## Delay Prediction Homework

- All Spice problems in this homework are to be done for technologies
  - If last digit of Student ID # is odd use Technology: tsmc018.model, Vdd = 2.5 V, default temp
  - If last digit of Student ID # is even use Technology : ami06.m, Vdd=3.3V, default temp
  - all input waveforms should have rise/fall times of 200 ps.

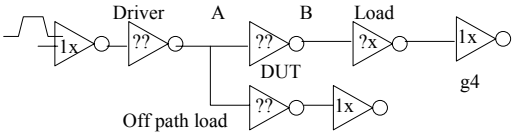


## Problem Statement

- Build 2D Transition, Propagation delay tables for 1X, 2X, 3X inverters
- Build Tplh/Tphl delay tables, rising/falling output transition tables
  - A total of 3 devices * 4 tables = 12 tables.
- Capacitive load points are measured in inverter equivalent loads. Table Cap load points should be: 1X, 3X, 6X, 12X inverter loads.
- Use three values for input transition time measured 30% to 70%.
  - Minimum input transition case: Largest inverter(3X) driving DUT
  - Maximum input transition case: 1X inverter driving DUT + off path load such that total load is 12X
  - Typical transition case (you pick).
  - Transition indexes are different for each DUT!
- Total number of points in table will be 3 * 4 = 12 data points.



## Part #1 – Data Measurement




a. Choose driver and offpath load for input transition case (minimum, maximum, typical).

b. Measure TPLH, TPHL for the gate marked as DUT for output loads of 1X, 3X, 6X, 12X

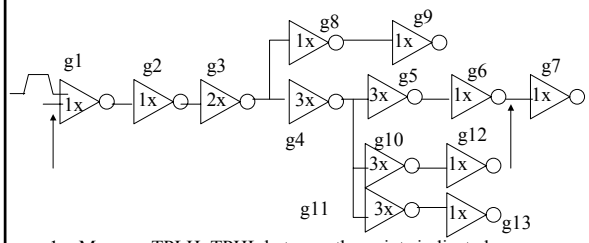
c. Measure Output Transition Rising, Output Transition falling same loads.

d. Repeat a-c for all driver cases

e. Repeat a-d for all DUT cases (1X, 2X, 3X).



## Part #2: Delay Prediction




1. Measure TPLH, TPHL between the points indicated.
2. Compute the delay based upon your lookup tables using Gaussian elimination (see Synopsys Library Compiler Guide, Chap 2, pgs 2-25 thru 2-28).



## Comparison

- Do delay computations by hand
  - Since starting rise/fall time = 200 ps, then the 30%-70% time for this would be 0.4*200 ps = 80 ps starting input transition time
- Plug data into Synopsys, compare your delays against what Synopsys predicts.
- Compare both against Spice measurements.
- Will talk more about Synopsys formats later
- For now, just gather data – Perl scripting could help!!!



## 'C' Code for Gaussian Elimination

- I have provided some C code for solving the set of equations required for the 2-D interpolation of the look up tables
  - mygauss.c -- contains 'gaussj' procedure which does the solving, and a 'main' routine for driving it. The main routine uses the equations from the Synopsys documentation
  - To compile, just do 'make' (I have provided a Makefile for doing the compilation.
- When executed, 'mygauss' will print out the starting matrixes, then print out the solution 'B' matrix.
- Make use of this code in any way that you see fit to perform the hand calculation

## Synopsys .lib file

- I have provided a Synopsys .lib file called *lab1.lib*
  - Only contains one cell definition 'INVX1' and this contains dummy data
  - You will have to add definitions for INVX2, INVX3 and provide the correct lookup table data
  - Order of pin definitions is IMPORTANT – Verilog model assumes that output pin is first, followed by input pin.
- To compile this library do:
  - swsetup synopsys
  - dc_shell –f compile_lab1.script



## Using Synopsys to Compute Delays

- I have provided a Verilog file that defines the test circuit
  - Name of file is 'path.v'
- A Synopsys script file is provided that can be used to compute delay, and internal net transition times, net loading
  - Script file name is 'path.script'
- To run synopsys using this script do:
  - dc_shell –f path.script



## Synopsys Docs

- Path to Synopsys Docs on 2-D interpolation

/opt/ecad/synopsys/default/doc/online/library/lcug2/lcug2_2.pdf

The PDF file is chapter 2.

lcug2 = Library Compiler User Guide, Part 2.



## Report

- Must be in a file titled 'report.pdf'
- Have a table that compares
  - your calculations vs. Synopsys
  - Your calculations vs Spectre
  - Synopsys vs Spectre
  - Give percent error for all comparisons . If 'x' is the golden value, then % error is 'y-x'/x * 100
  - If calculations/Synopsys deviate by more than 10% from Spectre, give me an explanation – stage by stage delay comparisons might help.
- Give your 'typical' drive choices and a rationale
- Explain the files in your submission and the methodology you used to produce your results. I want to be able to duplicate everything that you have done. This includes
  - Producing the raw data
  - Running Synopsys on your library file
  - Running any 'C' programs or equivalent that helped you with the delay calculations



## Submission Files

- I want all of your files except for Spectre simulation results (delete this directory before submitting your archive).
- Put all files in a directory called 'sim1'. Execute the submission script from the directory above 'sim1'
  - Do 'perl submit_ee8273_sim1.pl'
  - Will create a UU-encoded tar archive of your submission and email it to me
  - You can submit multiple times, I only look at the last one.