Introduction to FPGA Design using Quartus (Remote Console)

Intel Programmable Solutions Group

Larry Landis
Senior Manager University Outreach

- Larry Landis – University Outreach Senior Manager, Intel Programmable Solutions Group
- 36 years in electronics industry – design, sales, marketing, training (Altera 2004)
- Adjunct Lecturer for Digital Electronics Course – Santa Clara University

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

BSEE

LMU | LA
Loyola Marymount University

MSEE

SANTA CLARA BRONCOS

Adjunct Lecturer

Best Practices for Webex Training

- Mute your microphone unless you are speaking
- Be aware if your audio is in the main or your lab room
  - When you click YES in the audio question – your audio is directed to your own breakout room. The host and TAs will visit you there.
- Watch the chat window for important information, and send a chat to all with an inquiry if relevant to all – you need to exit your PC lab breakout room
- Don't be shy … Ask lots of questions!

Topics

- FPGAs at Intel
- Fundamentals of Digital Electronics
- FPGA Architecture
- Intel® Quartus® Prime Design Software
- FPGA Design Flow

Intel® FPGA Academic Ecosystem

- Train the next generation of FPGA Designers
- Increase Intel® FPGA presence in academia

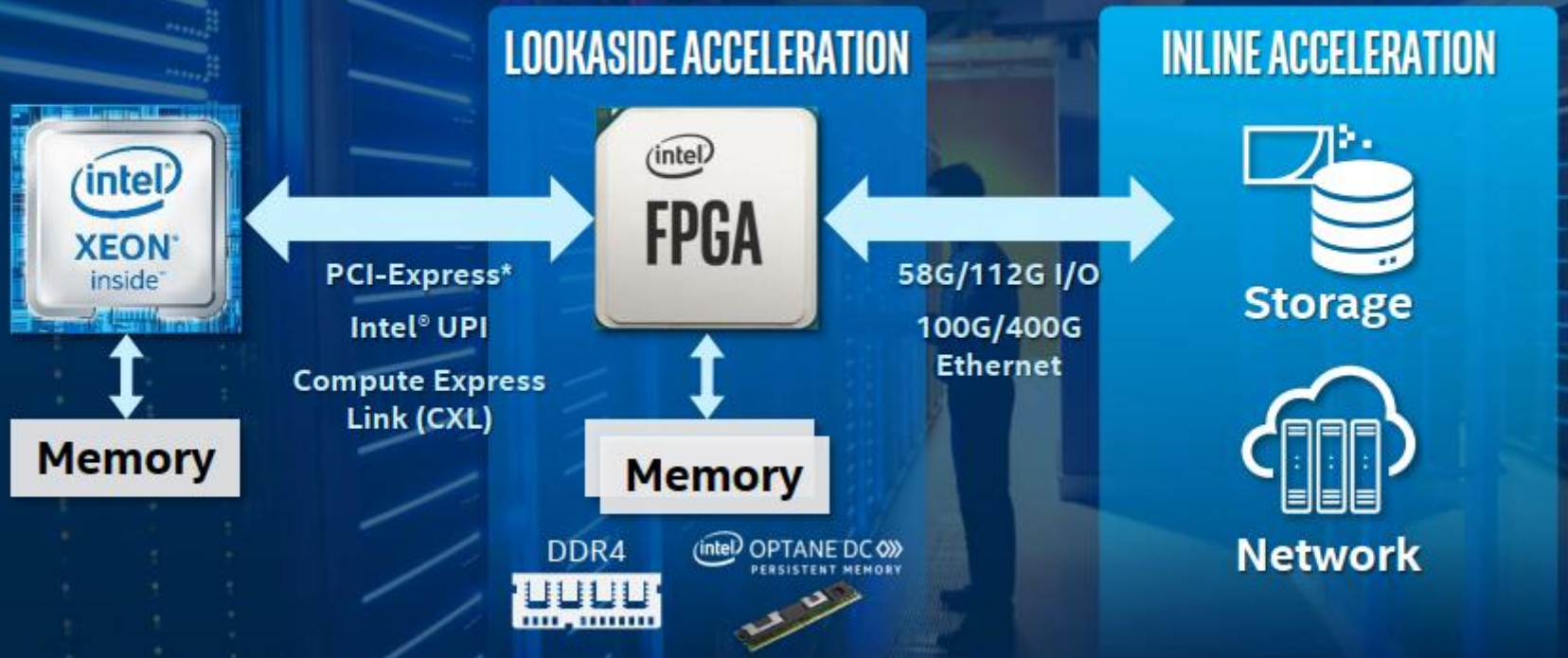
- Academic access to the latest generation of Intel FPGAs

- Nurture the talent pipeline for Intel and our customers
- Engage research on Intel FPGAs

INTEL® FPGA EE COURSEWORK OFFERINGS

- Undergraduate
  - Digital Logic
  - Digital Systems
  - Computer Organization
  - Embedded Systems

INTEL® FPGAS ACCELERATING THE CLOUD & ENTERPRISE



Website Access for the Intel® FPGA Devcloud

https://software.intel.com/en-us/devcloud/FPGA

intel

Field Programmable Gate Array (FPGA)

- Flexible, multi-functional reprogrammable silicon
- Custom hardware functionality
- Bare-metal speed and reliability
- Truly parallel in nature

Benefits of FPGA Technology

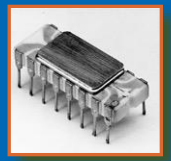
- Flexibility
- Time to market
- Performance
- Reliability
- Long-Term Maintenance – reprogram if features change or bugs found
- Many different applications – 5G, Data Center, Industrial, DSP
- Excellent prototyping vehicle

Rise of new markets

Intel's history with FPGAs

1968 — Intel is founded by Robert Noyce and Gordon Moore

1971 — World's first microprocessor

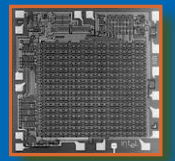


1983 — Founded

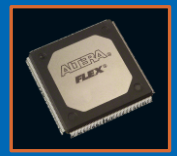
1984 — First PLD

1984 — Intel and Altera establish a joint marketing agreement

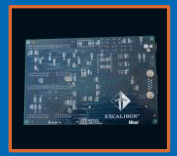
1994 — Altera purchases Intel's PLD business



1995 — First embedded RAM



2000 — First soft micro-processor



2002 — First embedded DSP

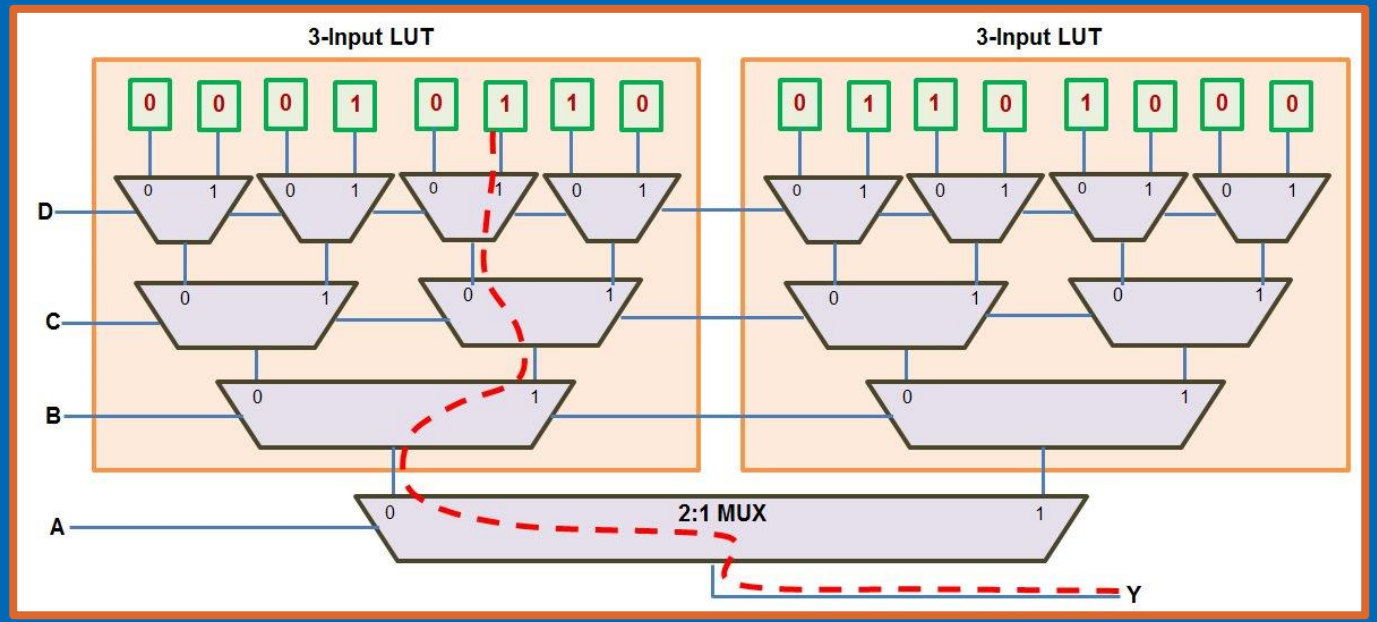
2012 — SoC FPGA

2013 — Intel and Altera start foundry relationship for 14nm FPGAs and SoCs

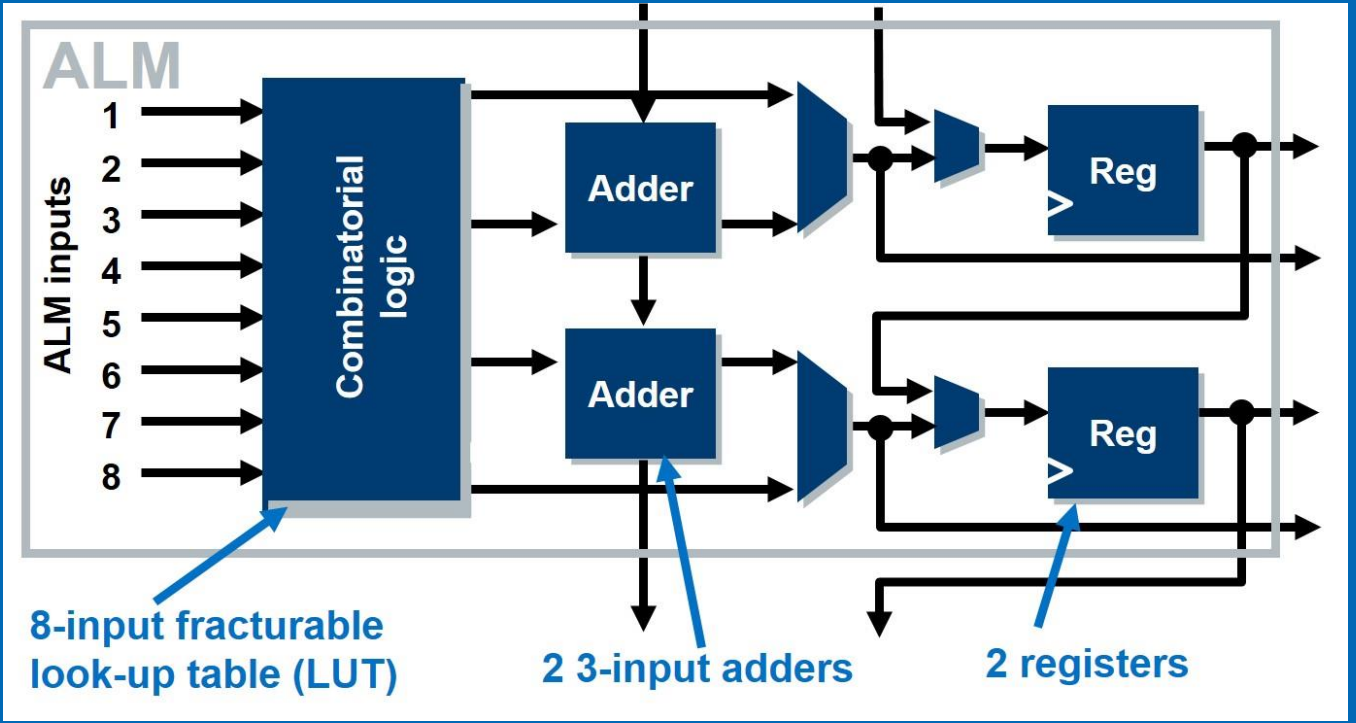
2016 — System-in-Package (SIP)

2015 — Intel acquires Altera

Look-Up Table (LUT): the foundation



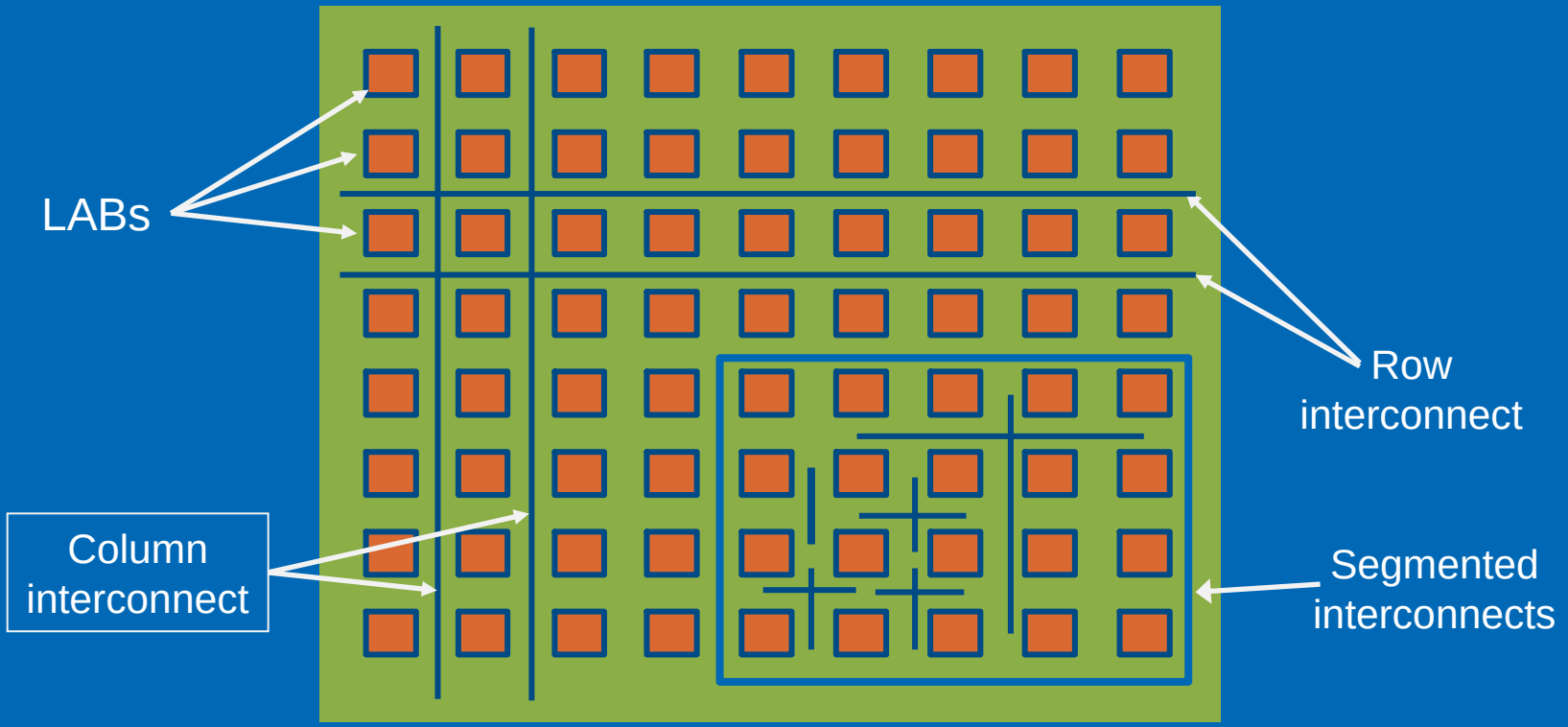
Logic Array Blocks



x 10

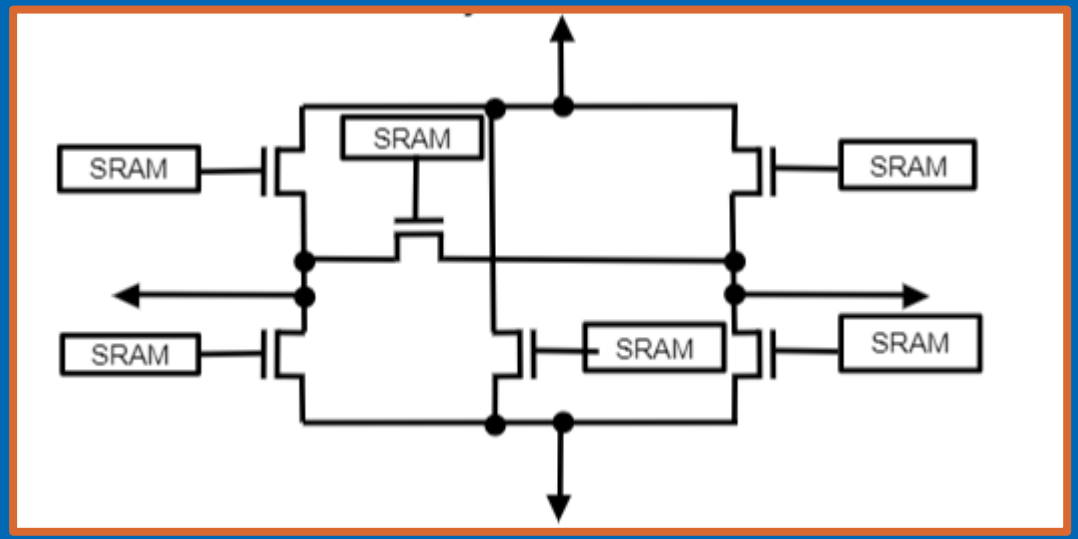
*Number of inputs and ALMs per LAB vary by product family

Building the Array



How switching fabric is programmed

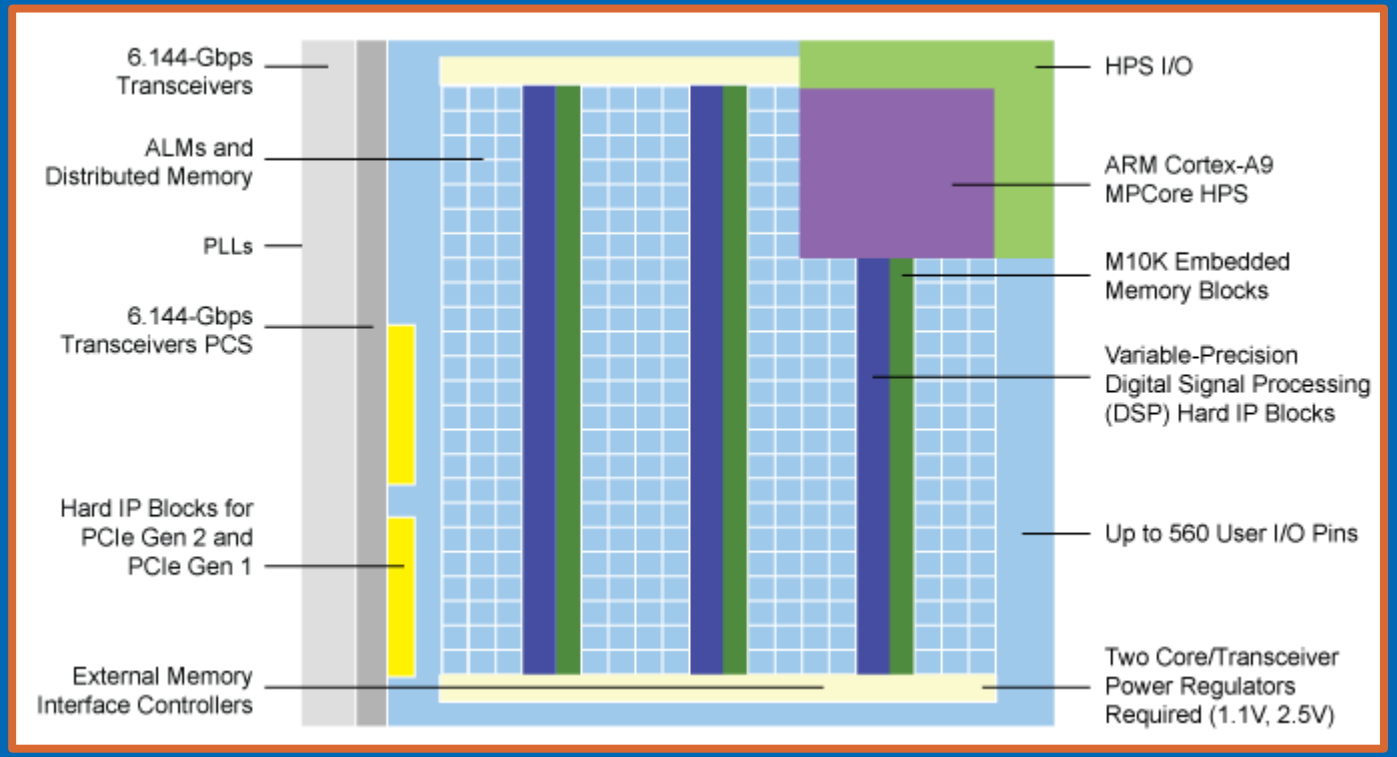
Row/Column Interconnect Junction



Programming info stored in a external non-volatile device

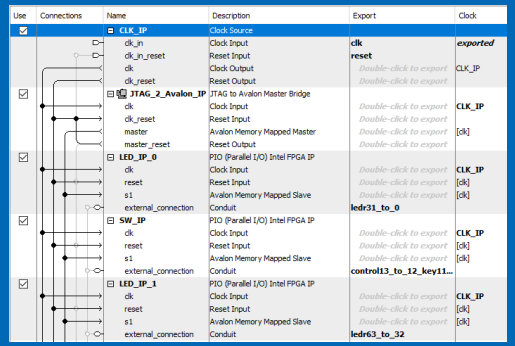
- Active: programmed automatically at power-on
- Passive: Intelligent host (CPU) controls programming

FPGAs “Hardened” features (Cyclone V)

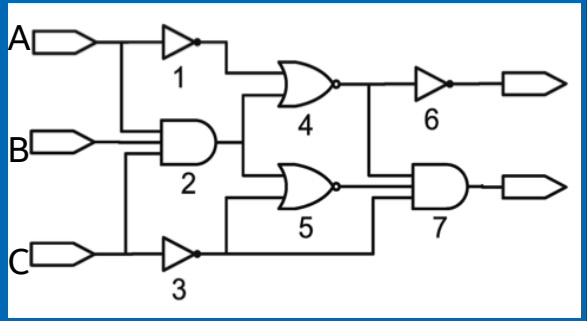


Describing FPGAs

- Schematics
- System Integration Tools – predefined blocks
- Hardware Description Languages (HDLs)
  - Verilog, VHDL are most popular
- High level languages
  - “HLS”
  - OpenCL
  - Data Parallel C++




```
try {

    sycl::queue q(sycl::default_selector{});

    const float A(aval);

    sycl::buffer<float,1> d_X { h_X.data(), sycl::range<1>(h_X.size()) };
    sycl::buffer<float,1> d_Y { h_Y.data(), sycl::range<1>(h_Y.size()) };
    sycl::buffer<float,1> d_Z { h_Z.data(), sycl::range<1>(h_Z.size()) };

    q.submit([&](sycl::handler& h) {

        auto X = d_X.template get_access<sycl::access::mode::read>(h);
        auto Y = d_Y.template get_access<sycl::access::mode::read>(h);
        auto Z = d_Z.template get_access<sycl::access::mode::read_write>(h);

        h.parallel_for<class nstream>( sycl::range<1>{length}, [=] (sycl::id<1> it) {
            const int i = it[0];
            Z[i] += A * X[i] + Y[i];
        });
      });
      q.wait();
}
```

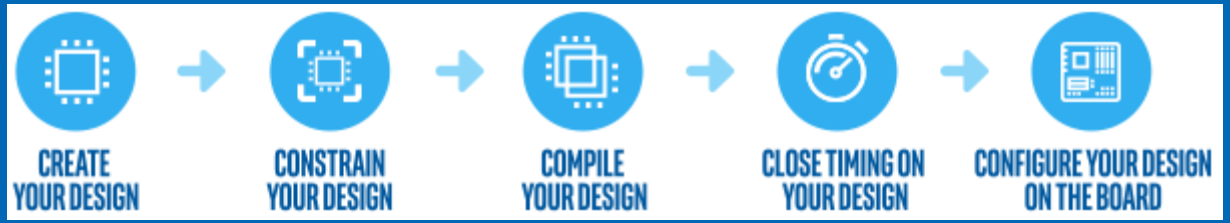


What is IP (Intellectual Property)

- Complex functions that Intel designs for our customers so they don’t have to design it themselves
  - Sometimes IP is free
  - The more complex stuff costs since its expensive to develop and make sure it works
- Examples: Ethernet Controller, PCIe Controller, soft processor, multiplier functions, etc.

Basics of Quartus

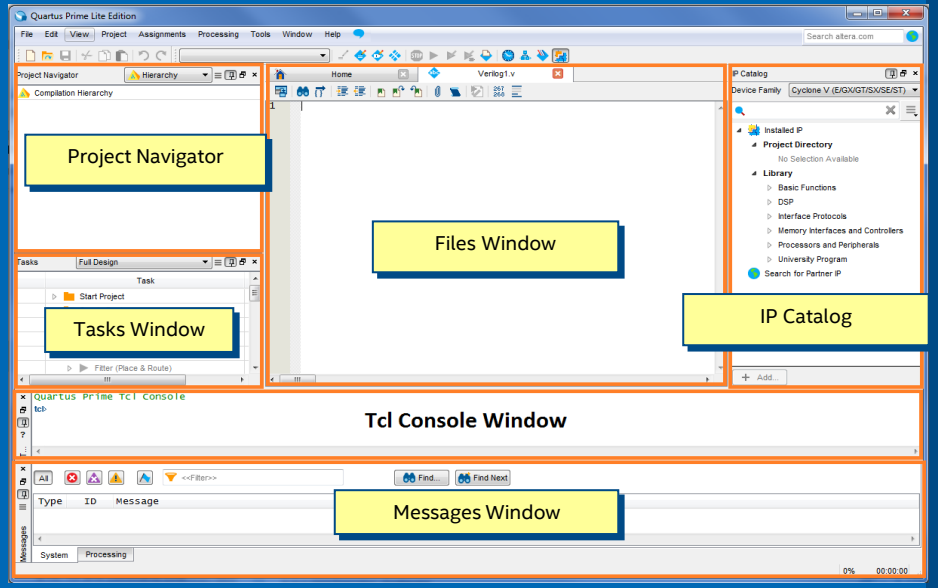
- Intel® Quartus® Prime Design Software is a tool for FPGA, SOC and CPLD design
- Includes synthesis, debug, optimization, verification and simulation
- Takes a description of an FPGA (schematic or HDL) and determines how the lookup tables are programmed



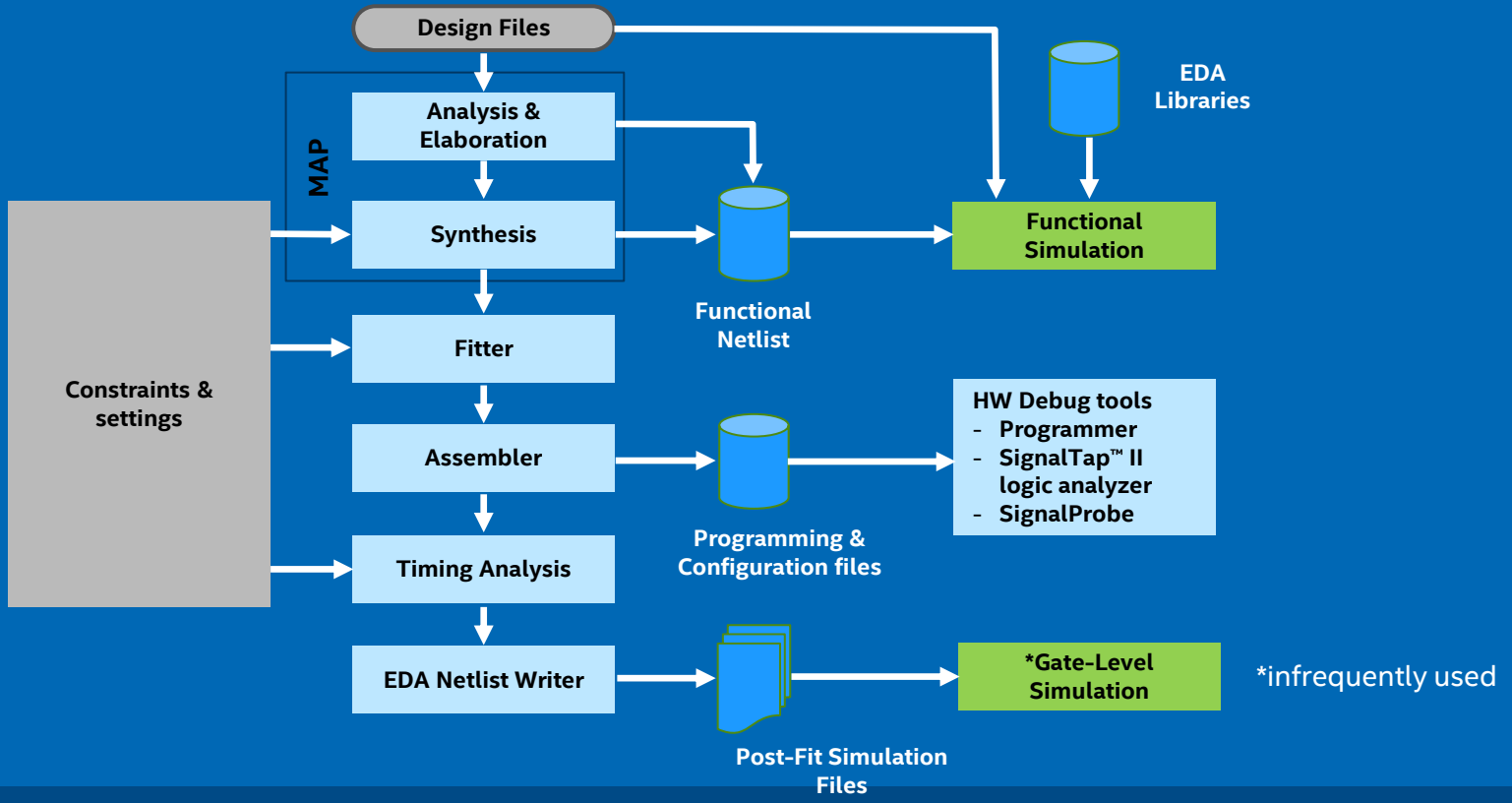
- Many formats to program an FPGA
  - In this class we will use a ".sof" file (SRAM object file)
  - The .sof file is "volatile" and needs to be reprogrammed every time the board is restarted

Quartus User Interface

- Quartus Prime Software Main Window
  - *Project Navigator* shows your project hierarchy, source files, design units, IP and design revisions in your project.
  - *Tasks* window shows the status of the design and can be used to run or re-run parts of the design flow
  - *Messages* window outputs messages from each process of the run.
  - *Files* window has tabs for the report browser, open design files and any other file opened by the user.
  - *IP Catalog* window is open by default and is used to generate IP functions that are to be used in your design.

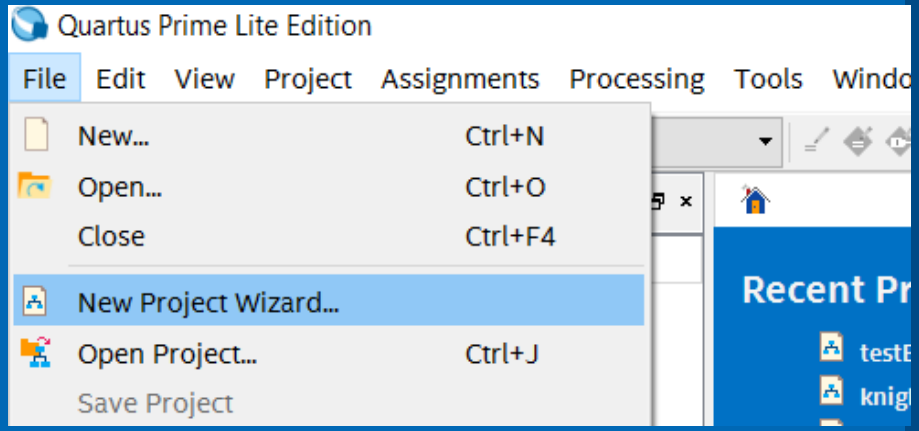


Tools Flow


Design Files
MAP
Analysis & Elaboration
Synthesis
Fitter
Assembler
Timing Analysis
EDA Netlist Writer
Constraints & settings
EDA Libraries
Functional Simulation
Functional Netlist
HW Debug tools
- Programmer
- SignalTap™ II logic analyzer
- SignalProbe
Programming & Configuration files
*Gate-Level Simulation
Post-Fit Simulation Files
*infrequently used

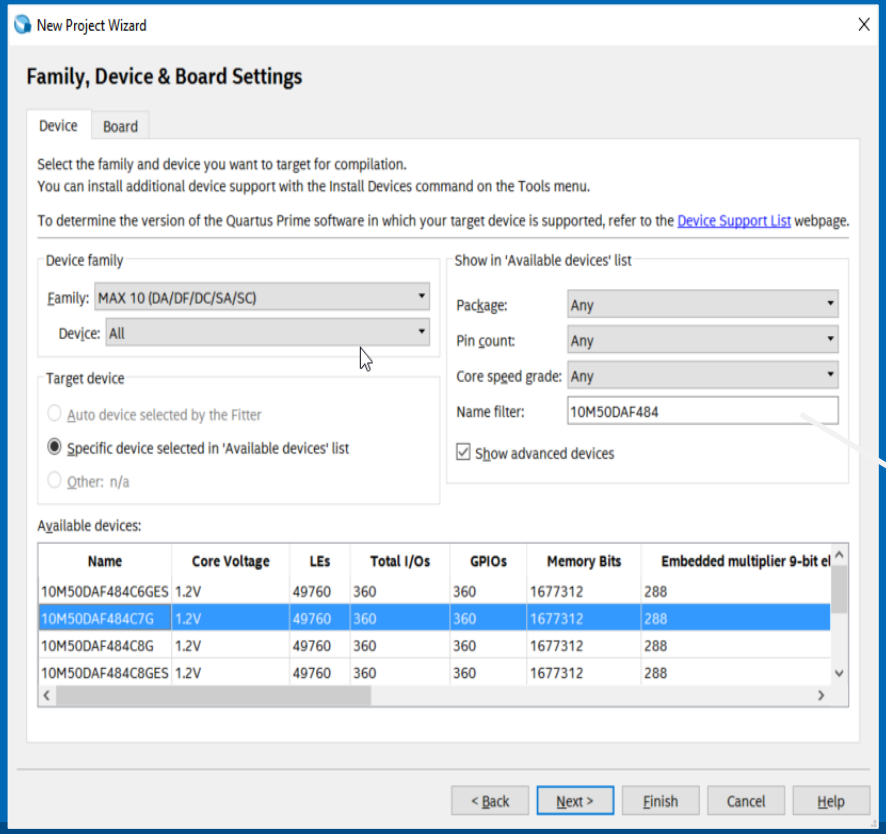
New Project Wizard

1. Name project
2. Set Working Directory & Top-Level Entity
3. Add source files
4. Select Device
5. EDA tool settings



All settings can be modified later. Some steps can be skipped.
The top level entity must match the top level module in your design exactly (case sensitive) in order to avoid a compile error.

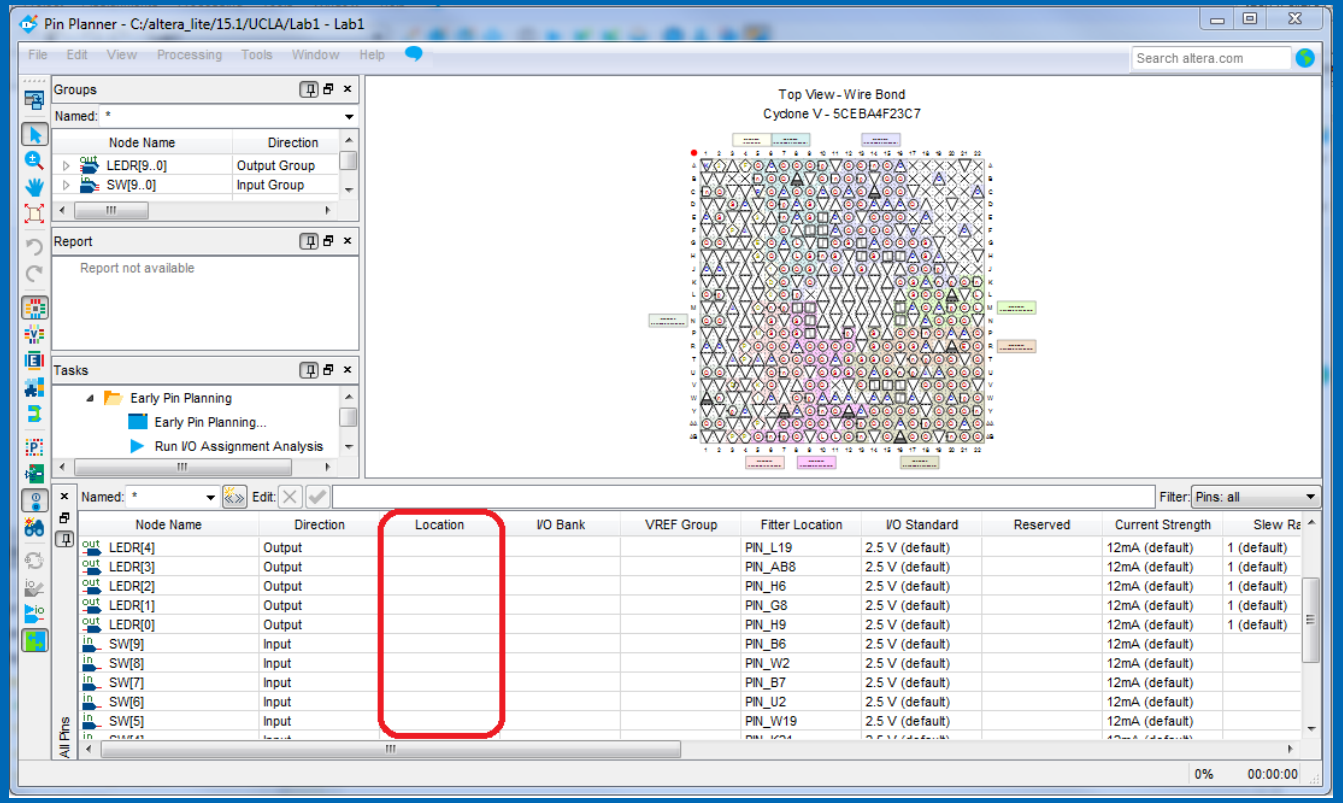
Family & Device Settings



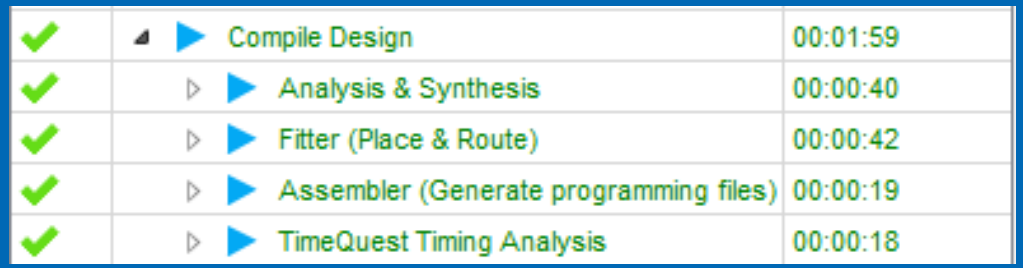
Expand the window so you can see all the fields

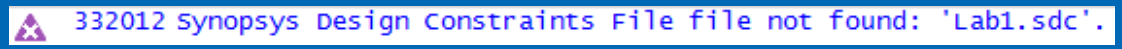
Get the part number for your specific device by looking on the chip on your board or the side of the box.

Pin Planner


Pin Planner - C:/altera_lite/15.1/UCLA/Lab1 - Lab1
File Edit View Processing Tools Window Help
Search altera.com
Groups
Named: *
Node Name Direction
LEDR[9..0] Output Group
SW[9..0] Input Group
Report
Report not available
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Top View - Wire Bond
Cyclone V - 5CEBA4F23C7
Named: * Edit: Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard Reserved Current Strength Slew Ra
LEDR[4] Output PIN_L19 2.5 V (default) 12mA (default) 1 (default)
LEDR[3] Output PIN_AB8 2.5 V (default) 12mA (default) 1 (default)
LEDR[2] Output PIN_H6 2.5 V (default) 12mA (default) 1 (default)
LEDR[1] Output PIN_G8 2.5 V (default) 12mA (default) 1 (default)
LEDR[0] Output PIN_H9 2.5 V (default) 12mA (default) 1 (default)
SW[9] Input PIN_B6 2.5 V (default) 12mA (default)
SW[8] Input PIN_W2 2.5 V (default) 12mA (default)
SW[7] Input PIN_B7 2.5 V (default) 12mA (default)
SW[6] Input PIN_U2 2.5 V (default) 12mA (default)
SW[5] Input PIN_W19 2.5 V (default) 12mA (default)
All Pins
0% 00:00:00

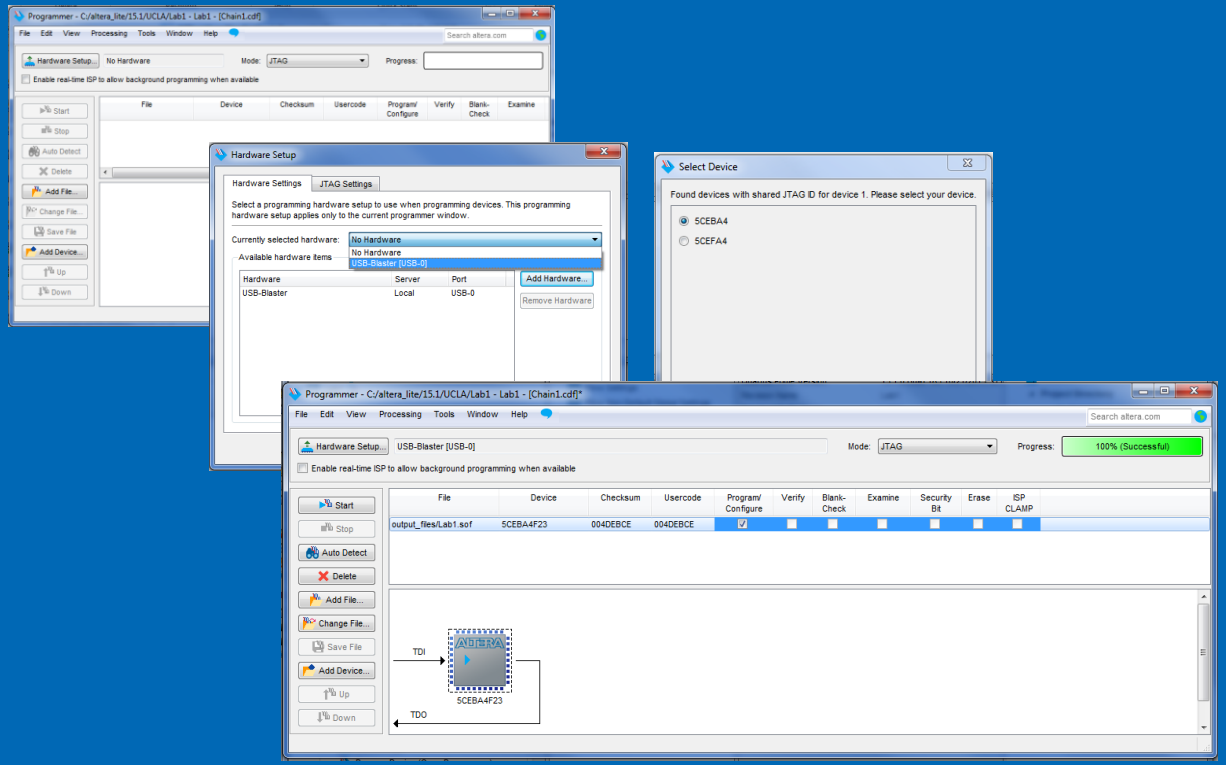
Compile your design



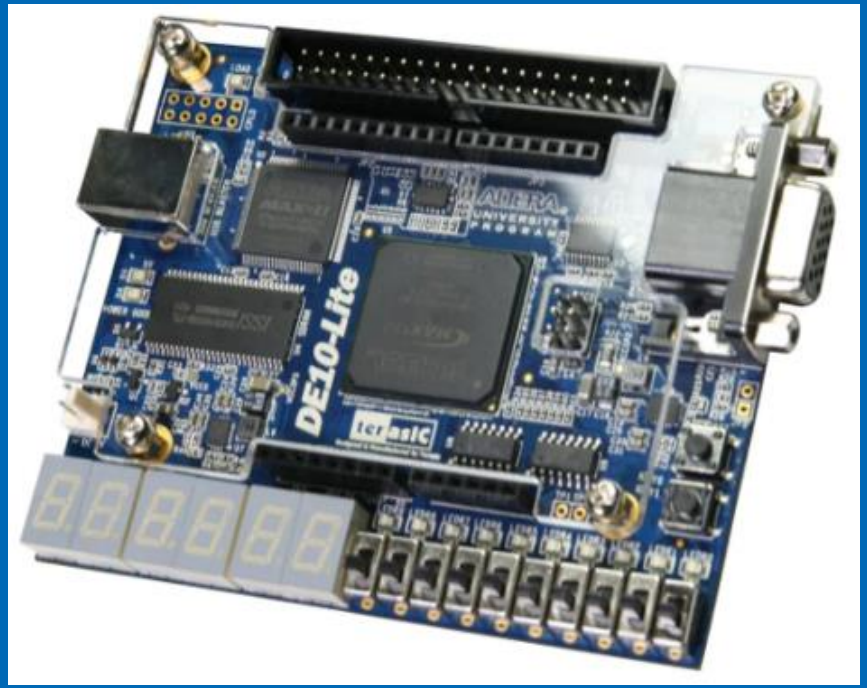


Warnings shown in blue won't prevent your design from compiling or being programmed, but they could indicate possible bugs. This lab does not have any design constraints, so the .sdc file is not needed. You will learn how to create one in the timing analysis workshop.

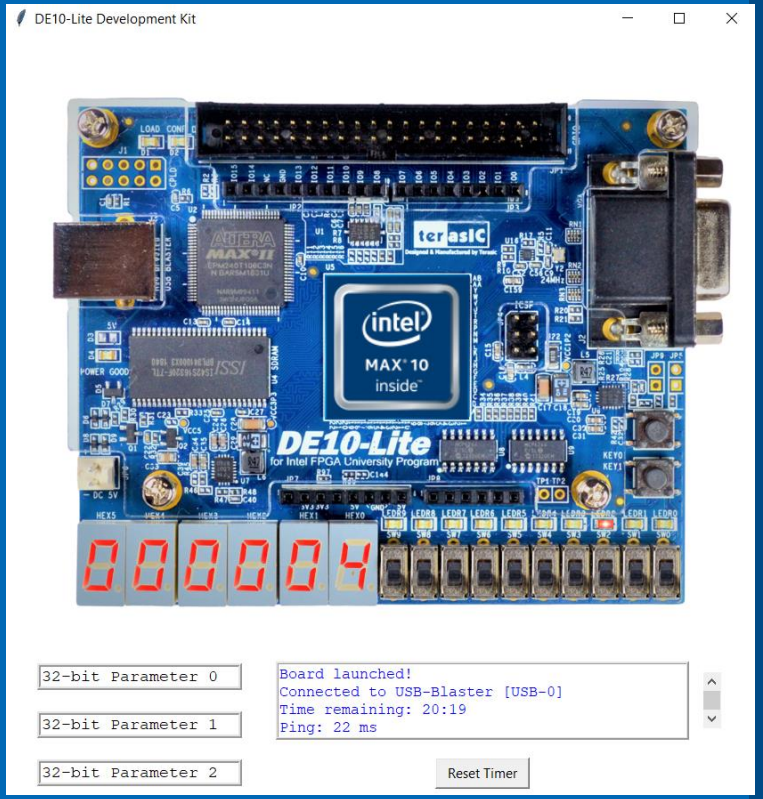
Program your FPGA



Test your design



Live Hardware



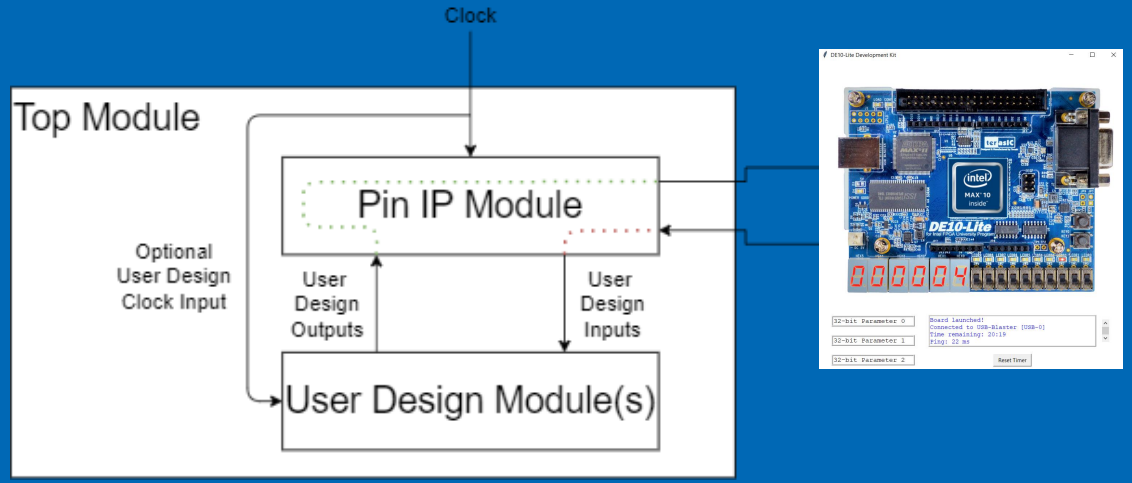
Remote Console

Covid-19: New challenges for remote learners

- In June 2020, we launched initiatives to help remote learners
- Addresses situation where student does not have a physical board in their possession
- Simulation based only requires Modelsim
  - Quick compiles
  - More setup with IP
  - Slower clock speed
- Hardware based requires Intel Quartus Prime-Lite
  - Longer compiles, but hardware accurate
  - More complex networking, adds a level of hierarchy in the design

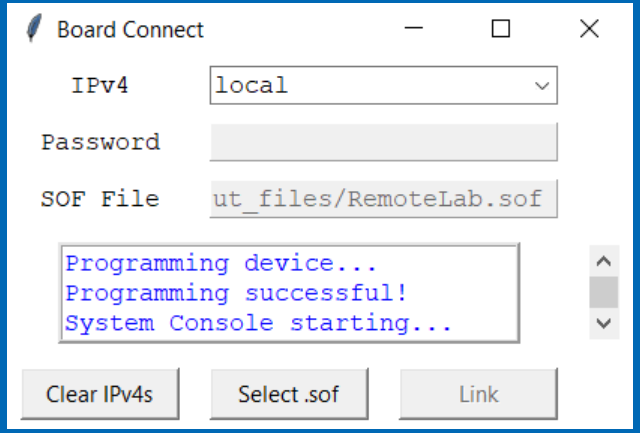
Remote console used in today's lab

- Support for remote GPIO-based labs.
  - Good for undergraduate curriculum.
- Leverages existing Quartus Prime Lite installation.
  - No additional software needed.
  - Up and running in minutes.
- Requirements:
  - Windows machine
  - Quartus Prime Lite
  - Access to local or remote development kit



Connecting to Development Kit

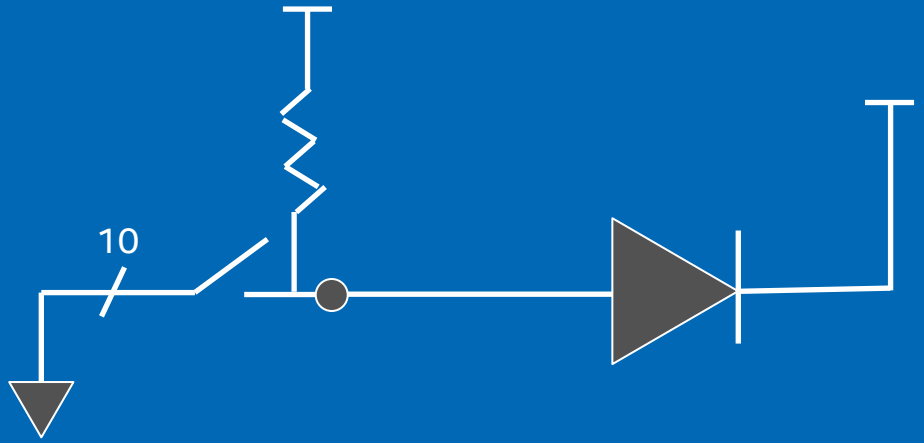
- Right-click launch task
- Select "Start"
- If development kit is locally connected keep IPv4 as "local"
  - Otherwise a host's IPv4 and password are used.
- Select project .sof.
- Link



New Remote Console Projects

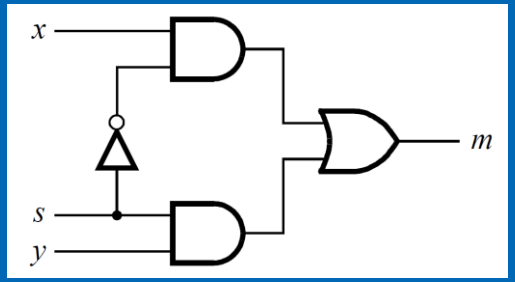
- Download the blankProject.qar example project for your dev kit.
- Unarchive the project into Quartus.
- Add source files and connect new design to top-level wrapper.
- Debug and compile.
- Launch Remote Console and validate functionality in hardware.

First Lab: Switch to LED



assign LED = SW;

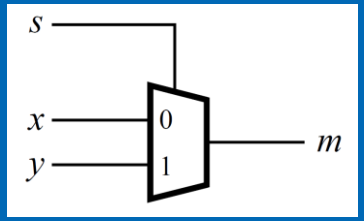
Next lab: Multiplexer



Circuit

s	m
0	x
1	y

Truth table

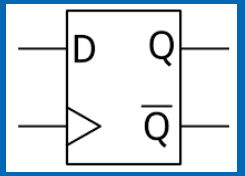


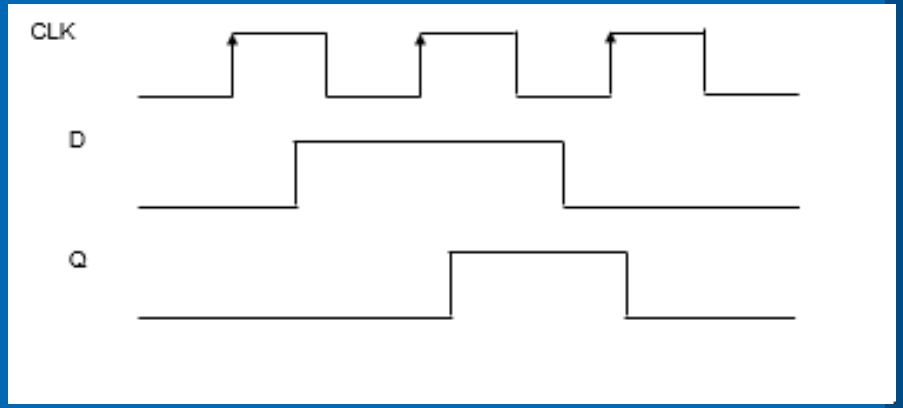
Symbol

The multiplexer can be described by the following Verilog statement:

```
assign m = (~s & x) | (s & y);
```

Knight Rider





Webex Cheat Sheet

- When using Webex training center, select a PC (see lower right corner). Your name is placed below the PC name. Select your own PC, not one that is already occupied.
  - If you select a PC that has a name below it, you can observe what others are doing, ok to do if agreed upon.
- **Audio connectivity has a mode where if you select "YES" when you connect to a machine, you have your own audio breakout room when you are working on your PC. Others can privately talk within your room but you lose audio connectivity to the main session. "NO" keeps your audio feed in the main session and you will hear background chatter.**
- When you leave your PC session, you can return to your PC in it's same state. Leave your PC session to see chats within the main session or listen to the main audio feed.
- Send a chat message in the main session if you are having problems and some one will join your session within the breakout room audio feed. When you rejoin your PC session select YES so you have the audio feed in your breakout room.
- Open up chat within your breakout room, that way the host/TA can communicate that way with you. Look for broadcast messages to all students. Periodically check chat in main session and your breakout session.
- If you are TA'ing, remind the host to make you a panelist so you can see the full chat feed.

Quartus and Design Tips

- When Quartus Prime Lite first starts for the very first time it might ask you about purchasing a license, select Run Quartus, all licenses are free for this lab.
- If things fail to compile, check your top Level Entity Setting → Setting → Top Level Entity and make sure that the module <design> matches your top level entity, including case.
- Check the LEDR[0] and LEDR[9] pins carefully in the Knight Rider lab and see if they sequence properly. If not, study the code carefully!
- Sometimes copy and paste from files into Quartus has carriage return formatting errors. If you see run on lines with no carriage return, you need to copy things over line by line, or add the appropriate file to your project. It's better to click on the links and download the files.

Learning more

- DE10-Lite development kit - $55 from Terasic with student discount
- http://fpgauniversity.intel.com – lots of helpful labs and tutorials
- Customer training site (just google Intel FPGA training)
  - In catalog, search on university
- Practice makes perfect!
- Thanks for attending!