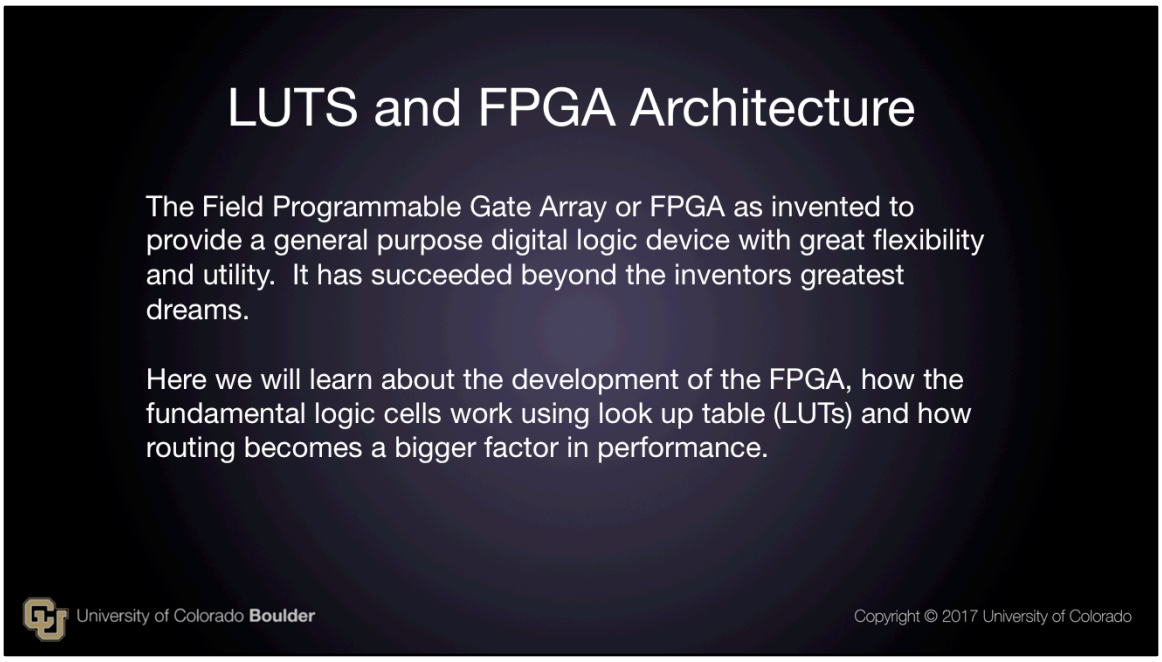
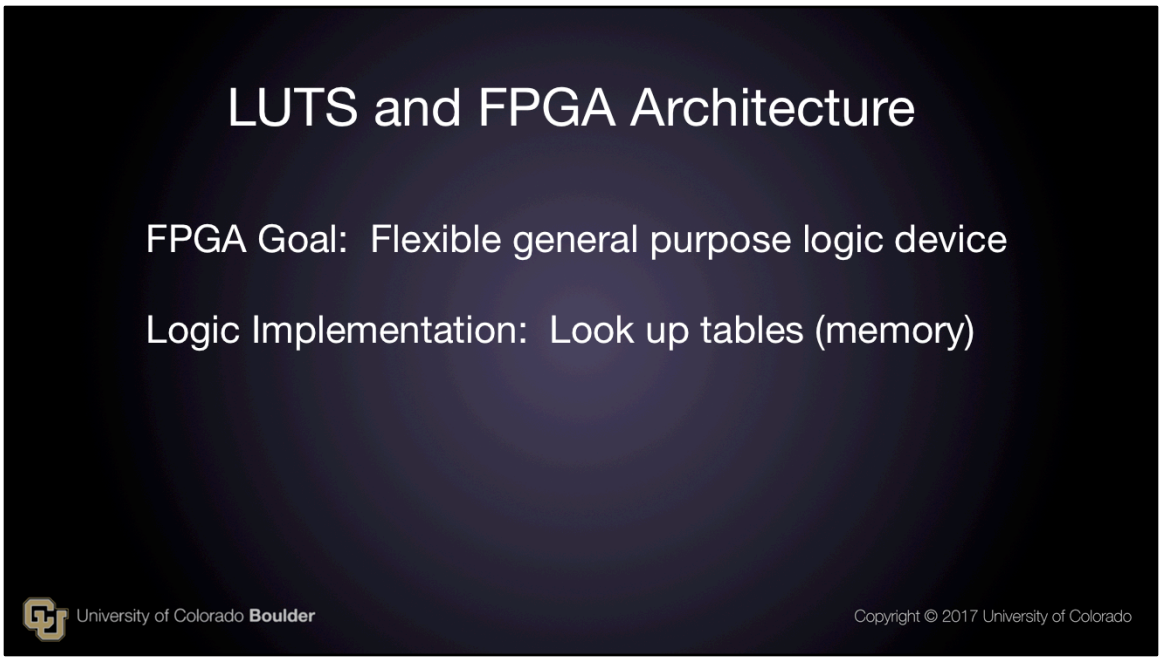
LUTS and FPGA Architecture

University of Colorado **Boulder**



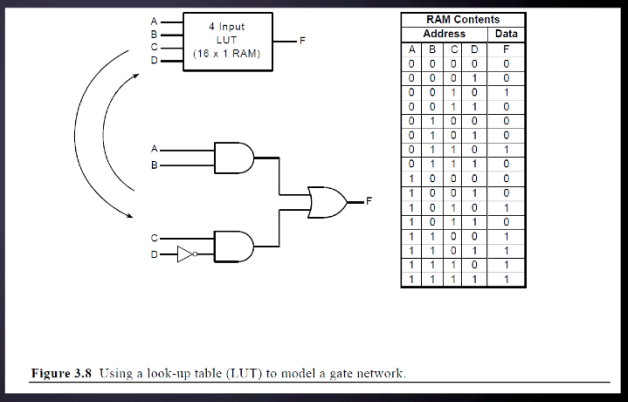
The Field Programmable Gate Array or FPGA as invented to provide a general purpose digital logic device with great flexibility and utility. It has succeeded beyond the inventors greatest dreams.
Here we will learn about the development of the FPGA, how the fundamental logic cells work using look up table (LUTs) and how routing becomes a bigger factor in performance.



With the intent to create a more flexible general purpose logic device, FPGAs were first designed to imitate Gate Arrays – integrated circuits with a sea of gates that were connected by added metal layers (wire routing) to define the functionality of the device.
However, instead of implementing the logic using transistor gates, the logic was in many cases implemented by memory fashioned as look up tables.

Title Helvetica Neue Regular 40pt

LUT can be used to implement any 4-input logic.



RAM Contents				
Address				Data
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Figure 3.8 Using a look-up table (LUT) to model a gate network.

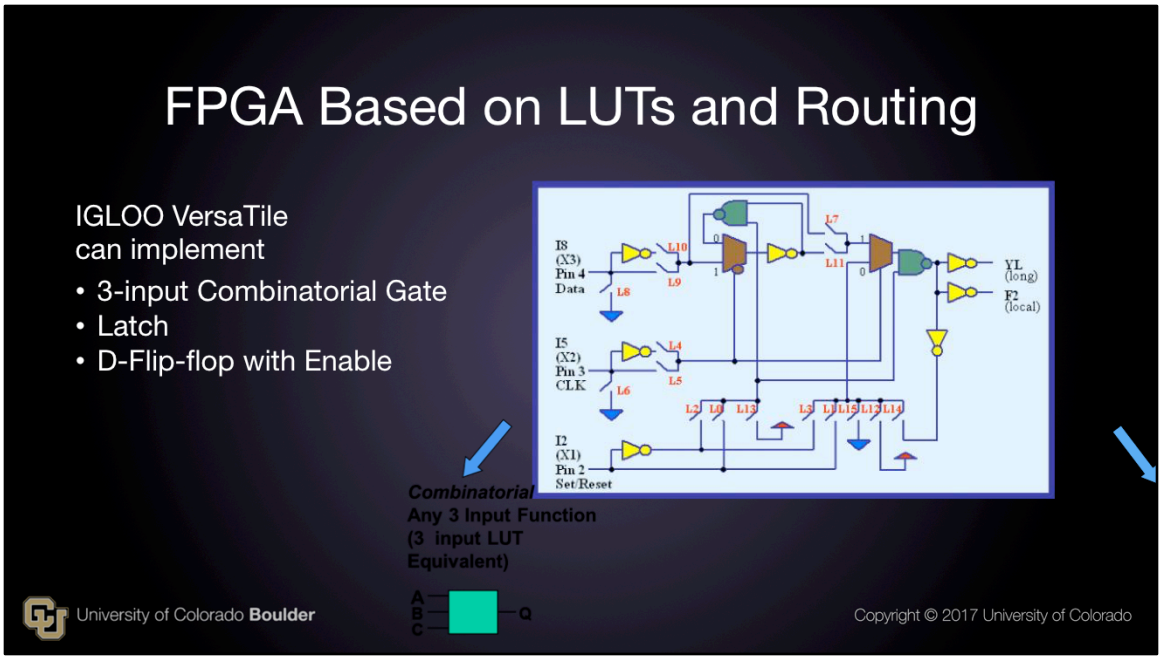
University of Colorado **Boulder**

This demonstrates how the LUT can be used to implement any 4-input, 1-output logic device.

Here, note the regular methodical way that the table is created. A has all zeros then all ones, B has 4 zeros and then 4 ones, etc. Then to create the output F, not that if A and B are true, F is true. This is the last 4 rows of the table, which implements the top AND gate. Also if C is true and D is not true, then F is true.

Given that all possible combination of the inputs are listed, we can create any logical expression for 4 inputs by how we program F. These expressions can also be implemented and AND-OR circuits as shown.

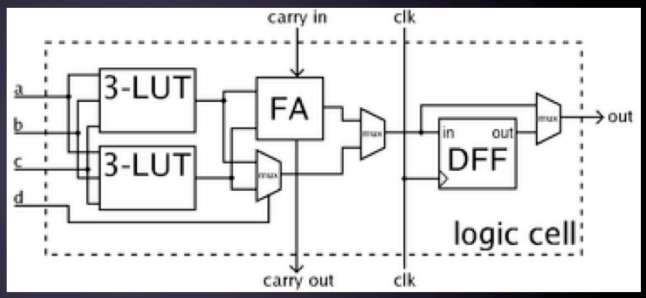
Semiconductor companies were good at making memories, which had regular highly optimized structures that were easily replicated. The first FPGAs were an array of logic cells consisting of 3 input LUTs.



http://www.actel.com/kb/article.aspx?id=SL1007

FPGAs architecture is different from CPLDs, as the logic element is smaller, usually implemented as a LUT. In this example, the Microsemi IGLOO Versatile can be either a 3 input gate of any type, or a sequential D flip flop or latch. The schematic depicts the structure of the logic element. Each L labeled switch is controlled by the configuration bits when the FPGA is programmed.

LUT can be used
To implement
any 4-input logic.



University of Colorado **Boulder**

https://en.wikipedia.org/wiki/Field-programmable_gate_array

Here is a simplified schematic of an FPGA logic cell. A typical cell consists of a 4-input LUT, a full adder (FA) and a D-type flip-flop. The LUTs are in this figure split into two 3-input LUTs. In *normal mode* those are combined into a 4-input LUT through the left mux. In *arithmetic* mode, their outputs are fed to the FA. The selection of mode is programmed into the middle multiplexer.

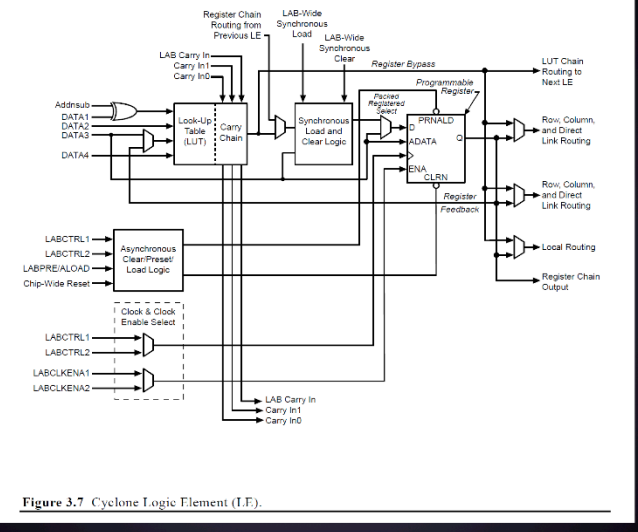
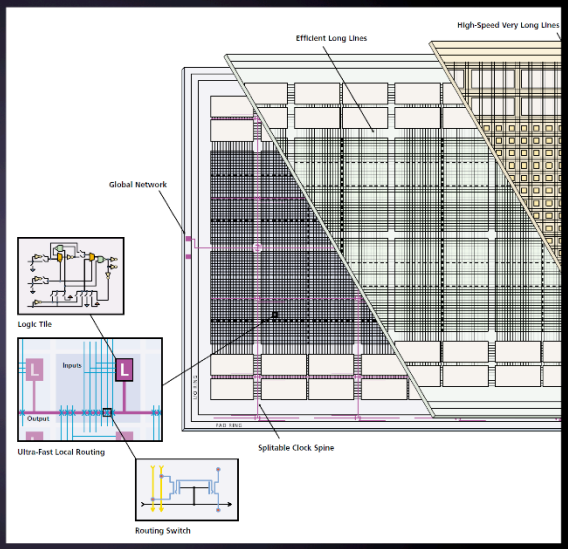


Figure 3.7 Cyclone Logic Element (LE).

This is a schematic of the Altera Cyclone Logic Element or LE. Inputs are on the left, outputs on the right. The core of this LE is a 4-input LUT which can be used to create any 4 input logic combinatorial circuit. The LUT output is part of a carry chain structure used to speed up arithmetic computations. The LUT output can be routed out directly, or registered by the flip flop. As you can see this circuit is design with versatility in mind.

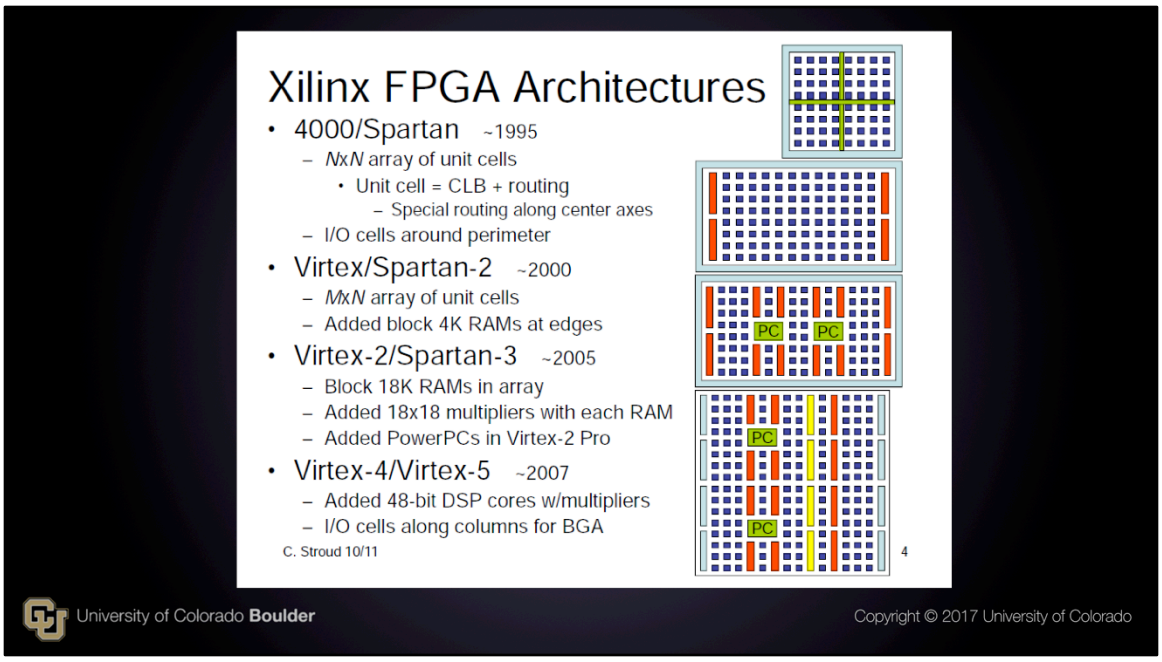
The FPGA user could determine the function of a small cluster of gates (the logic cell) and how that cluster connected to other clusters on the chip, gradually building up a circuit by connecting logic cells. However, to implement a high-fan in function like an address decoder, FGPAs need to cascade many stages of logic elements. This can lead to excessive and somewhat unpredictable delays. Timing analyzer tools help resolve these problems to some extent.

FPGA Routing Architecture



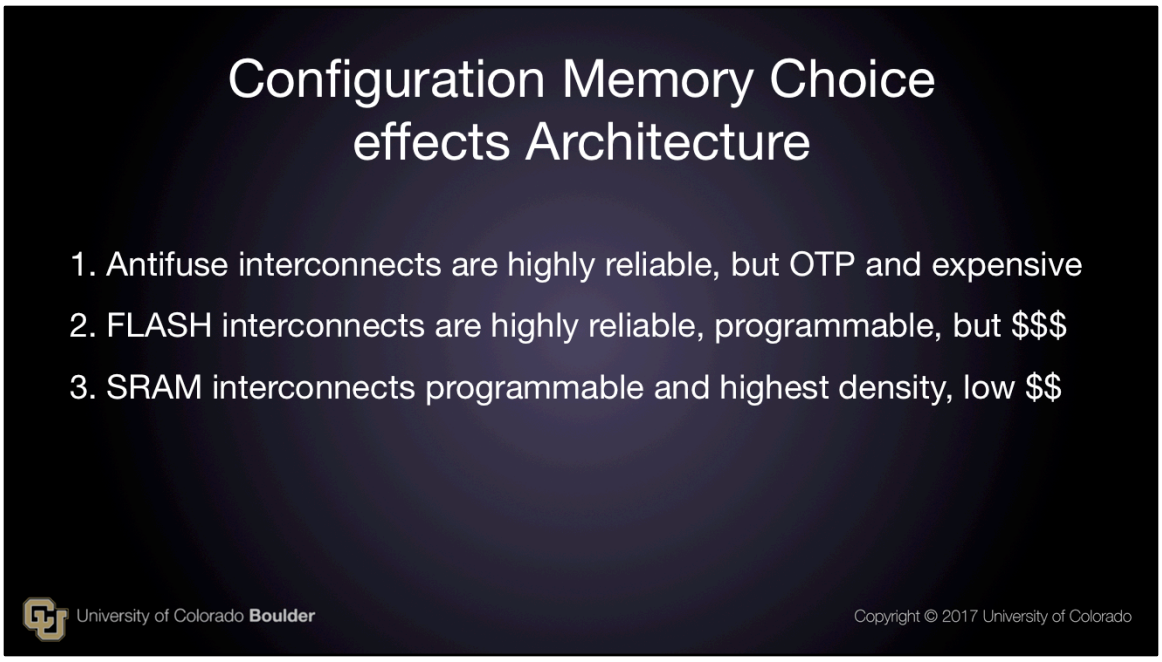
University of Colorado **Boulder**

The routing in an FPGA is hierarchical. At the lowest level, we have a routing switch that is controlled by a bit of SRAM, or a FLASH bit, or a fuse. There are many routing switches in the local routing to connect various logic elements or blocks. The FPGA is made of a sea of these logic blocks. Groups of the blocks are connected by long lines or very long lines. Common signals like clocks are routed on special routing networks called global networks.



This shows the evolution of FPGA architecture. In 1995, Xilinx made FPGAs with an NxN array of cells with global routing down the center spines. By 2000, they had added block 4K RAMS. In 2005, multipliers were added along with Hard Core processors (PowerPC) – among the earliest true PSOC parts. In 2007, they added DSP cores.

From this evolution it can be seen that FGPAs no longer operate in the realm of glue logic, but are becoming subsystems or complete systems in and of themselves. This development is fueled in part by the use of SRAM memory for the routing switches which has higher density than EEPROM as RAM processes are the first ones proven in a new semiconductor process geometry. The other development was the use of new means of design entry other than Boolean expressions, including schematic entry and HDL languages like Verilog and VHDL.



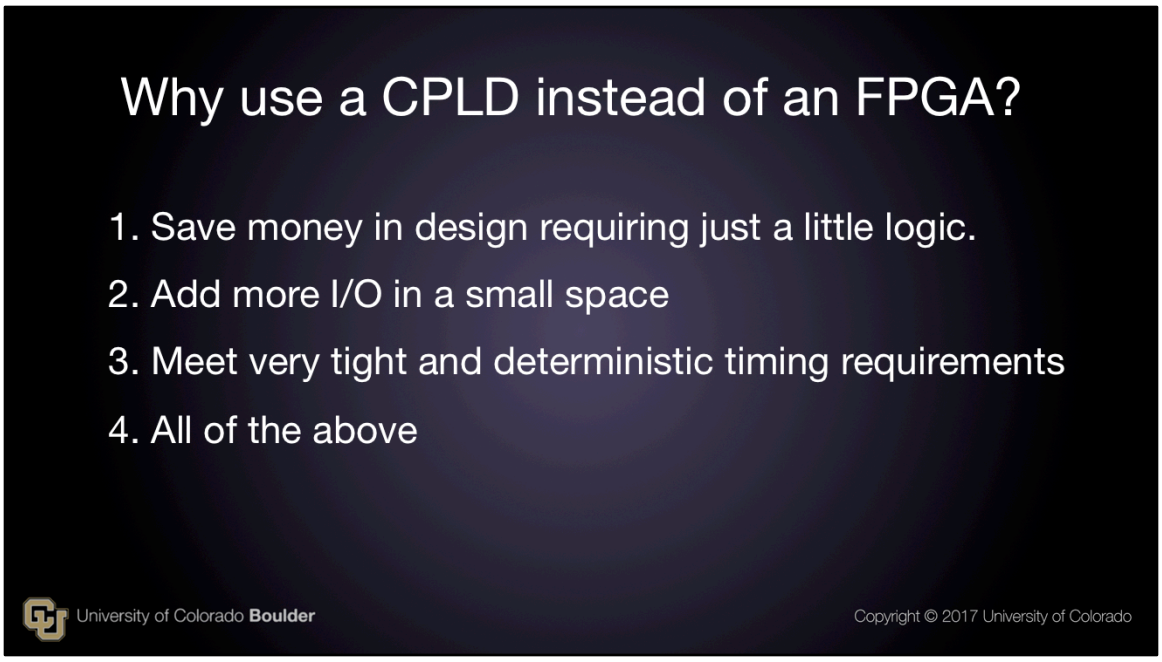
The type of configuration memory used to create the routing switches is the most important aspect in determining differences in performance and capabilities in FPGAs.

Many Programmable Logic Choices

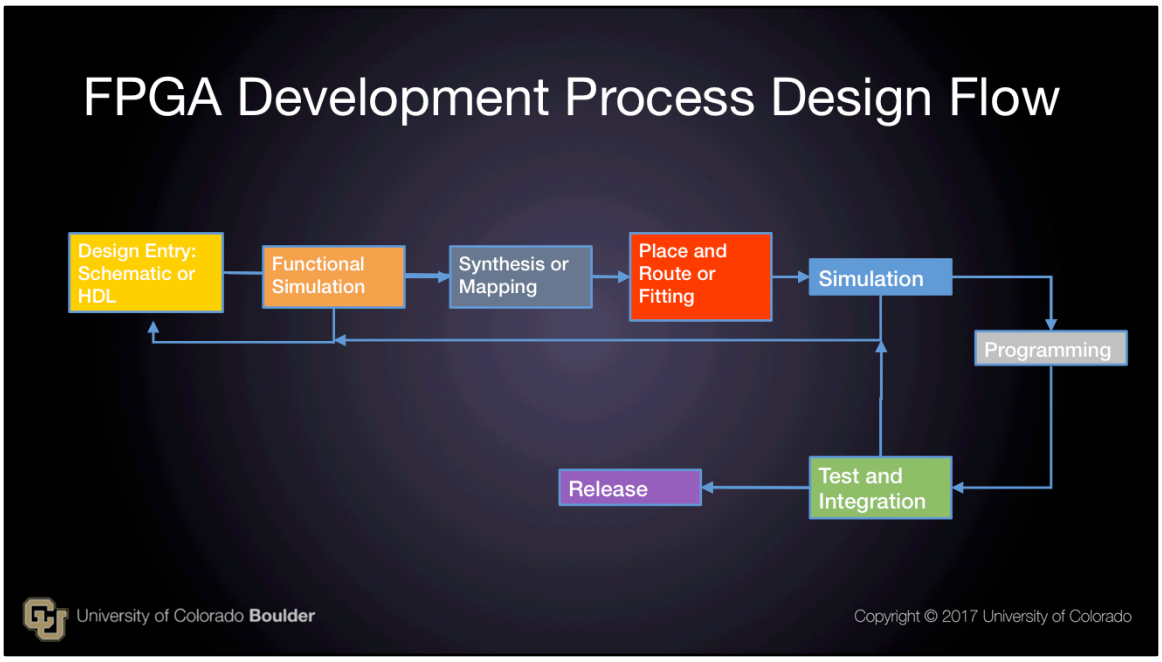
Product	Xilinx	Altera	Lattice	Microsemi
FLASH CPLD	X	X		
SRAM CPLD			X	
AntiFuse FPGA				X
FLASH FPGA	X			X
SRAM FPGA	X	X	X	
FLASH SoC				X
SRAM SoC	X	X		

University of Colorado **Boulder**

The 4 major PLD vendors have a variety of offerings that include many configuration types.



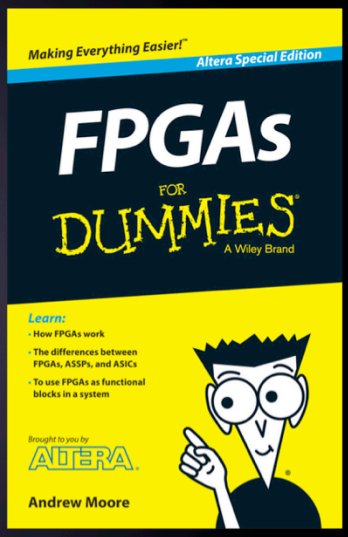
D all of the above



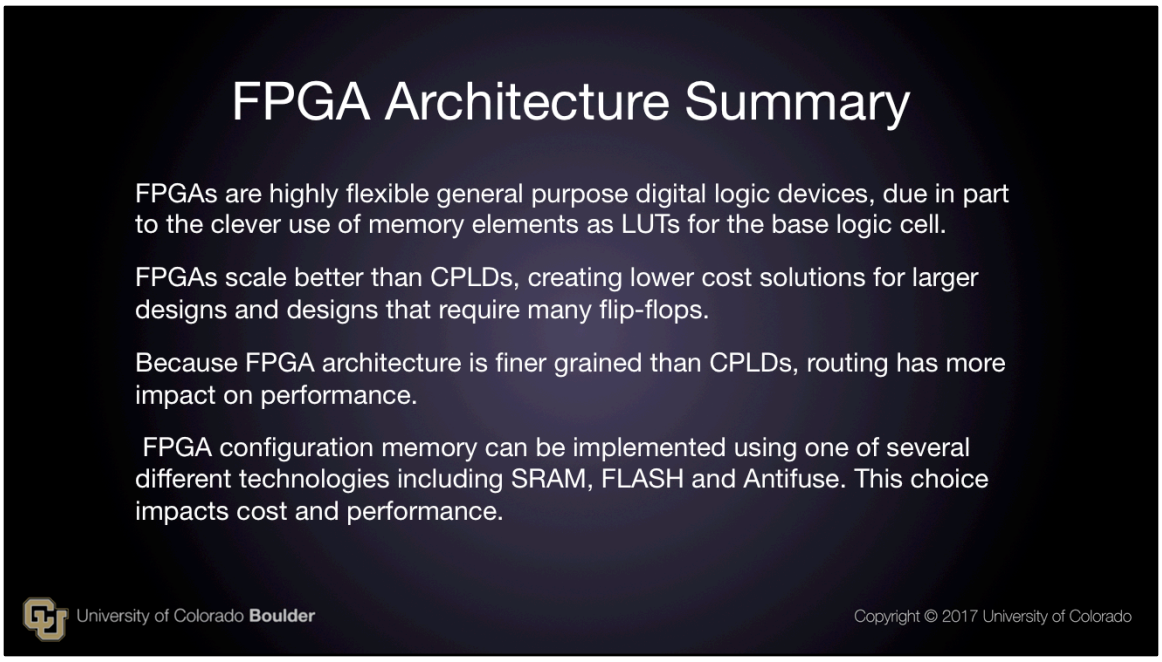
In module 2, you will learn how to create and FPGA design using FPGA design tools. The tools will use a process flow like the one depicted in the picture. The design begins with design entry, continues with simulation to verify the logic is implemented as expected, and then the tool will perform synthesis (also called mapping) to map the logic to the device architecture. Next the tool will create the interconnection between cells by place and routing (or fitting) the design. Another simulation after the fitting is done is a good practice. If this looks good, the next step to use the tool to create a programming file, which is then downloaded into the FPGA device for testing.

Great Free Resource!

See www.altera.com/en_US/pdfs/literature/misc/fpgas_for_dummies_ebook.pdf



This definition is from the book.



In this video we have learned

- FPGAs are highly flexible general purpose digital logic devices, due in part to the clever use of memory elements as LUTs for the base logic cell.
- FPGAs scale better than CPLDs, creating lower cost solutions for larger designs and designs that require many flip-flops.
- Because FPGA architecture is finer grained than CPLDs, routing has more impact on performance.
- FPGA configuration memory can be implemented using one of several different technologies including SRAM, FLASH and Antifuse. This choice impacts cost and performance.