Computer-Aided VLSI System Design
Lab3: Synthesis Lab: Design Compiler

TA: 邱仁皓 r12943008@ntu.edu.tw

Introduction

In this lab, you will learn:

1. Basic concepts about synthesis
2. How to use Synopsys Design Compiler (in text mode)

Data Preparation

1. Upload your files (Lab3.tar) to your work directory.
2. Decompress Lab3.tar with the following command:

```
tar -xvf Lab3.tar
```

3. Lab3 files are shown as below:

Files/Folder	Description
`Lab3_test_alu.v`	Testbench for the design (ALU)
`Lab3_alu.v`	RTL code of the design (ALU)
`Lab3_alu_run_syn.f`	File list for gate-level simulation
`syn.tcl`	Command file for Design Compiler
`.synopsys_dc.setup`	Setup file for Design Compiler

4. Enter Lab3 directory:

```
cd Lab3
```

Environmental Setup

1. Source the following cshrc files to run Design Compiler and VCS.

```
source /usr/cad/synopsys/CIC/synthesis.cshrc
source /usr/cad/synopsys/cshrc
```

Synopsys Design Compiler

1. Check the search path and libraries are set as the following:

```
set company "CIC"
set designer "Student"
set search_path
{/home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/lib \
/home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db \
$search_path}
set link_library "typical.db slow.db fast.db dw_foundation.sldb"
set target_library "typical.db slow.db fast.db"
set symbol_library "generic.sdb"
set synthetic_library "dw_foundation.sldb"
```

2. The function of Lab3_alu.v is from Lab2. We will use this example and practice Synopsys synthesis tool step by step.

3. Check the RTL simulation.

```
vcs -full64 -R Lab3_test_alu.v Lab3_alu.v
```

4. Build your working directory and start up Design Compiler (in text mode).

```
dc_shell
```

3. Load file with the following command:

```
read_file -format verilog {./Lab3_alu.v}
```

```
dc_shell> read_file -format verilog {./Lab3_alu.v}
```

4. Check the log information. If there are any errors or warning messages, you have to fix it. After that, checking all the registers are of filp-flop type. You have to modify your Verilog code if there is a latch in your circuit.

```
===============================================================================
|    Register Name    |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|       reg_B_reg     | Flip-flop |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
|      reg_ins_reg    | Flip-flop |   4   |  Y  | N  | Y  | N  | N  | N  | N  |
|      alu_out_reg    | Flip-flop |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
|       reg_A_reg     | Flip-flop |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
===============================================================================
Presto compilation completed successfully.
```

5. Write out the current design and check if each macro is mapped as you expect.

```
write –format verilog -hierarchy -output ALU_GTECH.v
```

ALU_GTECH.v

```
        reg_A[1]), .synch_clear(1'b0), .synch_preset(1'b0), .synch_toggle(1'b0), .synch_enable(1'b1) );
\**SEQGEN**  \reg_A_reg[0]  ( .clear(N8), .preset(1'b0), .next_state(
        inputA[0]), .clocked_on(clk), .data_in(1'b0), .enable(1'b0), .Q(
        reg_A[0]), .synch_clear(1'b0), .synch_preset(1'b0), .synch_toggle(1'b0), .synch_enable(1'b1) );
GTECH_AND2 C88 ( .A(N19), .B(N20), .Z(N22) );
GTECH_AND2 C89 ( .A(N22), .B(N21), .Z(N23) );
GTECH_OR2 C91 ( .A(reg_ins[2]), .B(reg_ins[1]), .Z(N24) );
GTECH_OR2 C92 ( .A(N24), .B(N21), .Z(N25) );
GTECH_OR2 C95 ( .A(reg_ins[2]), .B(N20), .Z(N27) );
GTECH_OR2 C96 ( .A(N27), .B(reg_ins[0]), .Z(N28) );
GTECH_OR2 C100 ( .A(reg_ins[2]), .B(N20), .Z(N30) );
GTECH_OR2 C101 ( .A(N30), .B(N21), .Z(N31) );
GTECH_OR2 C104 ( .A(N19), .B(reg_ins[1]), .Z(N33) );
GTECH_OR2 C105 ( .A(N33), .B(reg_ins[0]), .Z(N34) );
GTECH_AND2 C107 ( .A(reg_ins[2]), .B(reg_ins[0]), .Z(N36) );
GTECH_AND2 C108 ( .A(reg_ins[2]), .B(reg_ins[1]), .Z(N37) );
ADD_UNS_OP add_42 ( .A(reg_A), .B(reg_B), .Z({N46, N45, N44, N43, N42, N41,
        N40, N39}) );
SUB_UNS_OP sub_43 ( .A(reg_A), .B(reg_B), .Z({N54, N53, N52, N51, N50, N49,
        N48, N47}) );
```

How many "GTECH_OR2" are there after HDL translation? _____

6. Specify the clock as period 10ns. (100 MHz). We also set "**don't touch network**" and "**fixhold**" attributes.

```
create_clock -name "clk" -period 10 -waveform {"0" "5"} {"clk"}
set_dont_touch_network [find clock clk]
set_fix_hold clk
```

7. Type in the following commands to change the wire load model:

```
set_operating_conditions "typical" -library "typical"
set_wire_load_model -name "ForQA" -library "typical"
set_wire_load_mode "segmented"
```

8. Set operating environment, including input delay and output delay attributes.

```
set_input_delay -clock clk 2.5 inputA[*]
set_input_delay -clock clk 3.8 inputB[*]
set_input_delay -clock clk 4.5 instruction[*]
set_input_delay -clock clk 5.2 reset
set_output_delay -clock clk 8 alu_out[*]
```

9. Set design constraints, including max area, max fanout and max transition.

```
set_boundary_optimization "*"
set_fix_multiple_port_nets –all -buffer_constant
set_max_area 0
set_max_fanout 8 ALU
set_max_transition 1 ALU
```

10. Checks the current design for consistency.

```
check_design
```

```
****************************************
check_design summary:
Version:     U-2022.12
Date:        Sun Oct  6 15:53:58 2024
****************************************

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

Warning: In design 'ALU', cell 'C200' does not drive any nets. (LINT-1)
Warning: In design 'ALU', cell 'C201' does not drive any nets. (LINT-1)
1
```

10. Perform optimization for ALU

```
compile -map_effort medium
```

The mapping details will be displayed on the console.

```
  Beginning Pass 1 Mapping
  ------------------------
  Processing 'ALU'

  Updating timing information
Information: Updating design information... (UID-85)

  Beginning Implementation Selection
  ----------------------------------
  Processing 'ALU_DW01_sub_0'
  Processing 'ALU_DW01_add_0'

  Beginning Mapping Optimizations  (Medium effort)
  --------------------------------
Loading db file '/home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/slow.db'
Loading db file '/home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/fast.db'
```

11. Check if the circuit meets the set conditions. We can report timing with the following command, and generate **ALU.timing** file to record the timing information of the optimized design.

```
report_timing -path full -delay max -max_paths 1 -nworst 1 > ALU.timing
```

Check whether the slack is positive (meets timing constraints) or negative.

```
Operating Conditions: typical   Library: typical
Wire Load Model Mode: segmented

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

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  ALU                ForQA                 typical

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  alu_out_reg[0]/CK (DFFRX1)               0.00       0.00 r
  alu_out_reg[0]/Q (DFFRX1)                0.28       0.28 f
  alu_out[0] (out)                         0.00       0.28 f
  data arrival time                                   0.28

  clock clk (rise edge)                   10.00      10.00
  clock network delay (ideal)              0.00      10.00
  output external delay                   -8.00       2.00
  data required time                                  2.00
  -----------------------------------------------------------
  data required time                                  2.00
  data arrival time                                  -0.28
  -----------------------------------------------------------
  slack (MET)                                         1.72
                                   Positive Slack!
```

12. We can report power with the following command, and generate the **ALU.power** file to record the power consumption of the optimized design.

```
report_power > ALU.power
```

```
Library(s) Used:

    typical (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/typical.db)


Operating Conditions: typical   Library: typical
Wire Load Model Mode: segmented

Design        Wire Load Model            Library
------------------------------------------------
ALU                     ForQA            typical


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


Attributes
----------
i - Including register clock pin internal power


  Cell Internal Power  =  81.6216 uW   (94%)
  Net Switching Power  =   5.3812 uW    (6%)
                         ---------
Total Dynamic Power    =  87.0028 uW  (100%)

Cell Leakage Power     = 441.1685 nW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000           0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000           0.0000            0.0000            0.0000  (   0.00%)
clock_network    6.8557e-02         0.0000            0.0000        6.8557e-02  (  78.40%)  i
register         5.6897e-03      2.2111e-03        2.4572e+05       8.1465e-03  (   9.32%)
sequential         0.0000           0.0000            0.0000            0.0000  (   0.00%)
combinational    7.3746e-03      3.1701e-03        1.9545e+05       1.0740e-02  (  12.28%)
--------------------------------------------------------------------------------------------------
Total            8.1622e-02 mW   5.3812e-03 mW     4.4117e+05 pW    8.7444e-02 mW
```

13. We can also report area with the following command, and generate the **ALU.area** file to record the area of the optimized design.

```
report_area -nosplit > ALU.area
```

```
Library(s) Used:

    typical (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/typical.db)

Number of ports:                           82
Number of nets:                           210
Number of cells:                          111
Number of combinational cells:             79
Number of sequential cells:                28
Number of macros/black boxes:               0
Number of buf/inv:                         14
Number of references:                      14

Combinational area:                1003.163413
Buf/Inv area:                        61.106399
Noncombinational area:              903.016769
Macro/Black Box area:                 0.000000
Net Interconnect area:               18.000000

Total cell area:                   1906.180182
Total area:                        1924.180182
```

14. Synthesis ends if the results meet your requirements. Next, we must export the design to a file. The following command saves all the settings and results in **ALU.ddc**).

```
write -hierarchy -format ddc
```

15. We can also generate a file to store all the design constraints we have set.

```
write_sdc ALU.sdc
```

16. Finally, you have to type the following command to save the timing information.

```
write_sdf -version 2.1 ALU.sdf
```

17. You should also write gate-level netlist for gate-level simulation.

```
write –format verilog -hierarchy -output ALU_syn.v
```

18. For Verilog gate-level simulation, you should add

```
$sdf_annotate("ALU.sdf", my_alu);
```

in an initial block in your test bench to use timing information for simulation.

19. Run the gate-level simulation in the command line.

```
vcs -full64 -f Lab3_alu_run_syn.f +v2k +neg_tchk -R
```

Checkpoints

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Answer the question: How many "GTECH_OR2" are there after HDL translation? _____
2. Take a snapshot of "SDF annotation".

```
    ***    $sdf_annotate() version 1.2R
    ***     SDF file: "ALU.sdf"
    ***     Annotation scope: test_alu.my_alu
    ***     No MTM selection argument specified
    ***     No SCALE FACTORS argument specified
    ***     No SCALE TYPE argument specified
    ***     MTM selection defaulted to "TOOL_CONTROL":
                 (+typdelays compiled, TYPICAL delays selected)
    ***     SCALE FACTORS defaulted to "1.0:1.0:1.0":
    ***     SCALE TYPE defaulted to: "FROM_MTM"
    ***     Turnoff delay: "FROM_FILE"
    ***     Approximation (mipd) policy: "MAXIMUM"

    ***     SDF annotation begin: Sun Oct  6 16:23:11 2024

SDF Info: +pulse_r/100, +pulse_e/100 in effect

            Total errors: 0
            Total warnings: 0
    ***     SDF annotation completed: Sun Oct  6 16:23:11 2024
```

3. Take a snapshot of the successful gate-level simulation results.

```
Chronologic VCS simulator copyright 1991-2022
Contains Synopsys proprietary information.
Compiler version T-2022.06_Full64; Runtime version T-2022.06_Full64;  Oct  6 16:23 2024
Doing SDF annotation ...... Done

Congratulations!! Your Verilog Code is correct!!

$finish called from file "Lab3_test_alu.v", line 120.
$finish at simulation time           6558735000
           V C S   S i m u l a t i o n   R e p o r t
Time: 6558735000 ps
CPU Time:      1.920 seconds;       Data structure size:   0.2Mb
Sun Oct  6 16:23:20 2024
CPU time: 4.350 seconds to compile + .515 seconds to elab + .750 seconds to link + 1.966 seconds in simulation
[r12008@cad31 Lab3]$
```

Submission

1. **Due Tuesday, Oct. 8, 19:00. No delay is allowed.**
2. Selected students need to take snapshots of the results shown in the previous section, record them into a PDF file, and submit it to NTU COOL.
   Title: CVSD_Lab3_studentID (E.g. CVSD_Lab3_r12943008.pdf)