) is **-0.20**
   Max Area: Actual Area is **2501.35**

Question 1

To select multiple objects in Design Vision, hold the *Ctrl* key and left click the objects with the mouse button.

Question 2

The following functions are available using the right mouse button:

1. Zoom Fit All
2. Zoom In Tool
3. Zoom Out Tool
4. Pan Tool
5. Edit Attributes
6. Timing Paths Report
7. Properties

Question 3

The basic synthesis flow is as follows:

1. G. Set up library variables
2. B. Read in the unmapped design
3. D. Apply a constraint script file
4. A. Compile
5. C. Generate a constraint report
6. F. Determine if constraints are met
7. E. Save the mapped design

Question 4

Optimization and mapping of a design with Design Vision can be done by typing "**compile**" at the command window.

Question 5

By choosing **File** and **Save As**, it saves a design in Design Vision.

Question 6

Using synthesis in a design flow makes modifying and debugging much easier.

Question 7

`design_vision-xg` will invoke the GUI of the Design Compiler, where it supports mouse operation and some other visible objects. `dc_shell-xg-t` will only allow us to interface with Design Compiler using command line. It only activates the Design Compiler where the operations need to be done by using keyboard input at the prompt.

Question 8

`.synopsys_dc.setup` file is created to setup the compiler. It provides libraries and defines aliases on the components we'll use in our project.

Question 9

The following codes can be typed at the command window to check if the library variables are set up correctly:

```
printvar target_library
printvar link_library
```

Alternatively, one can invoke from the **File** and **Setup** from the menu bar.

Question 10

To read VHDL or Verilog code into Design Vision, choose **File** and **read**, then choose the corresponding file.

Question 11

The two optimization goals we can set on a design is the *smallest area* and *lowest time slack*.

Question 12

The `target_library` variable points to the desired technology library.

3 Lab Unit 3

Objectives

Improve a designs QoR (Quality of Results = Timing and/or Area) by repartitioning a design using the group and ungroup commands.

Discussions

Task 1

3. (a) Max Delay: Largest Violation (Slack) = **-0.197269**
   (b) Total Cell Area (Max Area): Actual Area = **2501.350010**

5. It violated the guideline of "avoid separating combinational logic across hierarchical boundaries" and "place hierarchy boundaries at register outputs". It can be improved by join block "DECODE" and "CNT" to improve the partitioning of this design.

Task 2

1.

Task	Compiled Design	Timing Slack	Actual Area
Task 1	Initial partitioning	-0.197269	2501.350010
Task 3	After partitioning	-0.00446081	2576.690010

Task 3

4. Max Delay: Largest Violation (Slack) = **-0.00446081**
   Max Area: Actual Area = **2576.690010**

5. Does the critical path cross any purely combinational block?
   ans: **Yes**

7. Did partitioning

   (a) Improve timing? By how much?
   Yes. Improved by 0.1928

   (b) Improve area?
   No. Worsen by 75.34

Question 1

It is important to partition a design correctly in the source code such that the design is well optimized and to speed up the synthesis process. It can also simplify constraints and scripts that need to be applied.

Question 2

The reason for not ungrouping the entire hierarchy is that it will be too messy. It will become not systematic and debugging will be tough. Furthermore, we may reuse the block in other part of the design if we partition them correctly. The most critical problem for a "flattened" design is that it will take too long time for the compilation.

Question 3

The 3 synthesis benefits we've gain from good partitioning are:

1. Time saving to detect error easier
2. Optimization can be done easily
3. Compile run time will be much faster

Question 4

To implement partitioning in the RTL code, we create the top-level hierarchy by grouping circuits with similar functionalities and clocks together by using the **group** command. On the other hand, we remove unnecessary sub-hierarchy and "glue logic" by using the **ungroup** command.

Question 5

1. Create blocks with reasonable sizes.
2. Separate core logic, pads, clocks, asynchronous logic and JTAG.

Question 6

By placing hierarchy boundaries at register outputs wil help simplify setting constraints on a design.

4 Lab Unit 4

Objectives

Write a simple DC-Tcl script file to compile a design.

Discussions

The script file `runit.tcl` is written as follows:

```
# Script file in lab 4

# Read the unmapped design netlist
read_ddc unmapped/PRGRM_CNT_TOP.ddc

# Set the current_design
current_design PRGRM_CNT_TOP
link

# Constrain design
source scripts/example.tcl

# Default compile
compile

# Generate and save constraint report
redirect -tee reports/PRGRM_CNT_TOP.rpt {report_constraint -all_violators}

# Save mapped design
write -hierarchy -format ddc -output mapped/PRGRM_CNT_TOP.ddc

# Quit DC
quit
```

5 Lab Unit 5

Objectives

Apply timing constraints to a design.

Discussions

Question 1

Input Delay = 1.25 ns
Output Delay = 3.25 ns

Question 2

`core_slow.db`, `gtech.db` and `standard.sldb` are the libraries in DC memory.

Question 3

The technbology library name for the target library is `ssc_core_slow`.

Question 4

The Time Unit is 1 ns.

Question 5

The Capacitive Load Unit is 1.000000 pf.

Question 6

The design is reset before applying design constraints to clear the previous unwanted constraint such that we could start a new design.

Question 7

It is important to check the library time units before setting constraints because we must know the time unit in order to set the time correctly.

Question 8

The `dc_shell-xg-t` command for setting a max area goal of 500 is:

```
set_max_area 500
```

Question 9

Timing has a higher priority compare to area goals.

Question 10

The name of the attribute is **max_area**.

Question 11

To check what area goal has been placed on a design, **report_attribute** is typed in command window and **max_area** attribute can be found from the report.

6 Lab Unit 6

Objectives

1. Apply operating conditions, a wire load model, and port environment attributes on a design
2. Save the attributes and constraints set on the `current_design` using the `write_script` command

Discussions

Question 1

The worst case operating condition would be under temperature of 125°C and voltage of 1.62 V.

Question 2

The following operating conditions are available:

1. slow_125_1.62
2. slow_125_1.62_WCT

Question 3

The 5KGATES wire load model will be used.

Question 4

`report_attribute` and `get_attribute`

Question 5

If the port environment is not model accurately, it is likely that even if the simulation passed, the designed circuit might fail the timing requirements in the real applications.

Question 6

The defailt operating condition is `slow_125_1.62` with temperature of 125°C and voltage of 1.62 V.

Question 7

There are 7 wire load models.

Question 8

DC would pick 40KGATES wire load model.

Question 9

0.000271 kΩ.

Question 10

CLK, D, E, Q.

Question 11

0.021000 pF.

Question 12

Yes, timing constraint is met.

Question 13

Different load have different capacitance. Loads with higher capacitance take more time to be charged up or pulled down. This would affect the timing of the circuit.

Question 14

1. per-unit-length resistance
2. per-unit-length capacitance
3. extrapolation slope

Question 15

1. set_driving_cell
2. set_max_capacitance
3. set_load
4. set_input_delay
5. set_max_area

7 Lab Unit 7

Objectives

1. Apply design rules and hold time constraints
2. Fix design rule violations
3. Fix hold time violations

Discussions

Question 1

```
set ALL_INS_EX_CLK [remove_from_collection \
                        [all_inputs] [get_ports CLK]]

set_max_transition 0.25 $ALL_INS_EX_CLK

set_input_delay -min 0.2 -clock my_clk $ALL_INS_EX_CLK

set_output_delay -min [expr 0.2-0.5] -clock my_clk [all_outputs]
```

Question 2

The data at the input of a register might not stay long enough for the register to correctly record down the value during rising edge.

8 Lab Unit 8

Objectives

Interpret various timing reports generated by `report_timing`.

Discussions

Question 1

By typing `check_timing` command, there are **no** unconstrainted timing paths.

Question 2

By typing `report_path_group` command, there are two path groups:

1. **default**
2. my_clk

Question 3

This is a setup time report.

Question 4

The start point is `Crnt_Instrn[20]`. It is an input port of internal register.

Question 5

The end point is `I_PRGRM_CNT/PCint_reg[0]`.

Question 6

It is under operating condition `slow_125_1.62` standard condition with temperature of 125°C and voltage of 1.62 V. The wire load model is 5KGATES.

Question 7

This timing path met its constrants since the slack is +0.03.

Question 8

The clock period for `Clk` is 4 ns

Question 9

For this case, input external delay is 1 ns. This number comes from the delay between the `Crnt_Instrn` register's clock input and its output.

Question 10

Yes. It can be observed that there are two partition, `I_PRGRM_DECODE` and `I_PRGRM_CNT` both having combinational logics. It means that the combinational logics are broken by the partitioning.

Question 11

The setup time required is given in `library setup time` which is 0.37 ns.

Question 12

It represents the clock skew. In other words, it means the clock edge may arrive earlier or later by 0.25 ns, where in the design, worst case is adopted.

Question 13

This timing report shows all timing with accuracy of 6 significant figure.

Question 14

They are the wire load delays. However, the wires have almost zero resistance and zero capacitance, thus the delays are zero.

Question 15

The **Fanout** column represent the sum of `fanout_load` on that particular port. If the `max_fanout` value is set, the value in this **Fanout** column must be smaller than the `max_fanout` value, or else violation will occur.

Question 16

This is a hold time report.

Question 17

The start point is `I_PRGRM_FSM/Current_State_reg[1]`.

Question 18

The end point is `Current_State[1]`.

Question 19

This timing path violate its constraints , the slack is –0.09.

Question 20

It is under operating condition `slow_125_1.62` standard condition with temperature of 125°C and voltage of 1.62 V. The wire load model is 5KGATES.

Question 21

This is not an appropriate operating condition. For hold time calculations, the minimum timing is concerned and the worst case happens when everything is the fastest. Thus, this condition is not appropriate for setting up the fastest condition.

Question 22

0.3 ns.

Question 23

0.46 ns.

Question 24

If only the clock uncertainty is zero it is enough. However for this case, the clock uncertainty is 0.25 ns. It means that the hold time requirement is $0.25 + 0.30 = 0.55$ ns. Therefore it needs another 0.09 ns to safely meet the requirement.