
DO NOT SHARE
SLIDES AND CLASS MATERIALS
ON ONLINE SITES
Course Hero

CSEE 4823 Lab#1
Design flow & Matlab

Xiaofu Pei

25th September 2020

Acknowledgement: Dave Fick at University of Michigan & previous TAs

Outline

- Introduction
- Topics
- Design Flow
- Project Example
  - LFSR (Linear-feedback shift register)
    - Matlab

Introduction

- TA: Xiaofu Pei(xp2175@columbia.edu)
- Lab Session:
  - Time: Friday
    - Session1: 9:30am - 11am
    - Session 2: 3:30pm - 5:00pm
  - Location: via Zoom
  - Office Hours : Integrated with lab session
- Q&A Sessions:
  - After the lab tutorial
- Topics: various CAD tools used in design flow

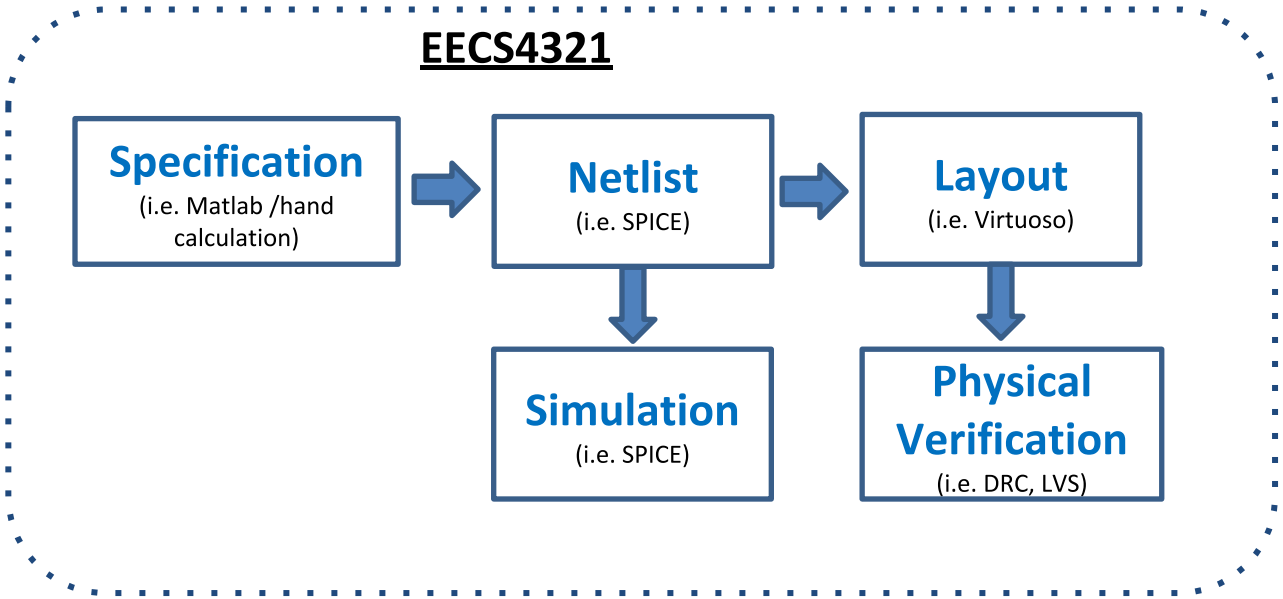
Topics

- Lab#1: Design flow & Matlab®
- Lab#2: Verilog HDL / ModelSim®
- Lab#3: Synthesis / Design Compiler®
- Lab#4: Timing and power analysis / PrimeTime®
- Lab#5: Memory Compiler
- Lab#6 and following labs: Project based lab

(Lab 3 and Lab 4 may be combined to meet schedule)

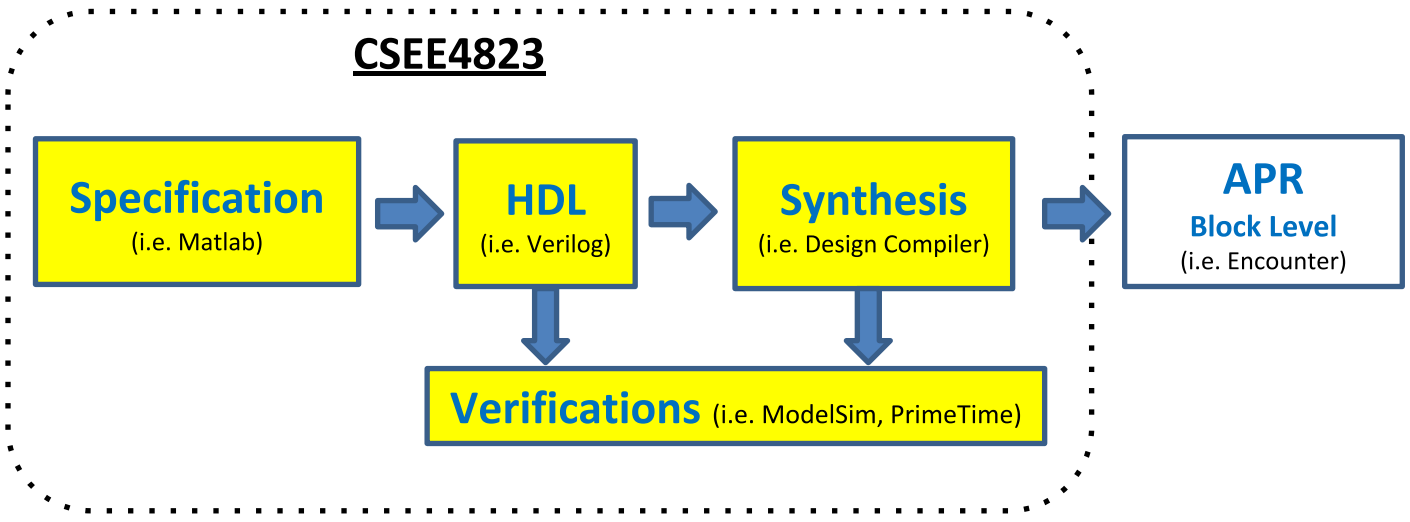
Design flows

- Custom Design Flow (EE4321 Digital VLSI circuits)
  - Analog Block
  - Area/Timing-critical digital blocks
  - Structured blocks (e.g. memories)

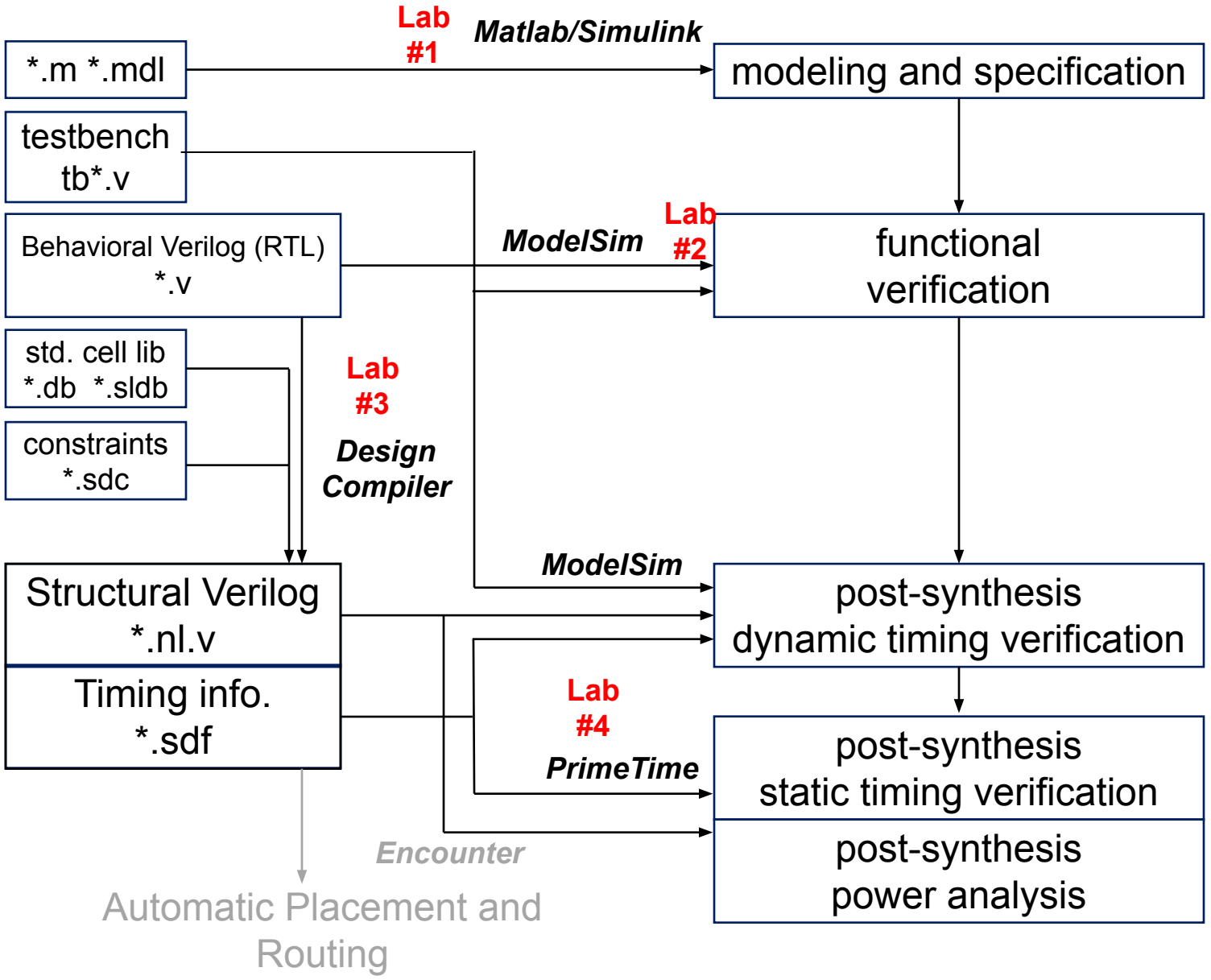


Design flows

- Semi-Custom Flow (**CSEE4823**, EE6920, EE6321)
  - Digital blocks
  - Need standard cells for synthesis
  - Large scale digital circuits (e.g. >10^6 transistors)
  - Reduces design time at the expense of performance compared to custom flow



Semi-Custom Flow



Timing Verification

- Static Timing Analysis (STA)
  - Checking all possible paths in a given combinational module
  - Covers all fast and slow paths to check for setup and hold violations
  - Tools: **PrimeTime** (Synopsys), Encounter (Cadence)

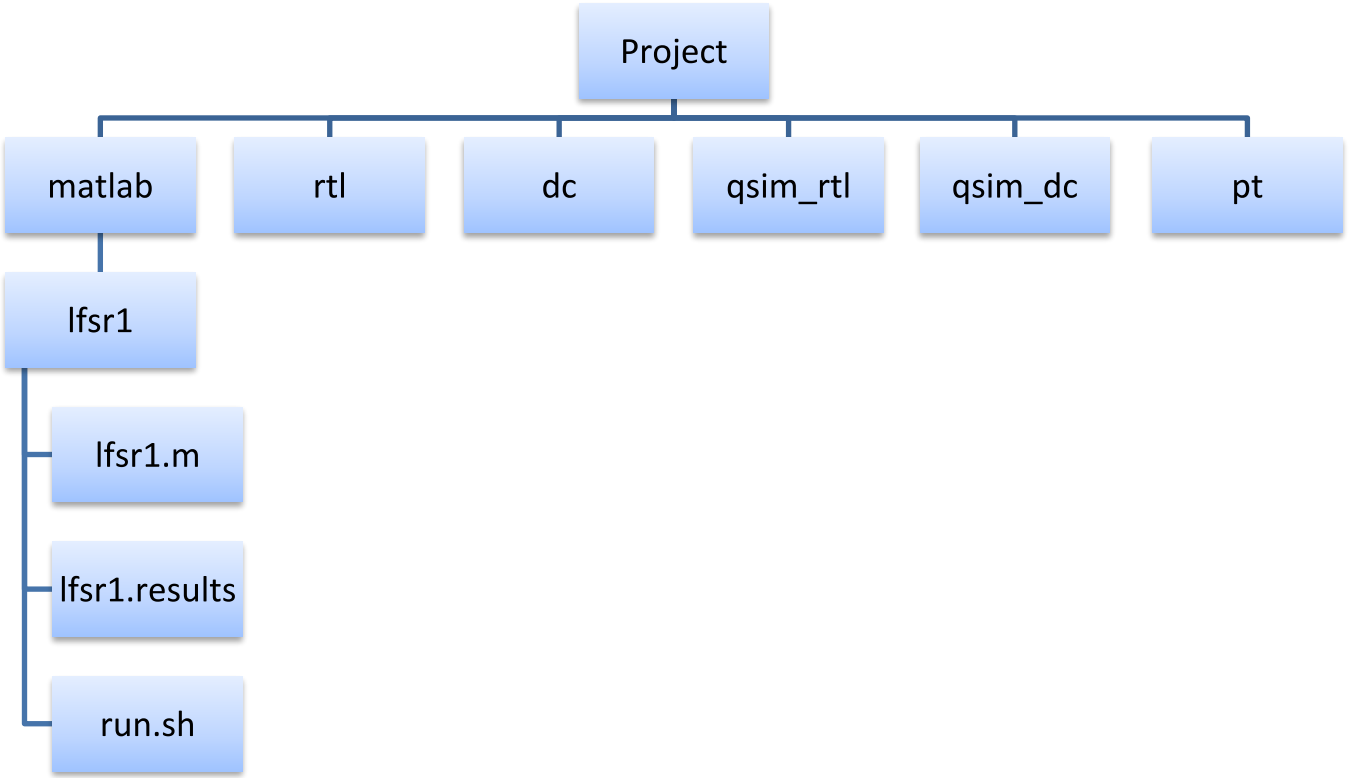
- Dynamic Timing analysis
  - Run verification testbenches at an operating frequency (important!)
  - Check for setup and hold time violations (if the library supports)
  - Tools: **ModelSim** (a.k.a. vsim and questasim), VCS, VerilogXL,

Before we start…

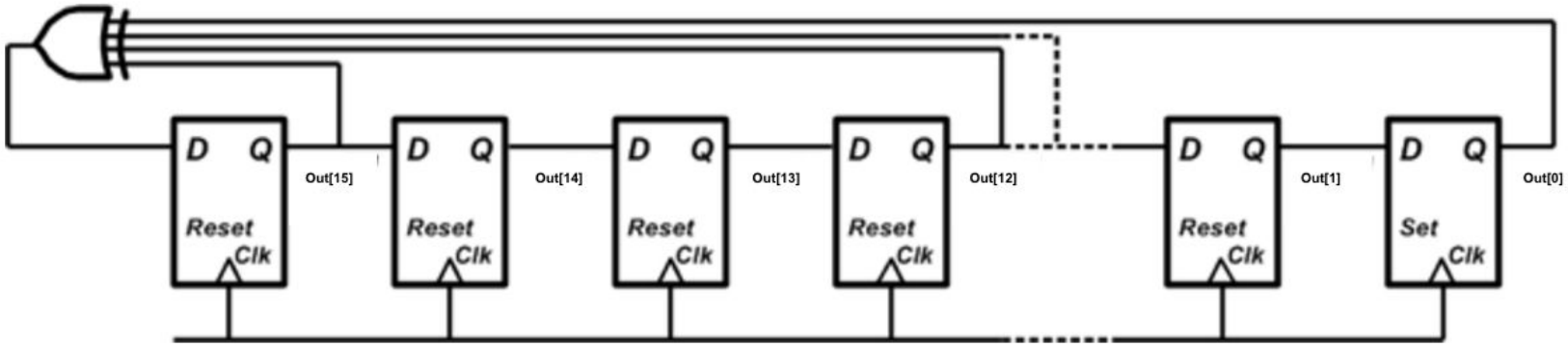
- Linux basic command (mkdir, rm, mv, cp …)
- For remote access: ssh, scp...
- Tools for editing scripts (gedit, vim, emacs…)
- Shell scripts (use *.sh to facilitate the simulation)
- Other useful tools for post-processing: Perl, Python…

Project Example - LFSR

- Setup your computer
  - Create your local folders (recommend the similar folder structure as the source code folder used)
  - Copy Matlab source codes from /courses/ee6321/share/project/matlab/lfsr1 to your local folder.



Project Example - LFSR



- Linear Feedback Shift Register
- Pseudorandom pattern generators
  - Generates periodic sequence
  - Start in a non-zero state (seed)
- FFs plus a few XOR gates
  - XOR outputs and feed back into input of one FF

Matlab

- First look into these files:
  - ***lfsr1.m***: main Matlab code - behavioral model
  - ***lfsr1.results***: results output file
  - ***run.sh***: used to run the Matlab, defining the running options, input and output files.
- Run Matlab with ***bash run.sh*** or ***./run.sh*** and check lfsr1.results file.

MATLAB Essentials

- **Computation -> powerful tool for matrix**
  - declare matrix (zeros, ones, randi)
  - index into the matrix (from 1)
  - access to subset of matrix
  - operations (built-in functions, e.g., sum, multiplication, element-wise operations, mean, exponentiation)
  - bitwise logical operations (bitand, bitor, ...)
  - data conversion to fixed-point
  - cell array
  - ...

MATLAB Essentials

- **Programming**
  - conditional statements (if-else, switch-case)
  - loops (for, while, ...)
  - function call
  - file I/O (fopen, fprintf, fclose...)
  - code section (%%)
- **Domain-specific toolbox**
  - DSP/ISP/CV….
  - Deep Learning…
  - a lot to explore

Lab Assignment

- Get familiar with Linux
- Try to write your own circuits in matlab and verify
- No submission needed this week