Exercise

Finish the following lab and answer the questions.

1. How long is the setup time and hold time of the flipflops used in the design?
2. What is the largest delay of the combinational paths, and how fast can the GCD operate?
3. How many standard cells included in the GCD?

1. Setup EDA Environment

% setdt vcs

% setdt syn

2. Synthesize GCD

a) Copy the file **/js1/songch/DA_VLSI/DA_VLSI_Lab3.tgz** to your home directory.

b) Extract the files % **tar zxvf DA_VLSI_Lab3.tgz,** and enter this directory.

c) Remove simulation module from **gcd_rtl_2009.v** and save as the new file **gcd_rtl_2009_syn.v**.

d) **% dc_shell-t**

e) If the dc_shell-t invoked successfully, you will get the following command prompt.

dc_shell>

dc_shell>source ./rm_setup/dc_setup.tcl // Setup variables and libraries

//If you want to write your own setup script, you can refer Step e in the following tutorial.

dc_shell> **analyze -format verilog** ./verilog/gcd_rtl_2009_syn.v // translate the verilog descriptions, here we place the Verilog source file in the directory ./verilog

dc_shell> **elaborate** gcd_rtl_top // module name of your top-level design

dc_shell> link // link to the libraries setup in link_library

f) To confirm the circuits you read in, you can run the following commands

dc_shell> list_designs // Show what are read into the memory

dc_shell> current_design // show the name of current design

dc_shell> list_libs // Show the library read into the memory (saed32rvt_ss0p95v125c, etc.)

dc_shell> all_inputs // show all the input ports dc_shell> all_outputs // show all the output ports

To constrain your design

dc_shell> create_clock –period 5 [get_ports clk] // Please see the appendix for the basic constraints.

g) To optimize your circuits

dc_shell> **compile** or **compile_ultra**

// optimization, compile/compile_ultra (use 'man command_name' to view the manual of the command)

After you optimize the circuits, you can confirm the performance, area and power.

```
dc_shell> check_design                // check the synthesized results
dc_shell> report_timing
dc_shell> report_area
dc_shell> report_power
```

h) To save your optimized circuit

```
dc_shell>change_names  -rule verilog  -hier
dc_shell>write  -f verilog  -hier  -o ./results/gcd_map.v
dc_shell>write  -f ddc  -hier  -o ./results/gcd_map.ddc
                //saved as .ddc format, a format for interaction between Synopsys tools.
```

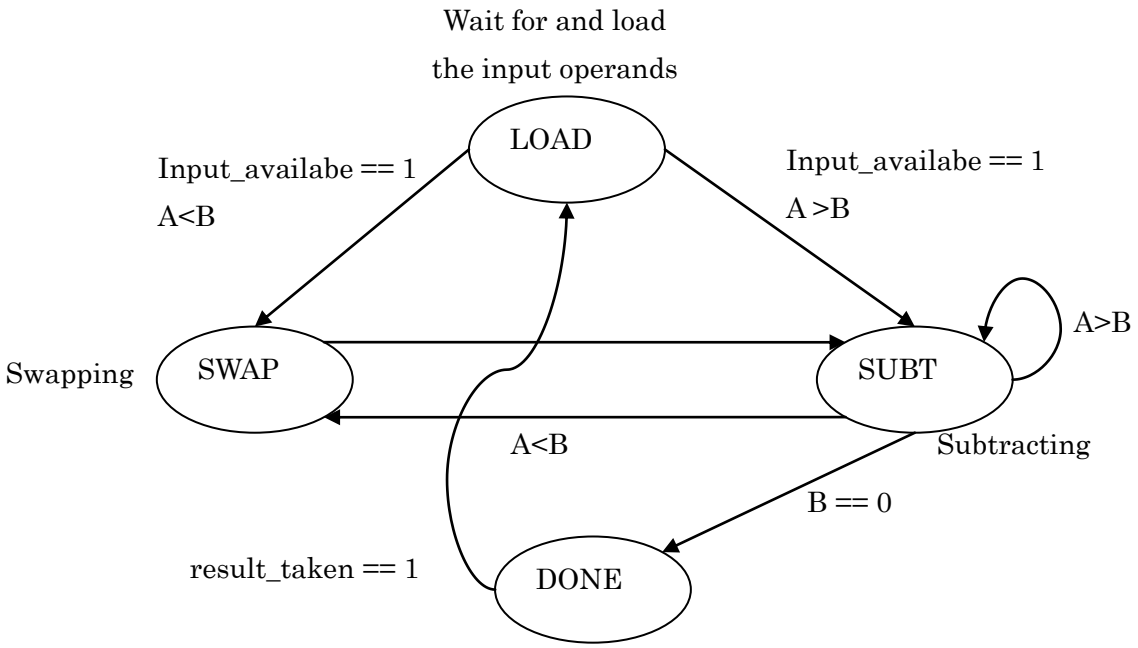
3. Optimize the Control Unit Module of GCD

a) A RTL design for computing greatest common divisor (GCD) is shown in Lab 2.
In the example, the FSM with three states, WAIT, CALC, and DONE, is used to control the data path unit. Here, you are asked to devise a four-state FSM, shown in the following figure, for the control unit of GCD.

b) You may use the code for lab3 and just implementing the FSM in Verilog by modifying the control unit module **gcdGCDUnitCtrl**.

c) Verify the functionality by simulation.

d) Synthesize your Verilog code to Gate-level Netlist.



Tips:

After you exit design compiler occasionally, you can invoke the dc_shell-t,
and use the following commands to read the synthesized result.

dc_shell> source dc_setup.tcl // Setup variables and libraries

dc_shell>read_ddc ./results/gcd_map.ddc

Please refer to **http://staff.ustc.edu.cn/~songch/da-ug.htm** for the following resources

1. Verilog-basics-2 for the Verilog HDL.

2. Documents for Design Compiler: http://202.38.80.152/dc_doc/tcoug.pdf (Synopsys Timing Constraints and Optimization User Guide)

Please read the error information carefully if any.

Constraining your design

- Design Compiler generates gate-level circuits with given design constraints satisfied.
  - Without constraints, DC cannot optimize
- Design Constraints
  - Design rules from technology library
  - Optimization Constraints
    - Timing, Area, etc.
- Constraint priorities
  - Design rule, Timing, Area

Optimization Constraints

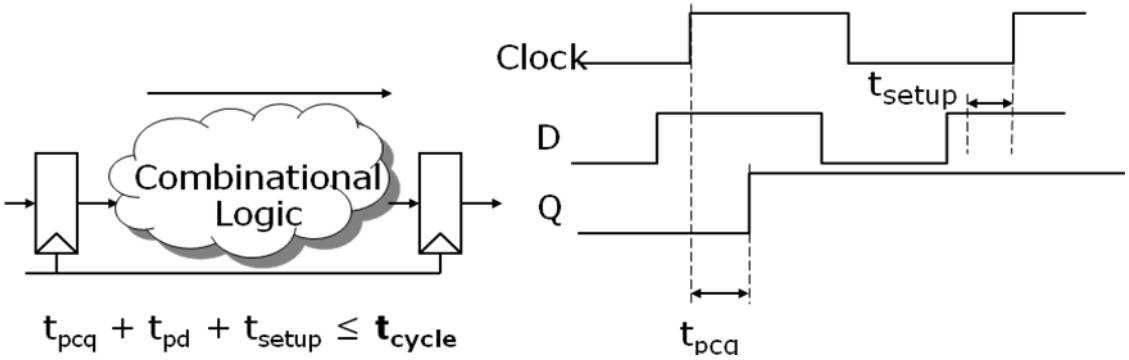
- Maximum delay constraint

  For combinational circuits, setup the target delay
  - *set_max_delay 10 –from [all_input] –to [all_output]*
- Maximum area constraint
  - *set_max_area 0*
- Clock setup
  - **create_clock –period 10 [get_ports clk]**

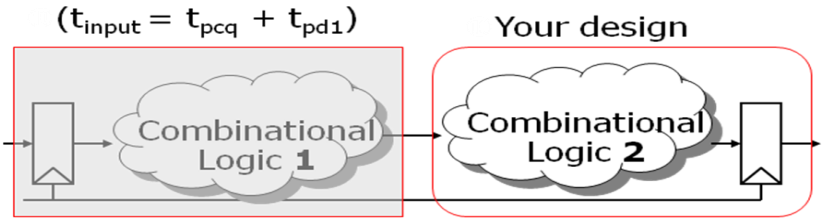
    Define a clock with source at port **clk** and period **10**

Setup time constraint

- By defining a clock, the combinational paths between registers must satisfy the setup time constraint.

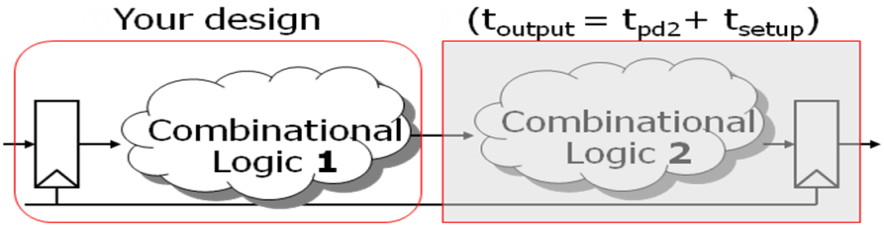


- The combinational paths from intput ports to registers must meet the setup time constraints
  - set_input_delay 5 –clock clk [get_ports *input_port_name*]



$t_{input} + t_{pd2} + t_{setup} \leq t_{cycle}$
t_{input}: the external input delay

- **The combinational paths from registers to output ports must meet the setup time constraints**
  - **set_output_delay 5 –clock clk [get_ports *output_port_name*]**

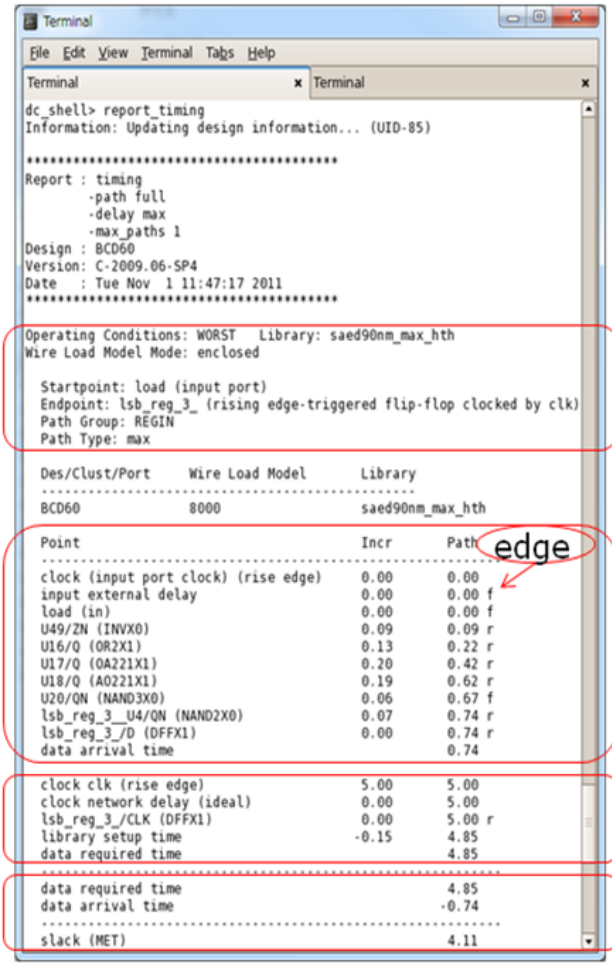


$t_{output} + t_{pd1} + t_{pcq} \leq t_{cycle}$
t_{output}: the external output delay

Timing Analysis

- **Do setup and hold check for all the combinational paths (static timing analysis)**
  - **report_timing (dc_shell command)**
  - ***-delay maximum* : [default]report the paths with maximum delay**
  - **-delay minimum : report the paths with minimum delay**

Examples



Critical Path Summary

1. Start point, endpoint, library, wire load model
2. Path type

max: **setup** time check

min: **hold** time check.

Critical Path Details

Incr: Increased delay

Path: Accumulated Delay

Setup Timing Check for the critical path

clock_network_delay: clock skew

library_setup_time: setup time for registers

Slack: the margin between the arrival time and required time.

MET: constraints met (positive)

VIOLATED: constraints violated

clock clock (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
gcddatapath/B_reg[0]/CLK (DFFPOSX1)	0.00	0.00 r
gcddatapath/B_reg[0]/Q (DFFPOSX1)	0.15	0.15 r
gcddatapath/sub_16/B[0] (gcd_dpath_DW01_sub_0)	0.00	0.15 r
gcddatapath/sub_16/U23/Y (INVX1)	0.11	0.26 f
gcddatapath/sub_16/U6/Y (OR2X1)	0.15	0.40 f
gcddatapath/sub_16/U2_1/YC (FAX1)	0.21	0.61 f
gcddatapath/sub_16/U2_2/YC (FAX1)	0.21	0.83 f
gcddatapath/sub_16/U2_3/YC (FAX1)	0.21	1.04 f
gcddatapath/sub_16/U2_4/YC (FAX1)	0.21	1.25 f
gcddatapath/sub_16/U2_5/YC (FAX1)	0.21	1.47 f
gcddatapath/sub_16/U2_6/YC (FAX1)	0.21	1.68 f
gcddatapath/sub_16/U2_7/YC (FAX1)	0.21	1.89 f
gcddatapath/sub_16/U2_8/YC (FAX1)	0.21	2.11 f
gcddatapath/sub_16/U2_9/YC (FAX1)	0.21	2.32 f
gcddatapath/sub_16/U2_10/YC (FAX1)	0.21	2.53 f
gcddatapath/sub_16/U2_11/YC (FAX1)	0.21	2.75 f
gcddatapath/sub_16/U2_12/YC (FAX1)	0.21	2.96 f
gcddatapath/sub_16/U2_13/YC (FAX1)	0.21	3.17 f
gcddatapath/sub_16/U2_14/YC (FAX1)	0.21	3.39 f
gcddatapath/sub_16/U2_15/YS (FAX1)	0.23	3.62 r
gcddatapath/sub_16/DIFF[15] (gcd_dpath_DW01_sub_0)	0.00	3.62 r
gcddatapath/U9/Y (AOI22X1)	0.05	3.67 f
gcddatapath/U8/Y (NAND2X1)	0.08	3.75 r
gcddatapath/A_reg[15]/D (DFFPOSX1)	0.00	3.75 r
data arrival time		3.75
clock clock (rise edge)	5.00	5.00
clock network delay (ideal)	0.00	5.00
gcddatapath/A_reg[15]/CLK (DFFPOSX1)	0.00	5.00 r
library setup time	-0.18	4.82
data required time		4.82
data required time		4.82
data arrival time		-3.75
slack (MET)		1.06

t_{pcq}

t_{pd}

t_{setup}

$$t_{pcq} + t_{pd} + t_{setup} \leq t_{cycle}$$

Clock Skew

clock skew + reserved margin before the layout

- set_clock_uncertainty–setup time_value [get_clocks clk]
- setup time check
  - Use the possible earliest arrival time of clock edge
- Hold time Check
  - Use the possible latest arrival time of clock edge