

# Designing Multipliers in FPGAs



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

# Designing Multipliers in FPGAs

- ⇒ Multiplying 2 numbers is a common digital logic or ALU function that can take considerable amounts of circuitry.
- ⇒ There are a variety of ways to implement multiplier circuits within FPGAs



University of Colorado **Boulder**

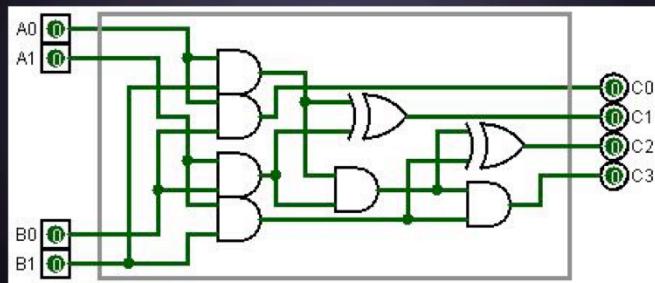
Copyright © 2017 University of Colorado

Multiplying 2 numbers is a common digital logic or ALU function that can take considerable amounts of circuitry.

There are a variety of ways to implement multiplier circuits within FPGAs

## Multipliers in Digital Logic: Gates

Adders are built with Gates, can we build multipliers this way, too?



A 2-bit by 2-bit binary multiplier



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

Adders are built with Gates, can we build multipliers with gates as well?

Here is an example, a 2-bit by 2-bit binary multiplier.

## Multipliers in Digital Logic: Gates

Adders are built with Gates, can we build multipliers this way, too?

Yes, using an array multiplier, which works like multiplication by hand, including partial product generation and partial product summation.

But the number of gates increases as  $n^2$

Beyond 4x4, a sequential design is more efficient

# bits	Approximate # gates
2x2	8
3x3	54
4x4	96
5x5	150
6x6	216
7x7	294
8x8	384



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

Yes, using an array multiplier, which works like multiplication by hand, including partial product generation and partial product summation.

But the number of gates increases as  $n^2$ . Beyond 4x4, a sequential design is more efficient.

## Many Ways to Build a Multiplier in an FPGA

1. Combinational circuits – fast but big
2. Sequential shift and add (state machine approach)
3. Specialty algorithms, like Booth's Algorithm or the Dadda Multiplier or Wallace Tree Multipliers
4. Memories (Look up tables)
5. Some combination of the above
6. Hard Multiplier Blocks if available



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

There are many ways to build a multiplier in an FPGA.

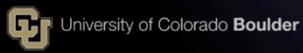
1. Combinational circuits – fast but big
2. Sequential shift and add (state machine approach) – small but slow
3. Specialty algorithms, like Booth's Algorithm or the Dadda Multiplier or Wallace Tree Multipliers – fast and small but complex
4. Memories (Look up tables) – simple but big
5. Some combination of the above
6. Hard Multiplier Blocks if available – best solution in most cases.

## Many Ways to Build a Multiplier in an FPGA

Sequential shift and add (state machine approach)

Smaller in area than combinational multipliers beyond

4x4 multiplies, but slower as  $2n$  clock cycles are required for shifting and adding.



Copyright © 2017 University of Colorado

Consider the Sequential shift and add (state machine approach)

This is smaller in area than combinational multipliers beyond 4x4 multiplies, but slower as  $2n$  clock cycles are required for shifting and adding.

# Many Ways to Build a Multiplier in an FPGA

## Memories for Multiplication

A 4x4 multiplication can be achieved with a memory with an 8-bit address.

A7...A4 is the multiplier and A3...A0 is the multiplicand.  
28 or 256 locations are needed in the memory.

Although memory cells are inexpensive, beyond 8x8 this becomes too expensive.



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

For small inputs, Memories for Multiplication can be a good solution.

A 4x4 multiplication can be achieved with a memory with an 8-bit address. The addresses A7...A4 can represent the multiplier and A3...A0 is the multiplicand.  $2^8$  or 256 locations are needed in the memory. Although memory cells are relatively inexpensive, beyond 8x8 this becomes too large and costly.

## Many Ways to Build a Multiplier in an FPGA

Hard Multipliers provide the greatest speed of any FPGA implementation, running at 200 MHz or more (50 ns per multiplication).

Typical MAX 10 or Cyclone V parts will have dozens to hundreds of 18 x 18 multiplier circuits available for your use.



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

Hard Multipliers provide the greatest speed of any FPGA implementation, running at 200 MHz or more (50 ns per multiplication).

Typical MAX 10 or Cyclone V parts will have dozens to hundreds of 18 x 18 multiplier circuits available for your use.

## Designing Multipliers in FPGAs Summary

- D- Creation of Digital Multipliers is a challenging design problem. Most solutions have significant drawbacks
- D- There are many ways to implement multipliers in FPGAs, including combinational circuits, sequential circuits (state machines), specialty circuits (Booth's algorithm, etc.), memories, and hard multiplier blocks.
- D- Many FPGAs contain multiple hard multiplier blocks which allow very fast 18x18 multiplication, providing a perhaps the best solution for implementing multipliers in digital logic.



University of Colorado **Boulder**

Copyright © 2017 University of Colorado

[https://en.wikipedia.org/wiki/System\\_on\\_a\\_chip](https://en.wikipedia.org/wiki/System_on_a_chip)

In this video we have learned

- Creation of Digital Multipliers is a challenging design problem. Most solutions have significant drawbacks
- There are many ways to implement multipliers in FPGAs, including combinational circuits, sequential circuits (state machines), specialty circuits (Booth's algorithm, etc.), memories, and hard multiplier blocks.
- Many FPGAs contain multiple hard multiplier blocks which allow very fast 18x18 multiplication, providing a perhaps the best solution for implementing multipliers in digital logic.