CSE 141L: Introduction to Computer Architecture Lab
Synthesis & Timing

Pat Pannuto, UC San Diego
ppannuto@ucsd.edu

Milestone 2 is due in 48 hours

- What to submit?
  - SOMETHING
- M1 feedback
  - Pay attention to things that should be revised for M2
  - Make changes obvious!
- M2 is about proving individual components work
  - How would you prove to your manager your component works?

Today's Objectives:
Understanding [a little bit of] Synthesis

- Difference between simulation and synthesis

- The parts of synthesis we cover in 141L
  - In particular, timing analysis

First, some gargantuan disclaimers

- The material presented today is a vastly simplified overview
- Present an imperfect understanding sufficient for needs in 141L
- Synthesis is *very* domain specific
  - And ultimately so is hardware design
  - Best practice for ASIC != Best practice for FPGA != …
  - Real-world, high-performance designs floorplan (coarsely) with arch design!

Okay, but what if I really do want to learn all of this stuff?

- In most cases, advanced hardware design is masters / PhD level
- Anything niche/specialized becomes more an *apprenticeship* than a *class*
- Who does this stuff in UCSD CSE? [n.b., there are many in ECE as well!]
  - Ryan Kastner (high performance FPGAs)
  - C.K. Cheng, Andrew Kahng, Alex Orailogu (Circuit Simulation, VLSI, and EDA Tools)
  - Hadi Esmaeilzadeh, Leo Porter, Pat Pannuto, Steven Swanson, Dean Tullsen, Yiying Zhang, Jishen Zhao ("Architecture"; very full stack)
  - Rajesh Gupta, Ryan Kastner, Pat Pannuto, Tajana Rosing (applied / embedded)
- Who did this stuff for a living for a long time?
  - John Eldon

Okay, but what if I really do want to learn all this stuff? Step 1: ***Try it yourself***

- There are a ton of great resources online for hardware development
  - My first google hit for 'gate delay fpga' is a *great* post:
    - https://stackoverflow.com/questions/8874705/how-can-i-calculate-propagation-delay-through-series-of-combinational-circuits-u
- Play around with the tools *beyond* what I show in class
  - Make a new workspace, play with basic circuits, look at all the reports the tools generate, look at some of the files in the work/ folders, try some of the other tools, etc etc
- You will also need some more of the ECE fundamentals
  - Mostly the solid-state electronics course pathway

Simulation is fast, and comparatively simple

- It looks at 'conceptually, how do we want hardware to behave?'

- It doesn't always map to things that can actually happen!
  - Evaluates time as "all the things that logically happened in this time step"

```
assign b = a;
assign c = #1 a;

initial begin
  a = '1;
  #5;
  a = '2;
end
```

"When" do b and c "become" 1 and 2?

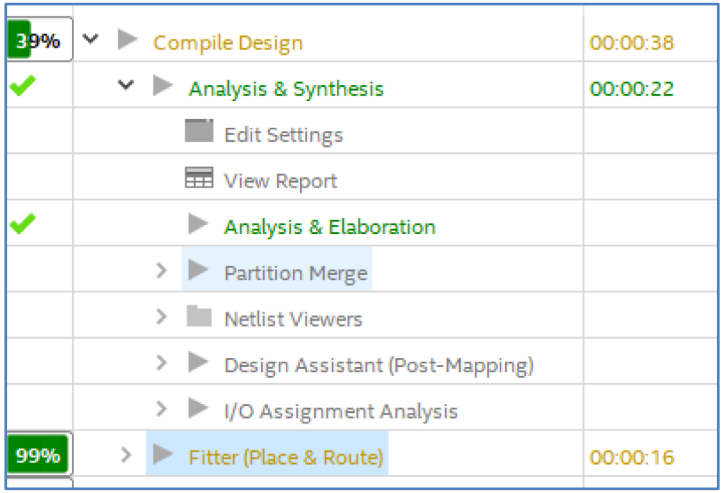
Aside: Modeling time in cyber physical systems is a deep, complex area of work

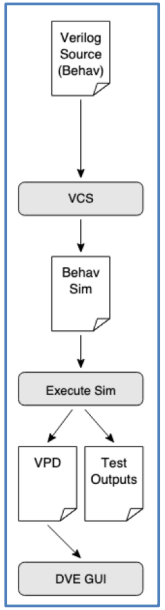
- Local experts: Rajesh and Tajana, kinda-sorta-maybe Pat

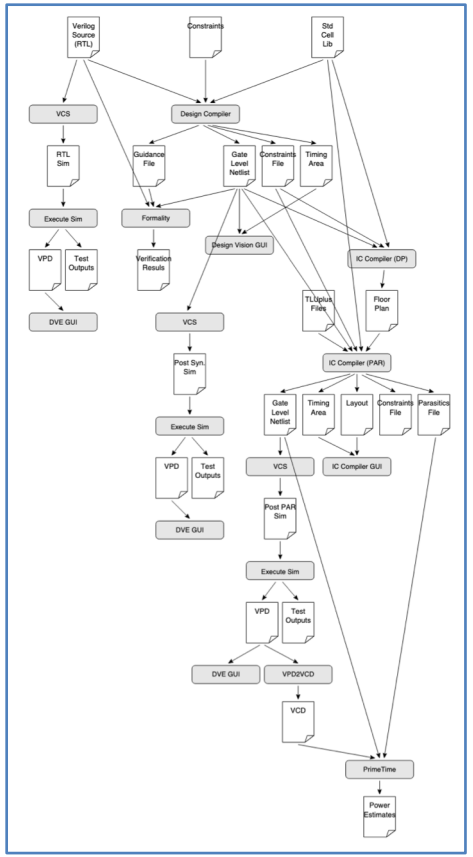
- What is time and how do we represent it?
  - Edward Lee at UC Berkeley

Synthesis adds all the details

- Comparison of the tool flow circa 2009 [from UCB CS250 FA09]

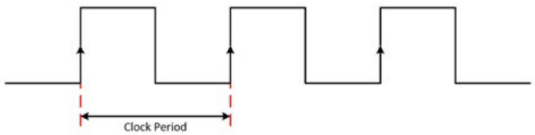




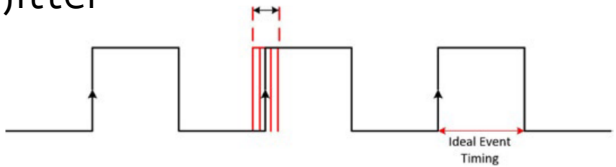


Going from a simulation clock to a real, hardware clock:

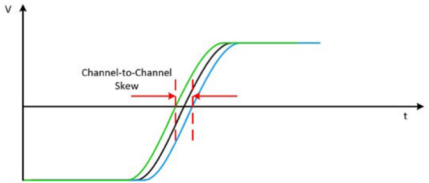
Ideal Clock



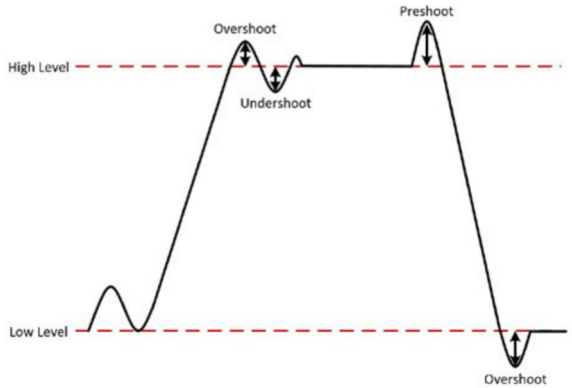
Jitter



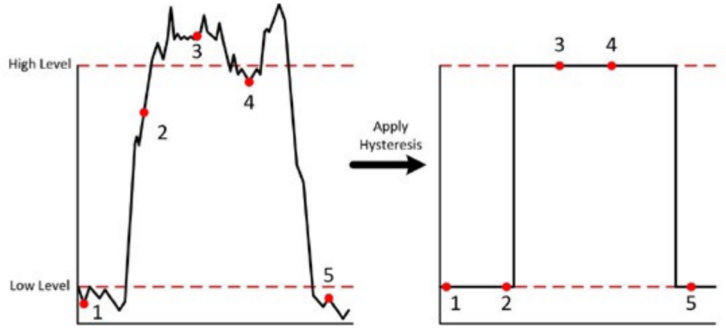
Clock trees -> slew



Clock Aberrations



1 is really 0.8, hysteresis gets us back

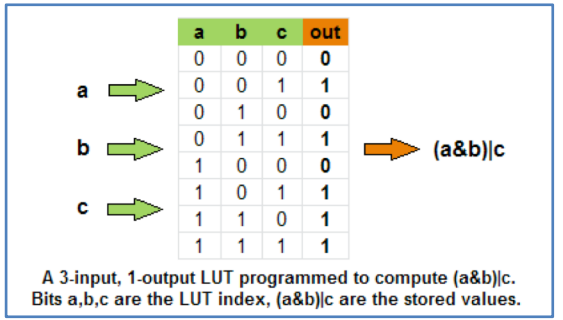


Graphics from https://download.ni.com/evaluation/pxi/Digital_Timing.pdf

Synthesis generates *netlists*

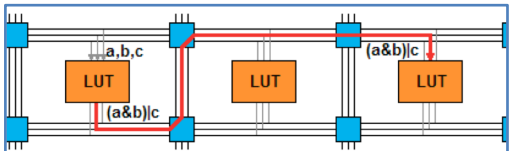
- Netlists express hardware out of basic building blocks
  - Could be literally descriptions of transistors
  - For custom chips, commonly "standard cells" (i.e. a DFF, an OR gate, etc)
    - Usually (under NDA) from your fab
  - For FPGAs, it's mostly LUTs and connections
    - Great blog post that digs into details:

      https://yosefk.com/blog/how-fpgas-work-and-why-youll-buy-one.html

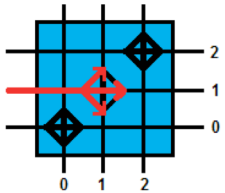


a	b	c	out
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

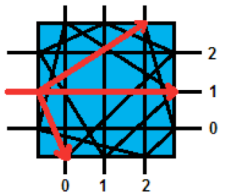
A 3-input, 1-output LUT programmed to compute (a&b)|c.
Bits a,b,c are the LUT index, (a&b)|c are the stored values.



Routing (a&b)|c from the LUT computing it to another through switch boxes



Disjoint switch box: signal from track 1 stays on track 1



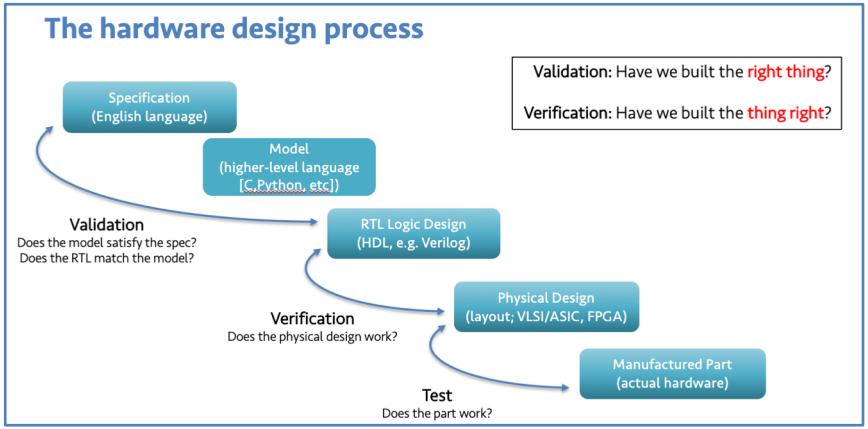
Wilton switch box: topology with track changes

Netlists are very technology-specific

- Both to the underlying synthesized hardware and the EDA toolchain
  - n.b. some netlists are 'just Verilog', but even then metadata does funny stuff

- What's significant to us is that we can *simulate synthesized netlists*
  - *Better* predict if HW will work
  - *Still* just simulation!

  - CSE148: Fast cores on FPGAs



Why does this technology-specificity matter for 141L?

- Designs must be synthesizable
- And performance must be on a level playing field

> in the block diagram file. Everyone will use Questa/ModelSim for simulation and Intel (formerly Altera) Quartus II for logic synthesis in the Cyclone IVE family, device EP4CE40F29C6.

(n.b. lecture switched to live demo after this slide)

Need to tell the tools about your clock — TopLevel.sdc

```
# Mind the filename! This must match your top level module name.


# The `create_clock` command defines a clock for the system
#
# There are _a ton_ of additional options for clocks to
# capture skew, jitter, distribution, etc; these go beyond
# the scope of this class. Your designs will probably want
# to modify the period (to go faster!), but nothing else.
create_clock -period 20.00 -name main_clock Clk


# This will automatically configure setup and hold time throughout your
# design, as opposed to you setting uncertainties explicitly. Setting up
# uncertainty manually is beyond the scope of this class, however timing
# analysis does require that uncertainty be set, so we let the tool do it.
derive_clock_uncertainty
```

The tools stop if meet your timing, only push if they have to

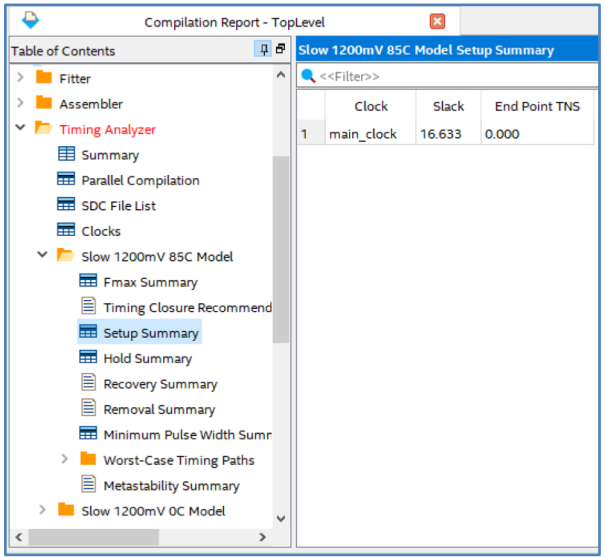
Timing Closure Recommendations

Summary

This design does not contain any failing setup paths. The worst-case slack is 16.633 ns.

Top Failing Paths

No paths fail setup timing.



What if we set clock to 20-16.633 ~= 3.0?

Timing Closure Recommendations

Summary

This design does not contain any failing setup paths. The worst-case slack is 0.447 ns.

Top Failing Paths

No paths fail setup timing.

Can we push to 3.0 – 0.447 ~= 2.0??

	Clock	Slack	End Point TNS
1	main_clock	-0.544	-2

Timing Closure Recommendations

Summary

This design contains failing setup paths with a worst-case slack of -0.544 ns. Run Report Timing Closure Recommendations for recommendations on how to close setup timing. For recommendations for any particu path, click the appropriate link in the table below.

Top Failing Paths

	Slack	From	To	Recommendations
1	-0.544	ProgCtr:PC1|ProgCtr[1]	ProgCtr:PC1|ProgCtr[9]	Report recommendations for this path
2	-0.524	ProgCtr:PC1|ProgCtr[0]	ProgCtr:PC1|ProgCtr[9]	Report recommendations for this path
3	-0.445	ProgCtr:PC1|ProgCtr[2]	ProgCtr:PC1|ProgCtr[9]	Report recommendations for this path