Synthesis using Design Compiler with 0.13um Library

CSA & VLSI Lab.

Design : FIR_FILTER

Tool : Synopsys Design compiler

Library : IDEC 0.13um

Operation CLK : 40MHz

Tcl_example

- set top FIR_FILTER

- set target_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_typ_120_p025.db
- set link_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_typ_120_p025.db
- set symbol_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_veri.sdb
- # you must check library path (in SUN29 Marchine)

- read_verilog ../src/FIR_FILTER.v

- analyze -format verilog ../src/FIR_FILTER.v
- # must defined by your environment (check the path again!!!)

- current_design $top

- link
- uniquify

- set_fix_multiple_port_nets -all -buffer_constants

Tcl_example

```
## AREA ##
set_max_area 100000

## CLOCK ##
create_clock -period 25 [get_ports CLK]

set_clock_uncertainty -setup 0.5 [get_clocks CLK]
set_clock_latency -source 0.2 [get_clocks CLK]

set_clock_latency -max 0.1 [get_clocks CLK]

set_clock_transition 0.25 [get_clocks CLK]
set_dont_touch_network [get_clocks CLK]

## INPUT DELAY ##
set_input_delay -max 8 -clock CLK [all_inputs]
remove_input_delay [get_ports CLK]

## OUTPUT DELAY##
set_output_delay -max 8 -clock CLK [all_outputs]

## OUTPUT PORT LOAD ##

set_load 0.003  [all_outputs]

## INPUT PORT TRANSITION ##
set_input_transition 0.4 [all_inputs]
remove_driving_cell [get_ports CLK]

# Caution : check the CLK port name : it distinguish whether capital or not
# Must redefine your design stretegy
```

Tcl_example

```
## MAX CAPACITANCE ##
set_max_capacitance 3 [all_designs]

## MAX TRANSITION and FANOUT ##
set_max_transition 0.55 [all_designs]
set_max_fanout 15 [all_designs]

## OPERATING CONDITIONS ##
set worst V165WTP1250
set best V195BTN0400
set typical V180TTP0250

# COMPILE #
compile -auto_ungroup delay -boundary -map_effort high
remove_unconnected_ports [find -hierarchy cell {*}] -blast_buses

current_design $top

report_port -verbose                > ./log/$top.report_port_verbose.log
report_design                       > ./log/$top.report_design.log
report_constraint
report_constraint -verbose          > ./log/$top.report_verbose.log
report_constraint -all_violators    > ./log/$top.report_all_violators.log
report_timing -delay max > ./log/$top.report_delay_max.log
report_timing -delay min > ./log/$top.report_delay_min.log
report_area   > ./log/$top.report_area.log
report_power  > ./log/$top.report_power.log
report_clock  > ./log/$top.report_clock.log

change_name -h -rule verilog    > ./log/change_names.v

write -f verilog $top -h -o ./db/$top.v
write_sdf -version 1.0      ./db/$top.sdf
write_sdc                   ./db/$top.sdc
```

Synthesis

Netlist
+
PAD
(208pin)

Synthesis
Again

Netlist
modification
(pad part)

_PAD module example

```
module FIR_FILTER_PAD ( RESET, CLK, WR, iDATA, oDATA );
  input [15:0] iDATA;
  output [38:0] oDATA;
  input RESET, CLK, WR;
  wire   wRESET, wCLK, wWR;
  wire   [15:0] wiDATA;
  wire   [38:0] woDATA;

  FIR_FILTER FIR1 ( .RESET(wRESET), .CLK(wCLK), .WR(wWR), .iDATA(wiDATA),
       .oDATA(woDATA) );


          vssoh        pad1();
               vssoh   pad2();
               vssoh   pad3();
               vssoh   pad4();


  phic pad33 ( .PAD(iDATA[0]), .PI(1'b0),      .PO(), .Y(wiDATA[0]) );
  phic pad34 ( .PAD(iDATA[1]), .PI(1'b0),      .PO(), .Y(wiDATA[1]) );
  phic pad35 ( .PAD(iDATA[2]), .PI(1'b0),      .PO(), .Y(wiDATA[2]) );
  phic pad36 ( .PAD(iDATA[3]), .PI(1'b0),      .PO(), .Y(wiDATA[3]) );
                                  ⋮
// Given to you " FIR_FILTER_PAD.v"
// Refer to the pdf file in the STD150_IO in the manual folder for more detailed info. about  PAD cell
```

Tcl_Example(syn. with pad)

```
set top FIR_FILTER_PAD
# check the top module name is set as  _PAD module

set target_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_typ_120_p025.db
set link_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_typ_120_p025.db
set symbol_library /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_SYNOPSYS/syn/std150e_veri.sdb

read_verilog ../src_pad/FIR_FILTER.v
read_verilog ../src_pad/FIR_FILTER_PAD_F.v

analyze -format verilog ../src_pad/FIR_FILTER.v
analyze -format verilog ../src_pad/FIR_FILTER_PAD_F.v
# synthesis with pad

current_design $top

link
uniquify

set_fix_multiple_port_nets -all -buffer_constants
```

Tcl_Example(syn. with pad)

- ## AREA ##
- set_max_area 100000

- ## CLOCK ##
- create_clock -period 25 [get_ports CLK]

- set_clock_uncertainty -setup 0.5 [get_clocks CLK]
- set_clock_latency -source 0.2 [get_clocks CLK]

- set_clock_latency -max 0.1 [get_clocks CLK]

- set_clock_transition 0.25 [get_clocks CLK]
- set_dont_touch_network [get_clocks CLK]

- set_ideal_network [get_clocks CLK]
- # Must insert this command !!!!!
- -> After synthesis, it should be modified as the command : set_propagated_clock [get_clocks CLK] in the .SDC file

- ## INPUT DELAY ##
- set_input_delay -max 8 -clock CLK [all_inputs]
- remove_input_delay [get_ports CLK]

- ## OUTPUT DELAY##
- set_output_delay -max 8 -clock CLK [all_outputs]

- # the other constraint can be same as the initial synthesis, but it can also be changed depending on the designer's strategy.

MUST : Pad cell reorganization

```
module FIR_FILTER_PAD ( RESET, CLK, WR, iDATA, oDATA );
  input [15:0] iDATA;
  output [38:0] oDATA;
  input RESET, CLK, WR;
  wire   wRESET, wCLK, wWR;
  wire   [15:0] wiDATA;
  wire   [38:0] woDATA;

  FIR_FILTER FIR1 ( .RESET(wRESET), .CLK(wCLK), .WR(wWR), .iDATA(wiDATA),
       .oDATA(woDATA) );
  vssoh_p pad1 (  );
  vssoh_p pad2 (  );
  vssoh_p pad3 (  );
  vssoh_p pad4 (  );
  vdd33oph_p pad5 (  );
  vdd33oph_p pad6 (  );
  vdd33oph_p pad7 (  );
  vssoh_p pad8 (  );
  vssoh_p pad9 (  );
  vssoh_p pad10 (  );
  vssiph_p pad11 (  );
  vssiph_p pad12 (  );
  vdd12ih_core_p pad13 (  );
  vdd12ih_core_p pad14 (  );
  vssiph_p pad15 (  );
  vssiph_p pad16 (  );
  vssiph_p pad17 (  );
  vssiph_p pad18 (  );
  vssiph_p pad19 (  );
  vdd12ih_p pad20 (  );
  vdd12ih_p pad21 (  );
  vssiph_p pad22 (  );
  vssiph_p pad23 (  );
  vssiph_p pad24 (  );
  vssiph_p pad25 (  );
  vssiph_p pad26 (  );
  vssiph_p pad27 (  );
  vssiph_p pad28 (  );
  vssiph_p pad29 (  );
  vssiph_p pad30 (  );
  vssiph_p pad31 (  );
  vssiph_p pad32 (  );
  phic_p pad33 ( .PAD(iDATA[0]), .PI(1'b0), .Y(wiDATA[0]) );
  phic_p pad34 ( .PAD(iDATA[1]), .PI(1'b0), .Y(wiDATA[1]) );
  phic_p pad35 ( .PAD(iDATA[2]), .PI(1'b0), .Y(wiDATA[2]) );
  phic_p pad36 ( .PAD(iDATA[3]), .PI(1'b0), .Y(wiDATA[3]) );
  phic_p pad37 ( .PAD(iDATA[4]), .PI(1'b0), .Y(wiDATA[4]) );
  phic_p pad38 ( .PAD(iDATA[5]), .PI(1'b0), .Y(wiDATA[5]) );
  phic_p pad39 ( .PAD(iDATA[6]), .PI(1'b0), .Y(wiDATA[6]) );
  phic_p pad40 ( .PAD(iDATA[7]), .PI(1'b0), .Y(wiDATA[7]) );
  phic_p pad41 ( .PAD(iDATA[8]), .PI(1'b0), .Y(wiDATA[8]) );
  phic_p pad42 ( .PAD(iDATA[9]), .PI(1'b0), .Y(wiDATA[9]) );
```

After synthesis,
intrinsic pad cell name should be reorganized like left figure (in the final netlist)

"_p": must attach all the pad cell name before the P&R entered.

(otherwise, Astro will not read your netlist.)

VCS Library path for PRE/POST Simulation

- /Tools/Library/SAMSUNG_0.13um/cellbased/SEC150E_VCS/verilog/STD150e.v
- The Other VCS way to use are perfectly same.
- Post Simulation for VCS is available when you use 0.13um Library