Tutorial on Cadence Genus Synthesis Solution

EE 201A – VLSI Design Automation – Winter 2026 UCLA Electrical Engineering

Instructor: Prof. Puneet Gupta

TA: Dedeepyo Ray

Based on slides from: Irina Alam/Mark Gottscho/Vishesh Dokani/Shurui Li

Logic Synthesis

- *RTL*: *Register-Transfer-Level* description of logic design (e.g., c = a + b)
  - Allows both *Behavioral* (add two numbers) and *structural* (connect module A to module B) design styles
  - Focus on higher-level functionality, e.g. sequential logic, pipeline stages, maybe even micro-architectural design
  - *Verilog/VHDL* are industry standard *hardware description languages (HDL)* used for RTL
- *Logic synthesis*: Automatically reduce RTL to *gate-level logic* (e.g., AND, NOT, wires, etc.) with help of a logic library
  - Still represented in Verilog, but lose intuition of higher abstraction. Purely structural design style
  - In fact, you can write your own gate-level logic implementation! Try it … ;)
    - … and you will rapidly appreciate the power of logic synthesis!
  - RTL `s1494.v` → gate-level `s1494_synth.v`
- Software analogy: compile C to assembly language

RTL vs. Gate-Level Example in Verilog

- RTL:

```
always @(posedge clock)
if(add1 && add2) r <= r+3;
else if(add2) r <= r+2;
else if(add1) r <= r+1;
```

- Logic gates:

```
AOI22_X1 g15017(.A1 (n_212), .A2 (n_249), .B1 (n_237), .B2
    (n_210), .ZN (n_322));
NOR2_X1 g15020(.A1 (n_211), .A2 (n_210), .ZN (n_246));
NOR2_X1 g15021(.A1 (n_275), .A2 (n_148), .ZN (n_209));
NOR2_X1 g15023(.A1 (n_258), .A2 (n_117), .ZN (n_208));
INV_X1 g15024(.A (n_207), .ZN (n_289));
```

Nangate* Open Cell Library

*not NANDgate! ☺ www.nangate.com

- Commonly used set of *standard cells* that are treated as primitives in the HDL
  - Needed for logic synthesis as well as physical design
- Includes logic representation
  - AND gate example: `ZN = (A1 & A2);`
- Includes physical design information
  - Cell timing
    - Capacitance, Delay, Transition time, etc.
  - Cell power
    - Leakage power, Capacitance, etc.
  - Cell area
- Different libraries for different *design corners*
  - NangateOpenCellLibrary_typical.lib for *typical corner*

Genus

- Industry standard synthesis suite.
- 2019 version of the traditional Cadence Encounter RTL Compiler (RC).
- Logic as well as physical synthesis.
- Genus has a Legacy UI to directly run old commands from RC.
  - Not permitted for Lab 2. You must use updated Genus commands.

Tool Setup & Documentation

- Use same setup files:

```
/w/class.1/ee/ee201a/ee201at2/csh_ee201a_setup
```

- Launch Genus with GUI (if display variable set)

```
$ genus -gui
```

- Launch Genus without GUI using Tcl (Tool Command Language) script of input commands

```
$ genus < synth.tcl
```

- Genus Documentation

```
$ acroread <path>.pdf
/w/apps3/Cadence/GENUS191/doc/{genus_comref,
  genus_attref, genus_user_legacy}
Example: $ acroread
  /w/apps3/Cadence/GENUS191/doc/genus_comref/genus_comref.pdf
```

  - Use Legacy User Manual only as a reference methodology guide.

Genus Logic Synthesis (1) - Example

```
set_db information_level 9  # Debug verbosity from 0 to 9
set_db hdl_error_on_blackbox true  # Generate error when cannot map a block (a cell in
   the library is missing)
set_db init_hdl_search_path {./}  # Path to Verilog design files
read_hdl -v2001 verilog_file # Read Verilog design files
set_db init_lib_search_path {./} # Path to .lib (Liberty, not library suffix)
   timing library
set_db library liberty_filename.lib    # Set filename of the Liberty file for our
   Nangate library
elaborate name_of_top_level_module # Initialize design from Verilog file (.v)

# Apply design constraints for logic synthesis
# Define clock with 1000ps period, no external timing slack, clock transition time
   (slew rate) of 100ps
set clock [define_clock -period 1000 -name ${clkpin} [clock_ports]]
set_input_delay -clock ${clkpin} 0 [vfind /designs/${DESIGN}/ports -port *]
set_output_delay -clock ${clkpin} 0 [vfind /designs/${DESIGN}/ports -port *]
dc::set_clock_transition .1 ${clkpin}

check_design -unresolved  # Check for unresolved refs & empty modules
```

Genus Logic Synthesis (2)

```
# Perform logic synthesis: technology mapping + logic optimization
syn_generic
syn_map
syn_opt

# List possible timing problems
report_timing -lint

# Output some useful results of synthesis
report_timing > output/synth_report_timing.txt
report_gates > output/synth_report_gates.txt
report_power > output/synth_report_power.txt

# Write the synthesized netlist as Verilog HDL
write_hdl > output/s15850_synth.v

# Write design constraints to SDC file
write_sdc > output/s15850.sdc
```

Genus Set Attribute Syntax

- Change of syntax for setting attribute values:

RTL Compiler	Genus
`set_attribute`	`set_db`

- Genus: All kinds of object attributes grouped under same command
  - Root attributes:

    ```
    set_db attribute_name value
    ```

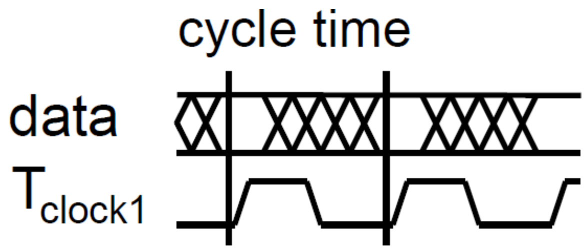
  - Design attributes:

    ```
    set_db design:/$DESIGN .attribute_name value
    ```

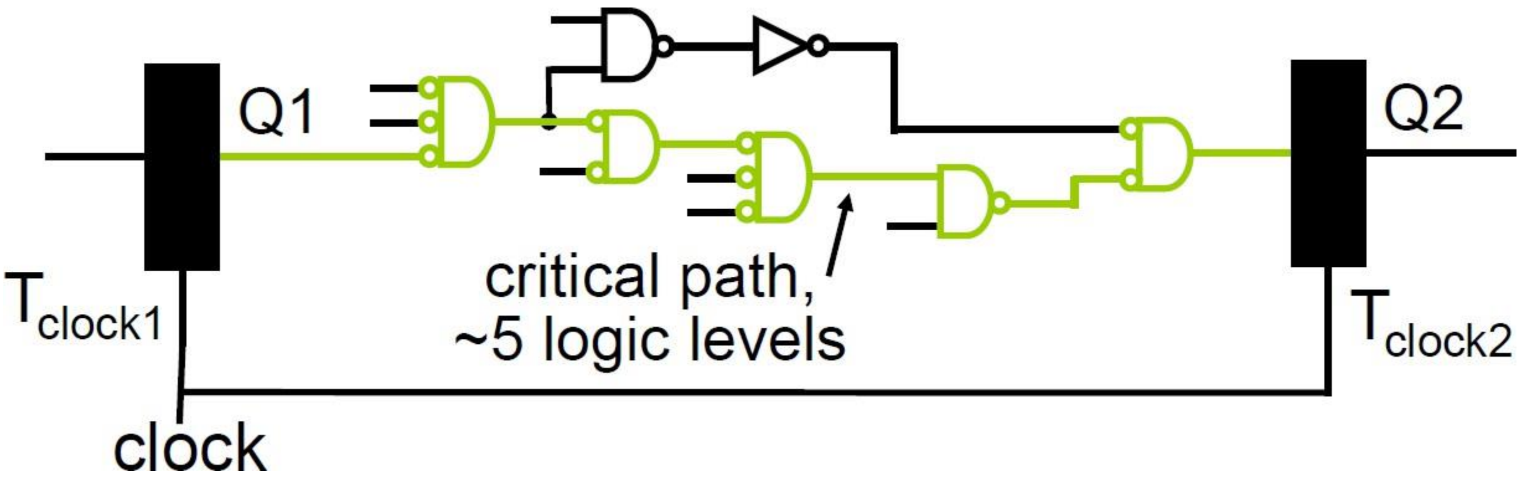
  - And so on for other object types

Getting Started With Lab 2 (1)

- Cycle time (T) cannot be smaller than longest path delay (Tmax)
- Longest (critical) path delay is a function of:
  - Total gate, wire delays
- **Slack = T - T_{max}**



$T_{max} \leq T$



Source: EE201A Lecture Slides

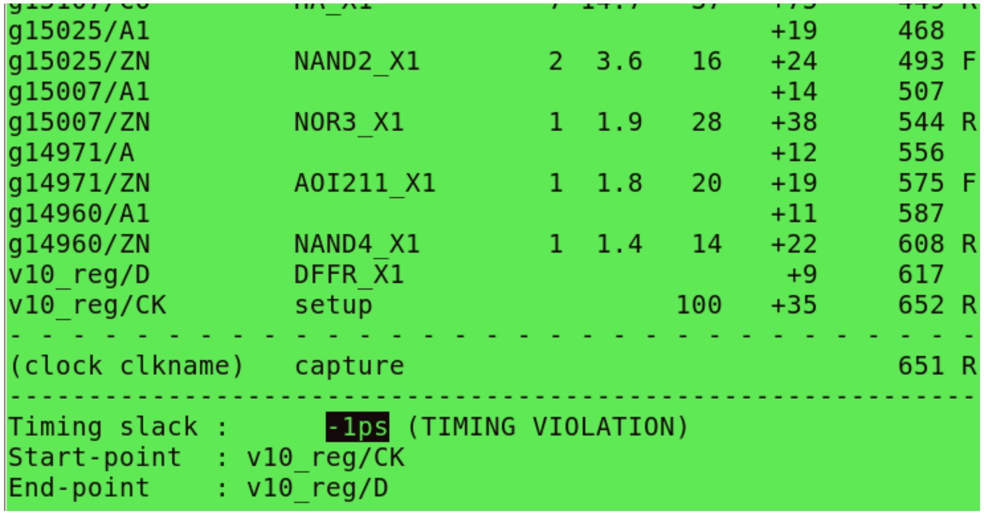
Getting Started With Lab 2 (2)

- **Run skeleton script:** `$ genus < lab2_skeleton.tcl`
- **Change clock period in script**

```
set clock [define_clock -period 651 -name clk [clock_ports]]
```

- **Check slacks**

```
genus:/>report_timing
```



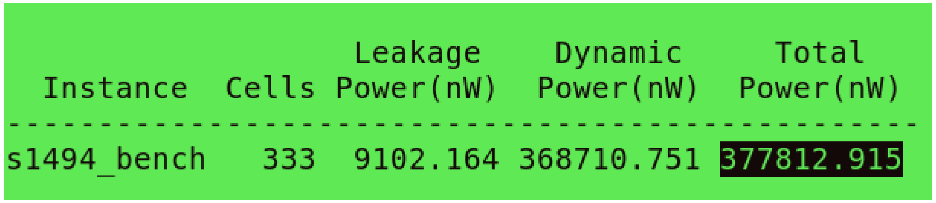
```
g15025/A1                                    +19     468
g15025/ZN          NAND2_X1     2  3.6   16  +24     493 F
g15007/A1                                    +14     507
g15007/ZN          NOR3_X1      1  1.9   28  +38     544 R
g14971/A                                     +12     556
g14971/ZN          AOI211_X1    1  1.8   20  +19     575 F
g14960/A1                                    +11     587
g14960/ZN          NAND4_X1     1  1.4   14  +22     608 R
v10_reg/D          DFFR_X1                    +9     617
v10_reg/CK         setup                100  +35     652 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clkname)    capture                           651 R
-------------------------------------------------------------
Timing slack :        -1ps (TIMING VIOLATION)
Start-point  : v10_reg/CK
End-point    : v10_reg/D
```

  - Slack: -1ps (delay of critical path is 652ps)

Getting Started With Lab 2 (3)

- **Check power**

```
genus:/> report_power
```



```
                         Leakage    Dynamic     Total
  Instance     Cells  Power(nW)  Power(nW)   Power(nW)
---------------------------------------------------------
s1494_bench     333   9102.164 368710.751  377812.915
```

- **Improve your design**
  - Read user and command reference manuals
  - What Genus commands do you think could improve your delay, area, and/or power?
  - Add Genus commands into the “right” positions in script
  - Experiment with ordering of commands during synthesis

Lab 2 Goals

- Work with Cadence Genus
  - to understand synthesis flow and familiarize yourself with a useful tool in industry
  - manipulate CAD tool to generate a better design from the same RTL
  - Learn basic Tcl scripting