



INTRODUCTION TO FPGA DESIGN IN QUARTUS (REMOTE EDITION)

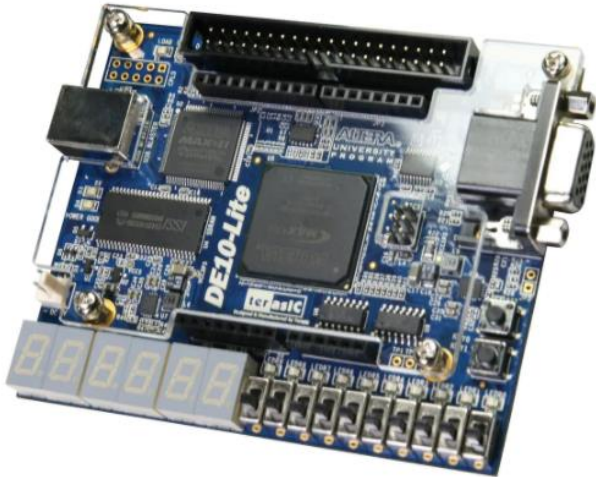
November 2020

TOPICS

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

COMPANION LAB

1. Download the Quartus Prime Lite Software:
2. We will run with a remote DE10-Lite development board



Quartus Prime Lite Edition

Release date: June, 2020
Latest Release: v20.1

Intel® Quartus® Prime
Design Software

Select edition:
Select release:

Operating System:

- ✓ The Quartus Prime Lite Edition Design Software, Version 20.1 includes functional and security updates. Users should keep their software up-to-date and follow the [technical recommendations](#) to help improve security. Additional security updates are planned and will be provided as they become available. Users should promptly install the latest version upon release.
- ✓ The Quartus Prime Lite Edition Design Software, Version 20.1 is subject to removal from the web when support for all devices in this release are available in a newer version, or all devices supported by this version are obsolete. If you would like to receive customer notifications by e-mail, please subscribe to our [customer notification mailing list](#).
- ✓ The Quartus Prime Lite Edition Design Software, Version 20.1 supports the following device families: Arria II, Cyclone 10 LP, Cyclone IV, Cyclone V, MAX II, MAX V, and MAX 10 FPGA. [More](#)

Combined Files Individual Files Additional Software

Download and Install Instructions: [More](#)

[Read Intel FPGA Software v20.1 Installation FAQ](#)

[Quick Start Guide](#)

Quartus Prime Lite Edition (Free)

Quartus Prime (Includes Nios II EDS)
Size: 1.6 GB MD5: 23479E4E46326A06C4EC63D0E6B6F192
** Nios II EDS on Windows requires Ubuntu 18.04 LTS on Windows Subsystem for Linux (WSL), which requires a manual installation.
** Nios II EDS requires you to install an Eclipse IDE manually.

ModelSim-Intel FPGA Edition (Includes Starter Edition)
Size: 1.2 GB MD5: D5E635E3229104088B0185D920ACC694

Devices

You must install device support for at least one device family to use the Quartus Prime software.

Arria II device support
Size: 499.1 MB MD5: A439BB4873E69BD23E36ACED3EA9BA9C

Cyclone IV device support
Size: 466.0 MB MD5: 14E47510CE47DC0ED92C8CB97D76488

Cyclone 10 LP device support
Size: 266.7 MB MD5: AF64DC8C96035DD084E9E8B621EB1378

Cyclone V device support
Size: 1.3 GB MD5: 2B30D36171D1BEB159971F89206E6D9F

MAX II, MAX V device support

MAX 10 FPGA device support
Size: 285.4 MB MD5: 10284D7121A02F6DD20C8C41C9ECECB

Note: The Quartus Prime software is a distributed software product. Depending on your download speed, download times may be lengthy.

FIELD PROGRAMMABLE GATE ARRAY (FPGA)



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

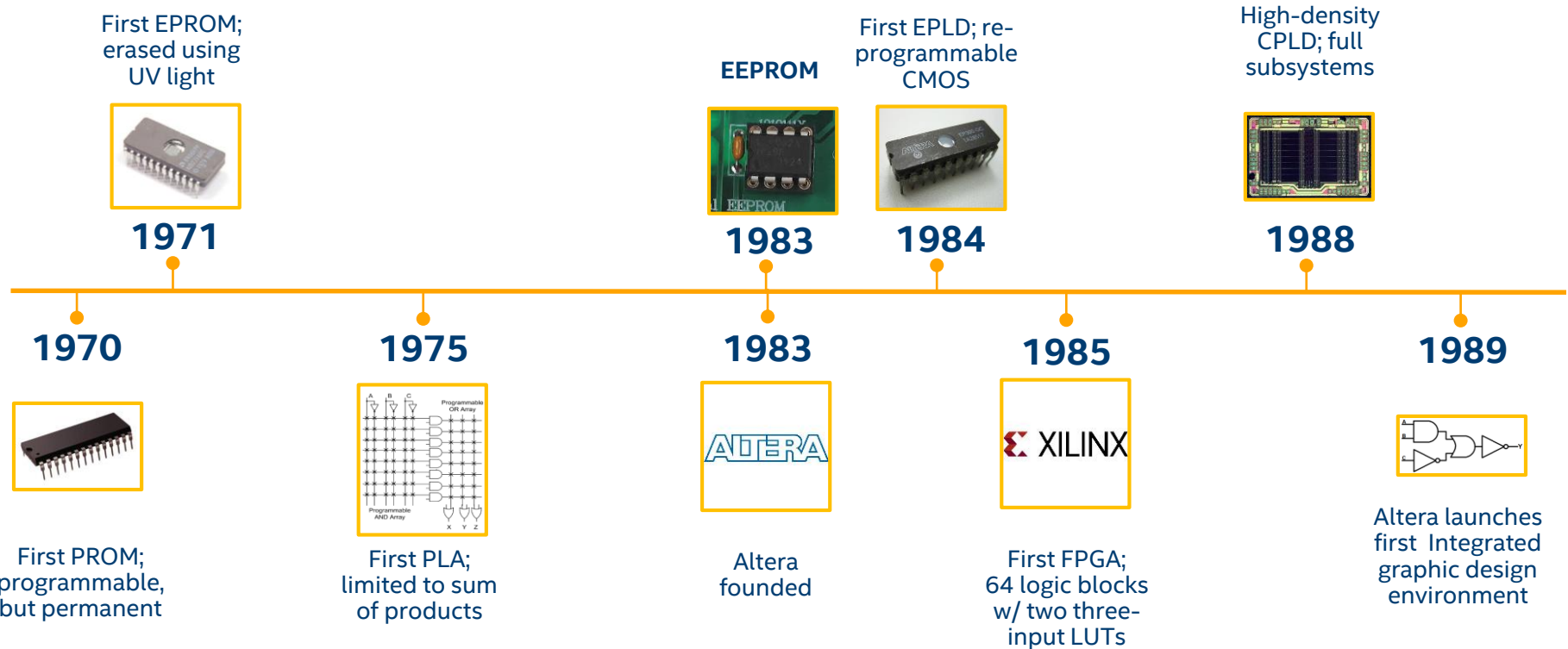
BENEFITS OF FPGA TECHNOLOGY

- 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

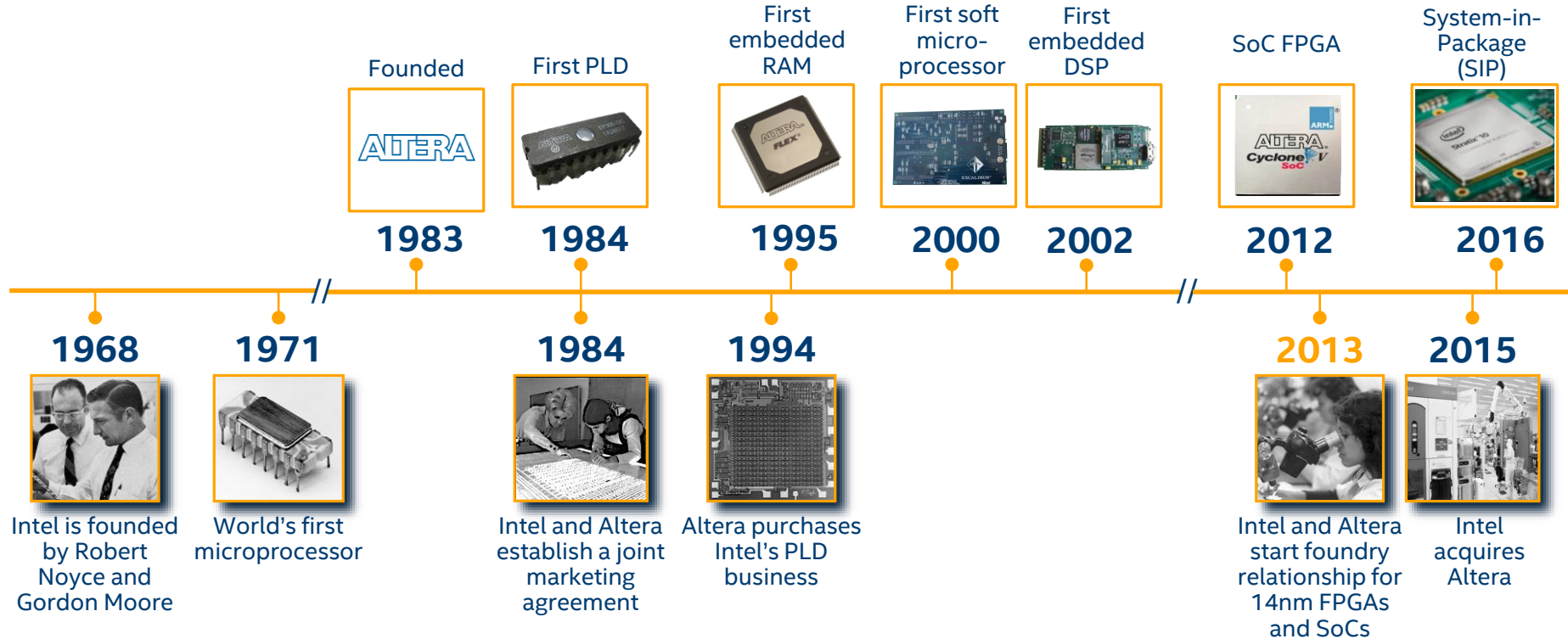
THE RISE OF NEW MARKETS



THE BIRTH OF FPGAs

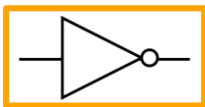


INTEL'S DEEP HISTORY WITH FPGAs



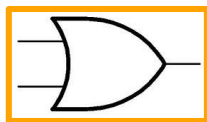
BACK TO THE BASICS

A	Z
0	1
1	0



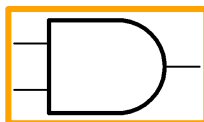
Inverter
 $Z = \sim A$

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1



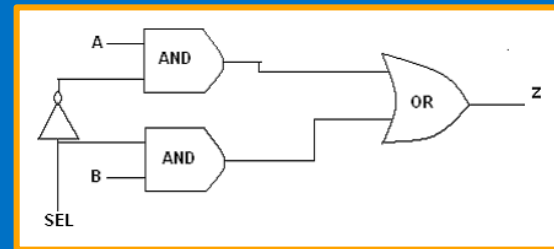
OR
 $Z = A | B$

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



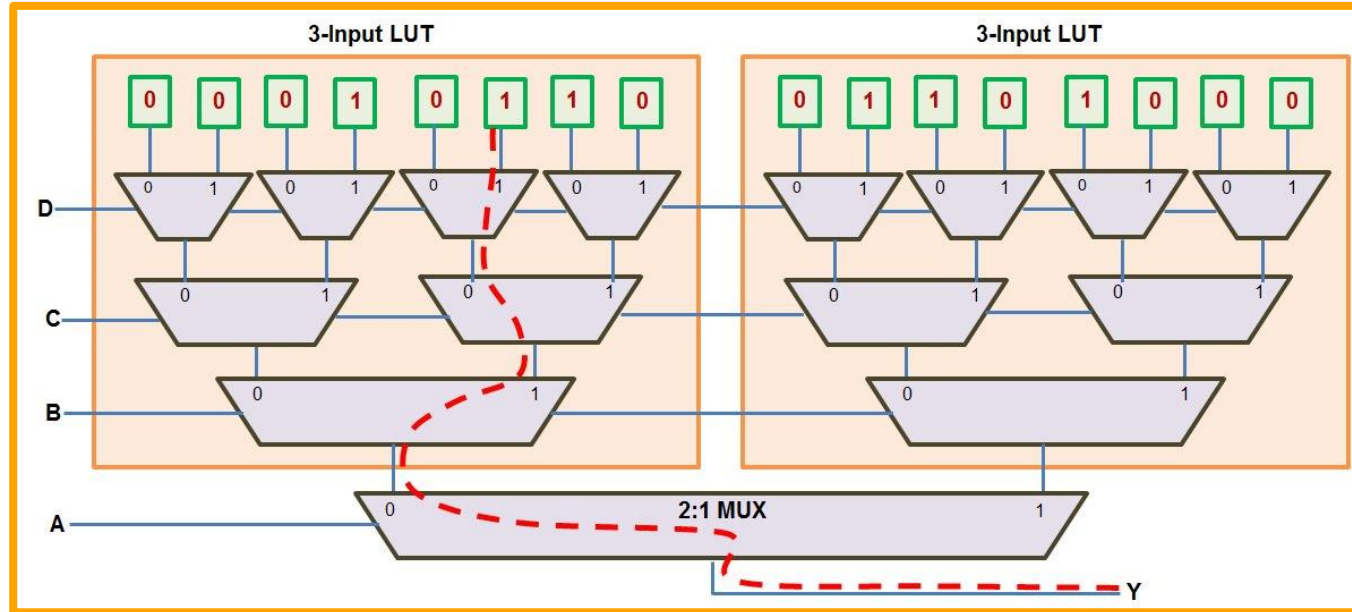
AND
 $Z = A \& B$

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

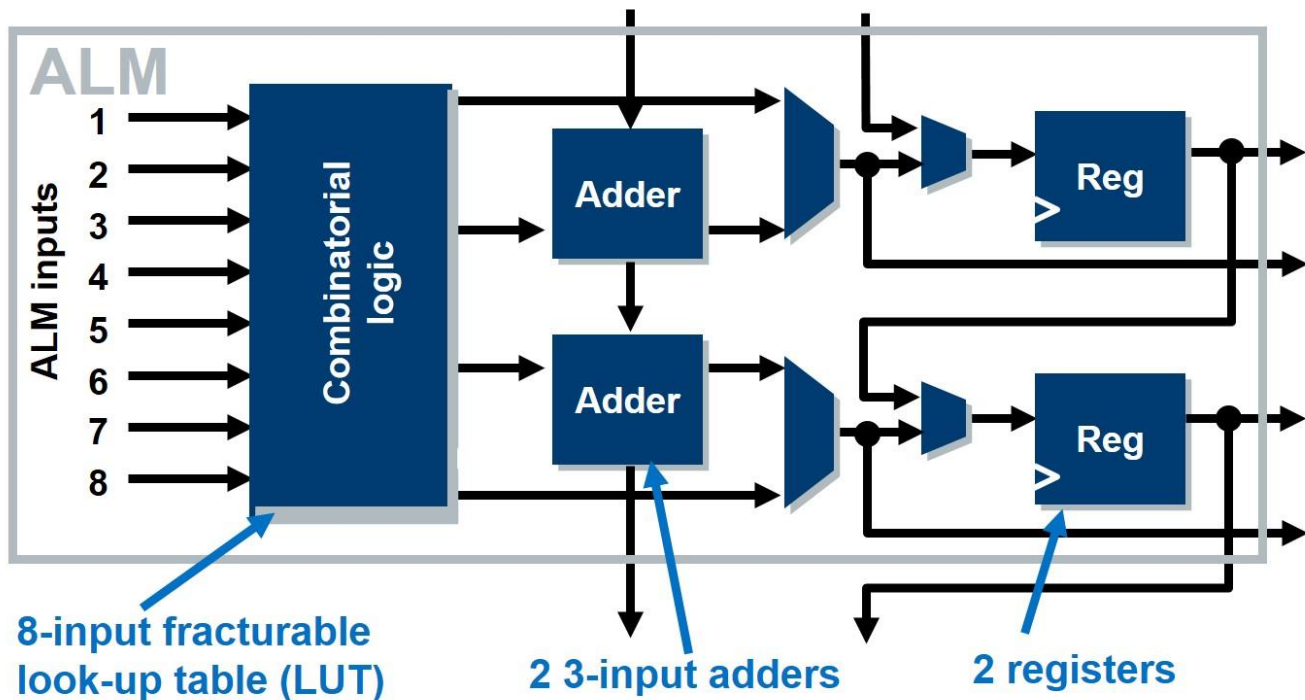


2:1 MUX
 $Z = (\sim S \& A) | (S \& B)$

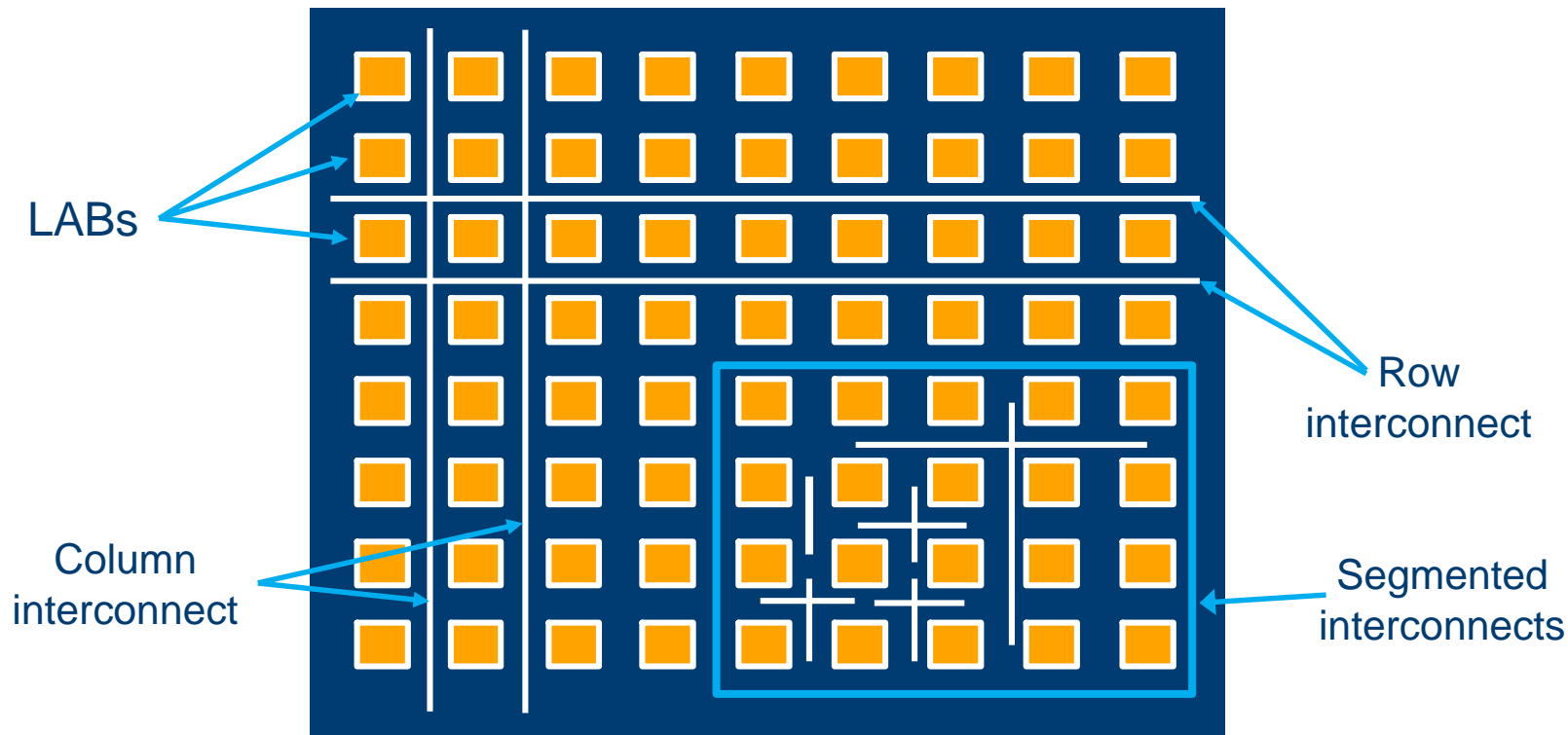
LOOK-UP TABLE (LUT): THE FOUNDATION



LOGIC ARRAY (BUILDING) BLOCKS

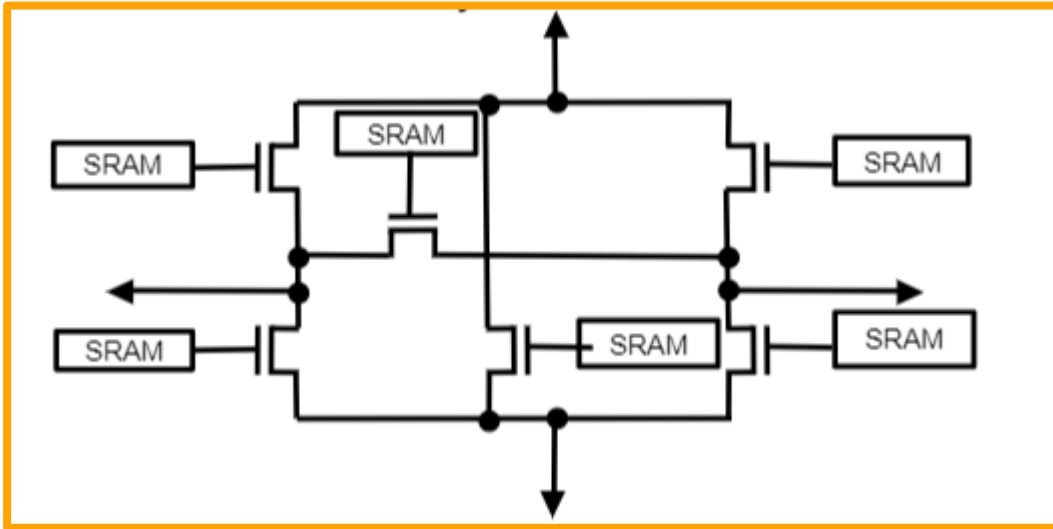


BUILDING THE ARRAY



HOW ITS PROGRAMMED

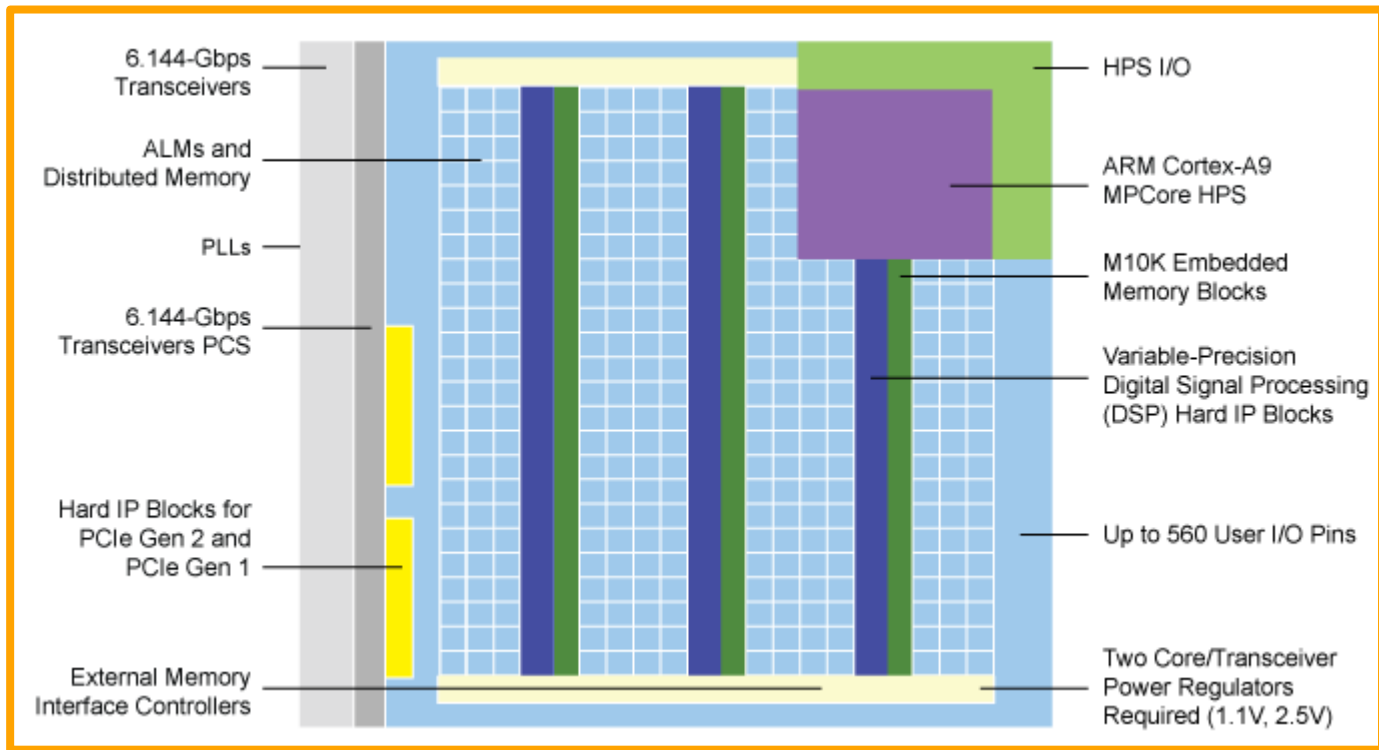
Row/Column Interconnect Junction



Programming info stored in a external non-volatile device

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

MODERN FPGAS



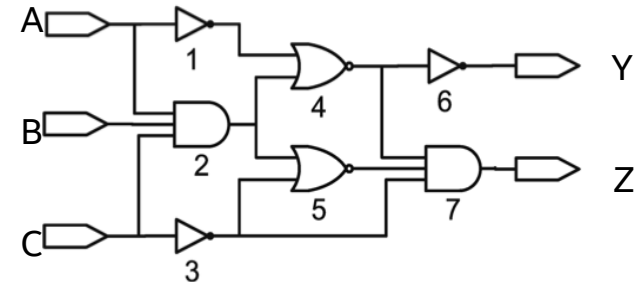
DESCRIBING FPGAS

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

Use	Connections	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		CLK_IP	Clock Source		
		clk_in	Clock Input	clk	exported
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to export	CLK_IP
		clk_reset	Reset Output	Double-click to export	
<input checked="" type="checkbox"/>		JTAG_2_Avalon_IP	JTAG to Avalon Master Bridge		
		clk	Clock Input	Double-click to export	CLK_IP
		clk_reset	Reset Input	Double-click to export	[8k]
		master	Avalon Memory Mapped Master	Double-click to export	
		master_reset	Reset Output	Double-click to export	
<input checked="" type="checkbox"/>		LED_IP_0	PSD (Parallel I/O) Intel FPGA IP		
		clk	Clock Input	Double-click to export	CLK_IP
		reset	Reset Input	Double-click to export	[8k]
		s1	Avalon Memory Mapped Slave	Double-click to export	
		external_connection	Conduit	ledr31_to_0	
<input checked="" type="checkbox"/>		SW_IP	PSD (Parallel I/O) Intel FPGA IP		
		clk	Clock Input	Double-click to export	CLK_IP
		reset	Reset Input	Double-click to export	[8k]
		s1	Avalon Memory Mapped Slave	Double-click to export	
		external_connection	Conduit	control13_to_12_key11...	
<input checked="" type="checkbox"/>		LED_IP_1	PSD (Parallel I/O) Intel FPGA IP		
		clk	Clock Input	Double-click to export	CLK_IP
		reset	Reset Input	Double-click to export	[8k]
		s1	Avalon Memory Mapped Slave	Double-click to export	
		external_connection	Conduit	ledr63_to_32	

$Y = \sim(\sim(\sim A \mid (A \& B \& C)))$;
 $Z = (\sim(\sim A \mid (A \& B \& C))) \& ((A \& B \& C) \mid \sim C) \& \sim C$;

```
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();  
}
```

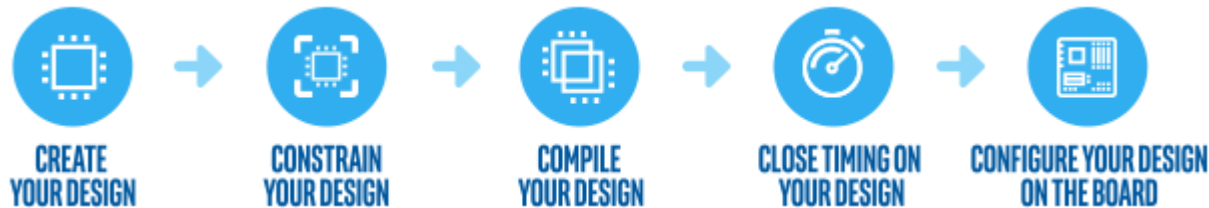


WHAT IS IP (INTELLECTUAL PROPERTY)?

- 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.

INTRO TO 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

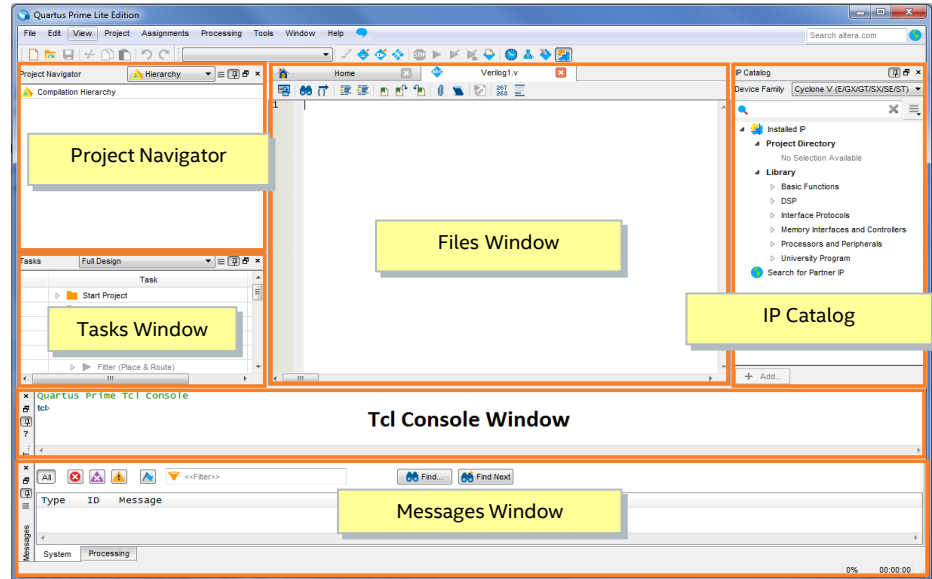


- Many formats to program an FPGA
 - In this class we will use a “.sof” file (SRAM object file)
 - The .sof file is “volatile” and needs to be reprogrammed every time the board is restarted

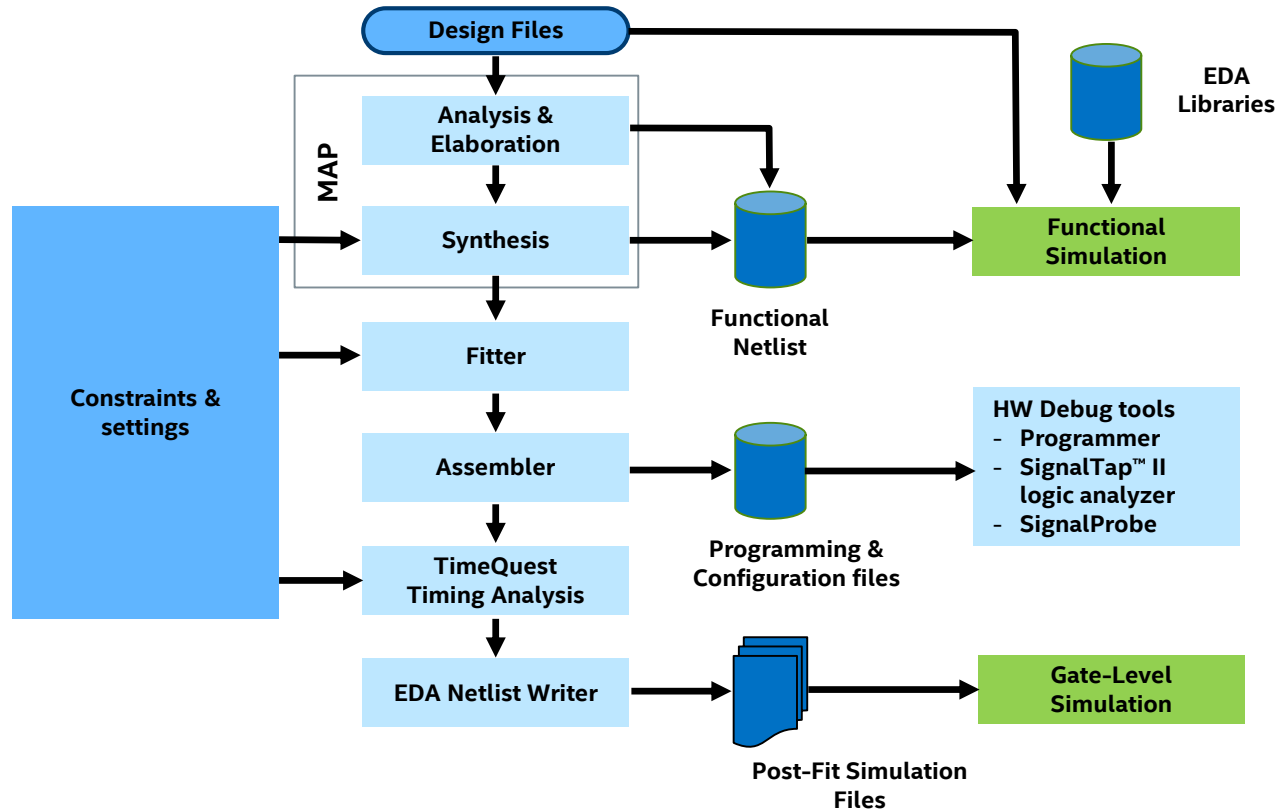
QUARTUS USER INTERFACE

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 OVERVIEW



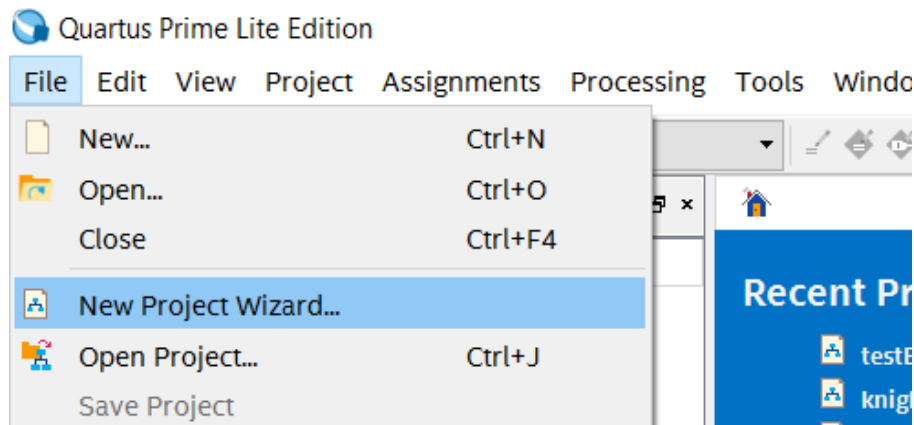
NEW PROJECT WIZARD

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.



HINT: FAMILY & DEVICE SETTINGS

New Project Wizard

Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: MAX 10 (DA/DF/DC/SA/SC)

Device: All

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter: 10M50DAF484

☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit cl
10M50DAF484C6GES	1.2V	49760	360	360	1677312	288
10M50DAF484C7G	1.2V	49760	360	360	1677312	288
10M50DAF484C8G	1.2V	49760	360	360	1677312	288
10M50DAF484C8GES	1.2V	49760	360	360	1677312	288

< Back Next > Finish Cancel Help

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
LED[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

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate
LED[4]	Output				PIN_L19	2.5 V (default)		12mA (default)	1 (default)
LED[3]	Output				PIN_AB8	2.5 V (default)		12mA (default)	1 (default)
LED[2]	Output				PIN_H6	2.5 V (default)		12mA (default)	1 (default)
LED[1]	Output				PIN_G8	2.5 V (default)		12mA (default)	1 (default)
LED[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)	
SW[4]	Input				PIN_U24	2.5 V (default)		12mA (default)	

0% 00:00:00

COMPILE YOUR DESIGN

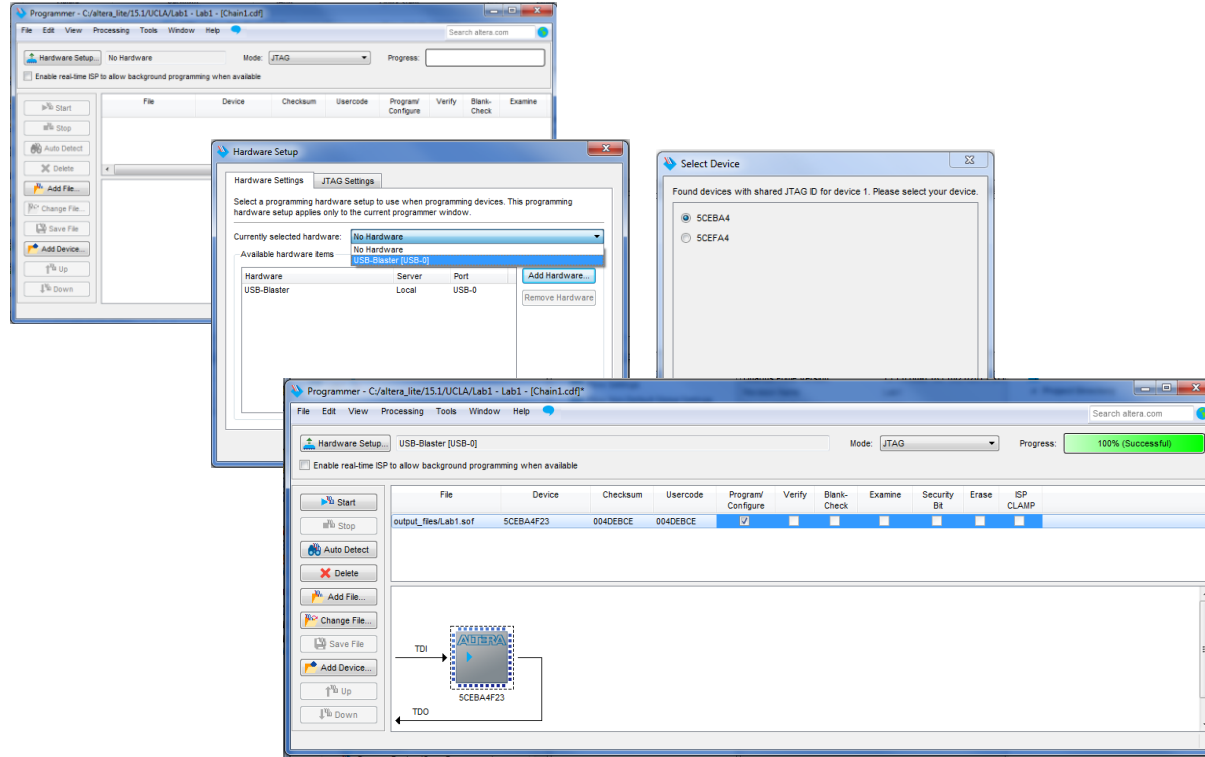
✓	▶ Compile Design	00:01:59
✓	▶ ▶ Analysis & Synthesis	00:00:40
✓	▶ ▶ Fitter (Place & Route)	00:00:42
✓	▶ ▶ Assembler (Generate programming files)	00:00:19
✓	▶ ▶ TimeQuest Timing Analysis	00:00:18

⚠ 332012 Synopsys Design Constraints File file not found: 'Lab1.sdc'.

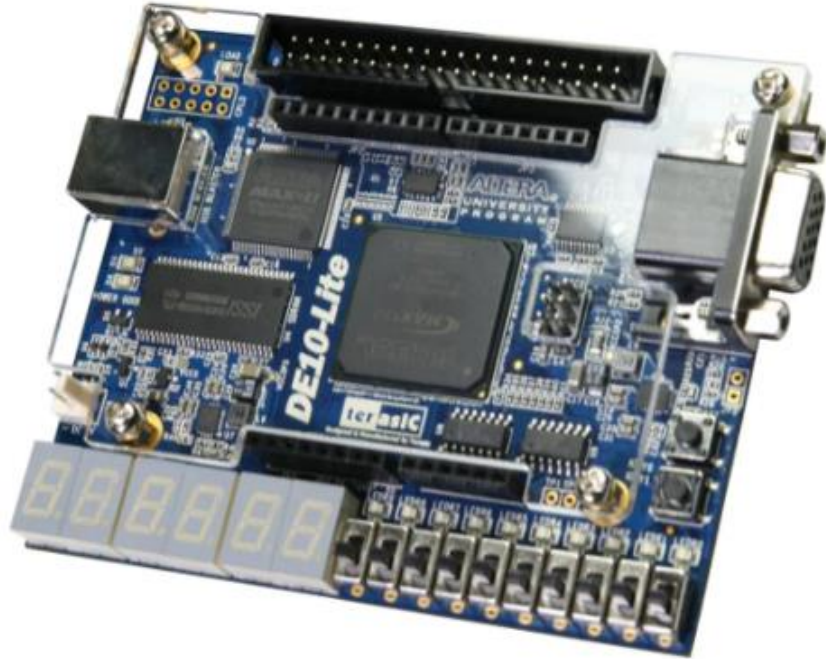


Warnings shown in blue won't prevent your design from compiling or being programmed, but they could indicate possible bugs. This lab does not have any design constraints, so the .sdc file is not needed. You will learn how to create one in the timing analysis workshop.

PROGRAM YOUR FPGA

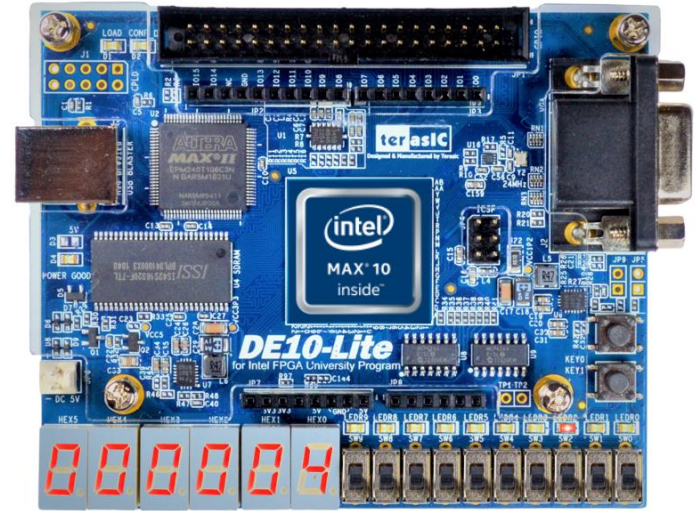


TEST YOUR DESIGN



Live Hardware

DE10-Lite Development Kit



32-bit Parameter 0

32-bit Parameter 1

32-bit Parameter 2

Board launched!
Connected to USB-Blaster [USB-0]
Time remaining: 20:19
Ping: 22 ms

Reset Timer

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

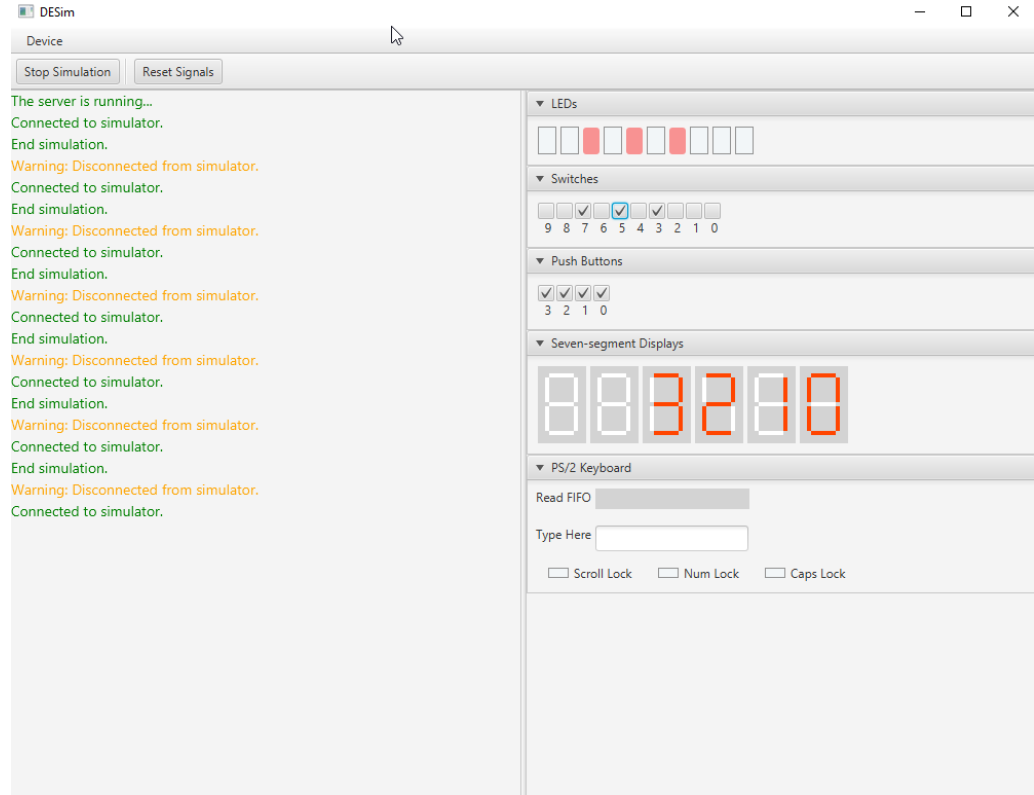
SIMULATION CONSOLE FOR FPGAS

Supports: Switches, Push Buttons,
LED, 7-Segments, PS2 character i/f
and VGA Display

Modelsim mimics behavior of
board

Good for initial bring-up and
debug

- No need for viewing waveforms



OVERVIEW

Support for remote GPIO-based labs.

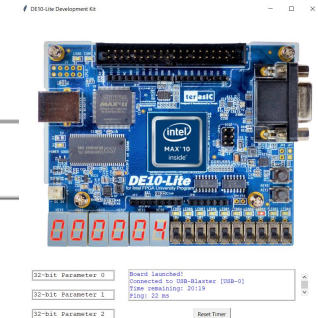
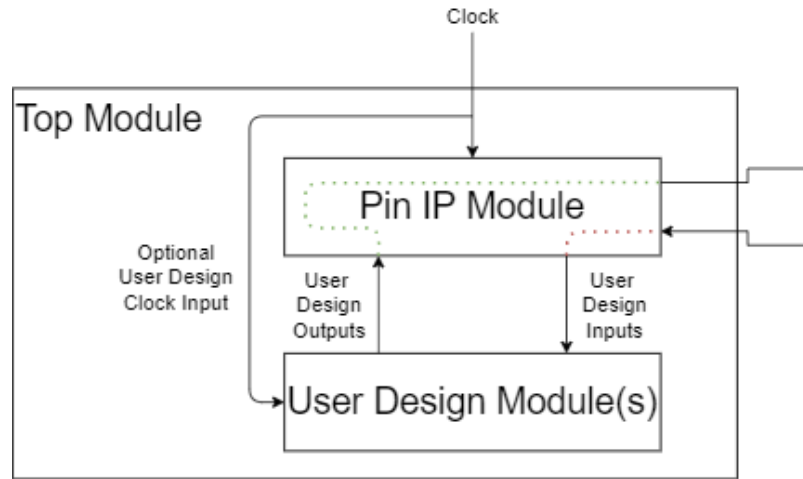
- Good for undergraduate curriculum.

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



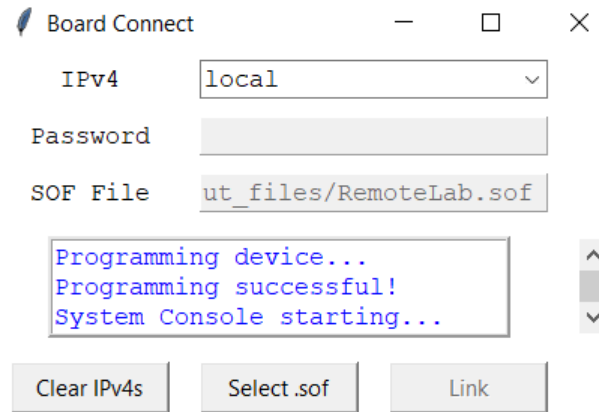
CONNECTING TO DEVELOPMENT KIT

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.

HOSTING DEVELOPMENT KITS

Requirements:

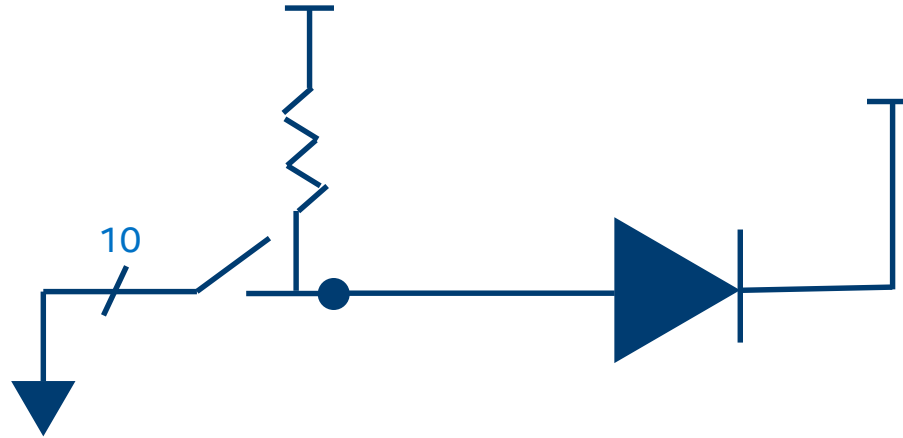
- Linux Ubuntu 18.04
- Static IP assignment
- Enough USB connections (supports USB hubs)
- Quartus Prime Lite Programmer and Tools package
- Clients with VPN privileges

Run setup bash script as root

Distribute IPv4/passwords to clients

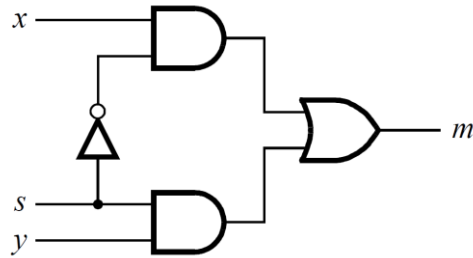
- Saved in jtag_server_passwords.txt

FIRST LAB: SWITCH TO LED



assign LED = SW;

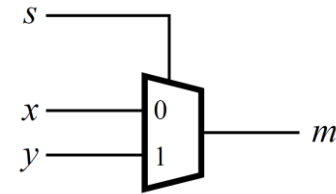
NEXT LAB: MULTIPLEXER



Circuit

s	m
0	x
1	y

Truth
table

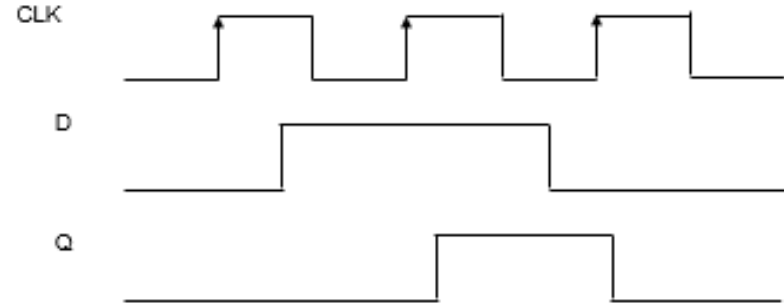
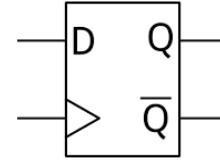


Symbol

The multiplexer can be described by the following Verilog statement:

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

KNIGHTRIDER



FINAL 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.

