

Designing Logic with LUTs



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

LUTS and Logic Design

- FPGAs use Look up tables (LUTs) as the base logic element.
- LUTs are quite versatile for logic design.



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

In this video we will consider the use of LUTs in logic design. FPGAs use Look up tables (LUTs) as the base logic element. LUTs are quite versatile for logic design. Learning to love LUTs takes some time, but will be worth the effort.

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

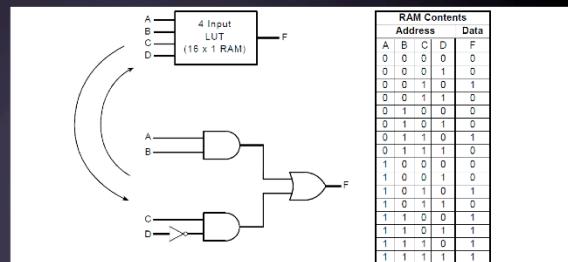
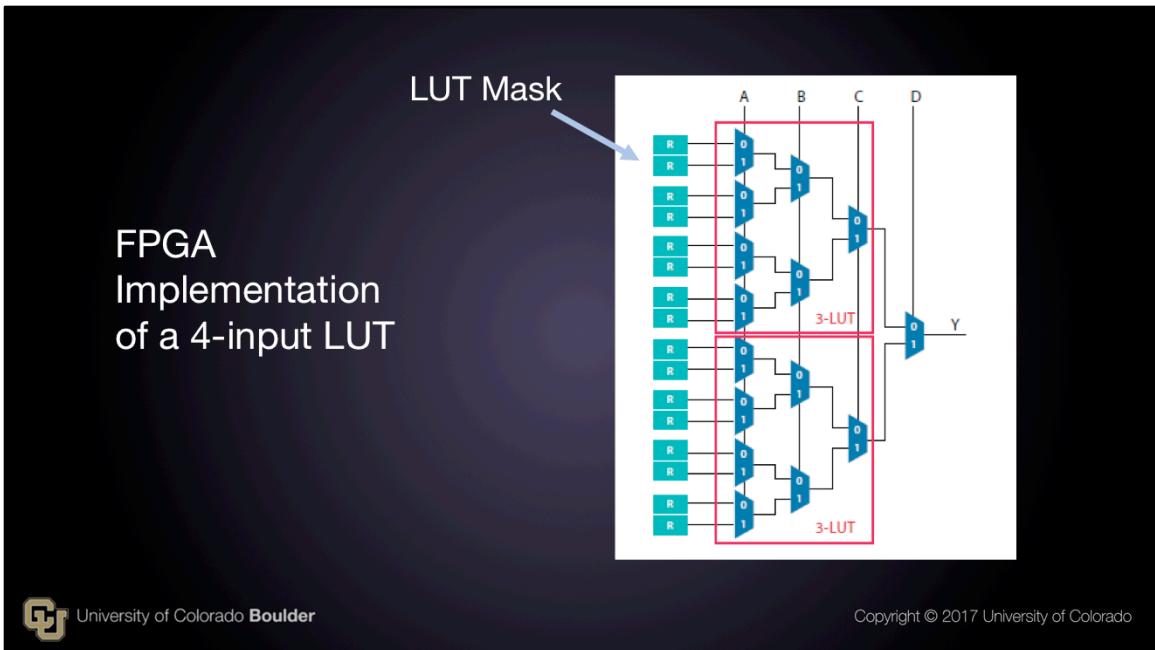


Figure 3.8 Using a look-up table (LUT) to model a gate network.

In the previous video we discussed how LUTS can be used to implement any n-input logic expression, n being the number of inputs to the LUT, as in this example. For a 3-input LUT, we need 8 memory locations, for 4 input, 16 memory locations, and so on.

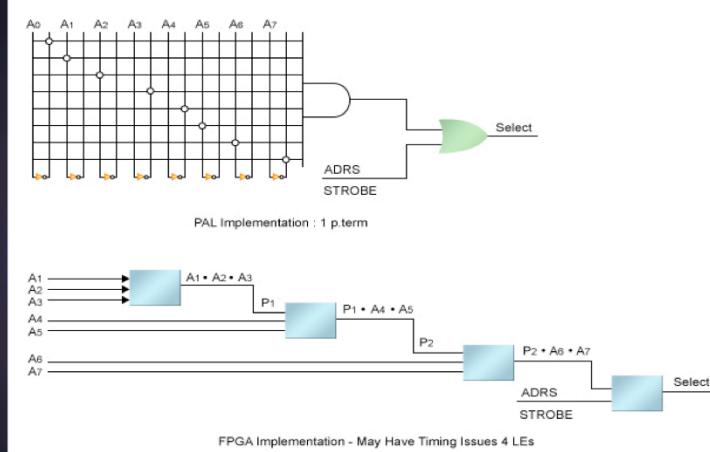


www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf

A LUT is typically built out of SRAM bits to hold the configuration memory LUT-mask and a set of multiplexers to select the bit of SRAM that is to drive the output. The figure shows a 4-LUT, which consists of 16 bits of SRAM and a 16:1 multiplexer implemented as a tree of 2:1 multiplexers. The 4-LUT can implement any function of 4 inputs (A, B, C, D) by setting the appropriate value in the LUT-mask. it can also be built from two 3-LUTs connected by a 2:1 multiplexer.

We can implement more logic in the larger LUT devices. However, there is a tradeoff between LUT size and the amount of routing needed. Bigger LUTs means less routing, but also more memory locations. Let's look at some examples to see why.

8-bit address decoder using 8-input logic versus 3-input LUTs



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

https://www.altera.com/solutions/technology/system-design/articles/_2013/in-the-beginning.html

If we used a 8-input LUT to create an 8-bit address decoder, only one logic cell would be needed. The figure shows an implementation in a PAL, but the 8-input LUT would look similar. However, if we use 3-input LUTs, then 4 levels of logic elements and the associated routing between them will be needed as shown.

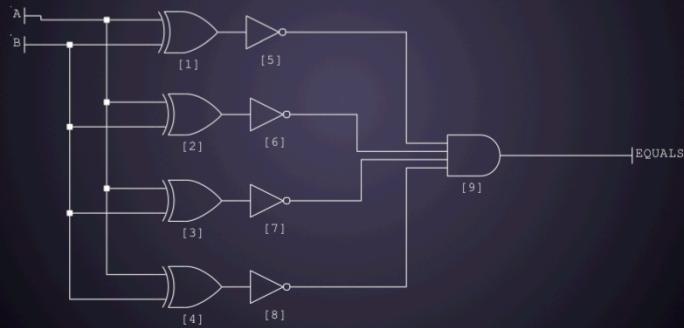
How many 4-input LUTs will a 4-bit comparator require?

- A. 1
- B. 2
- C. 3
- D. 4

How many 4-input LUTs will a 4-bit comparator require?

- A. 1
- B. 2
- C. 3
- D. 4

How many 4-input LUTs will a 4-bit comparator require?



[Annotate] The XNOR of the first 2 bits can be implemented in 1 4-input LUT, as can bits 3 and 4. The 4 input AND on the end can also be created in a 4-input LUT, so the answer is 3 total 4-input LUTs are required to implement a 4-bit comparator.

How many 4-input LUTs will a full adder require?

- 1
- 2
- 3
- 4

How many 4-input LUTs will a full adder require? Think about this, and we will look at it in depth in our next video.

LUTs and Logic Design Summary

- ⇒ FPGAs use LUTs of various sizes to implement logic.
- ⇒ There is a tradeoff between the size of the LUT and the amount of routing required. Larger LUTs can create more logic which will require less routing; Smaller LUTs will require more routing, but offer better logic efficiency.
- ⇒ LUTs can be used to create a variety of combinatorial logic functions.



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

In this video we have learned

- FPGAs use LUTs of various sizes to implement logic.
- There is a tradeoff between the size of the LUT and the amount of routing required. Larger LUTs can create more logic which will require less routing; Smaller LUTs will require more routing, but offer better logic efficiency.
- LUTs can be used to create a variety of combinatorial logic functions.