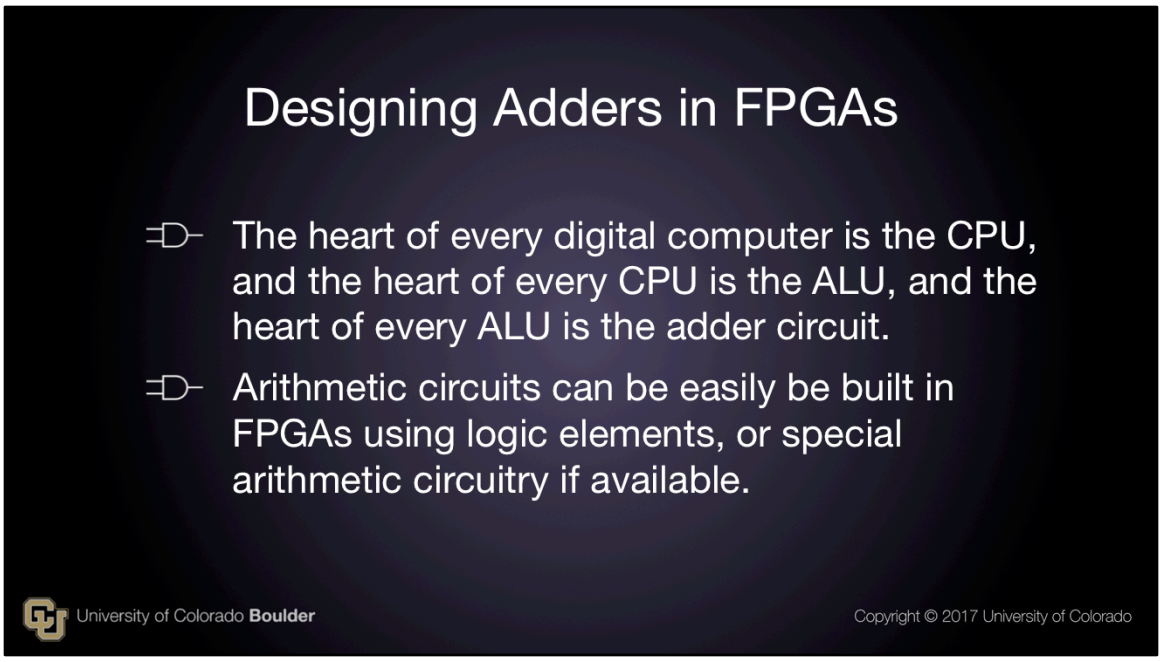
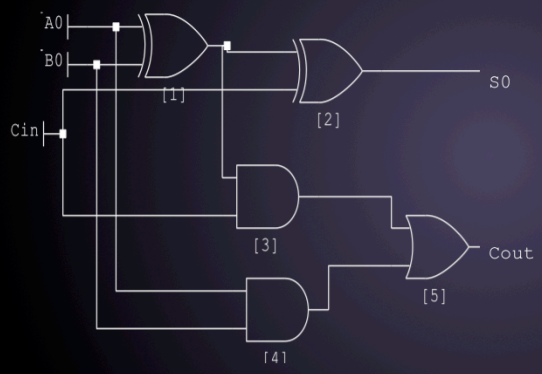
Designing Adders in FPGAs

University of Colorado **Boulder**



The heart of every digital computer is the CPU, and the heart of every CPU is the ALU, and the heart of every ALU is the adder circuit. Adders are fundamental. Once you can add you can subtract by adding a complement, and you can also multiply by counting how many time you add. It's no surprise then, that FPGAs are infused with structures to ease the design and improve the performance of adders.

Full 1-bit Adder in Gates



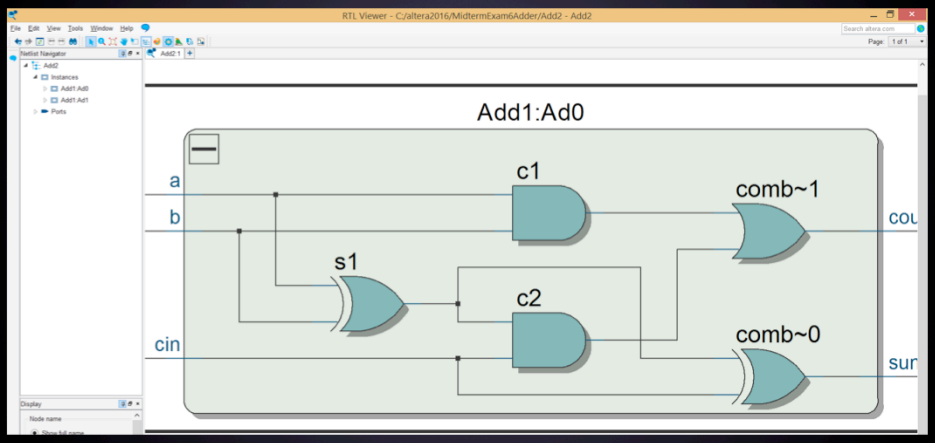
Truth Table

A0	B0	Cin	So	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

University of Colorado **Boulder**

A full bit adder can be constructed of as few as 5 gates as shown here. In an FPGA, we could implement this in a 3-input or 4-input LUT if it had 2 outputs. In some cases the LUT architecture will have an additional output for the carry out in addition to the sum out.

Full 1-bit Adder implemented in FGPA , RTL View

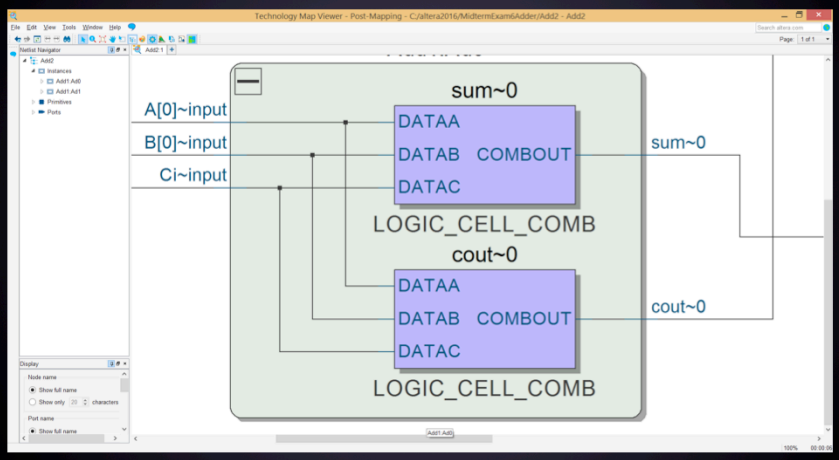


University of Colorado **Boulder**

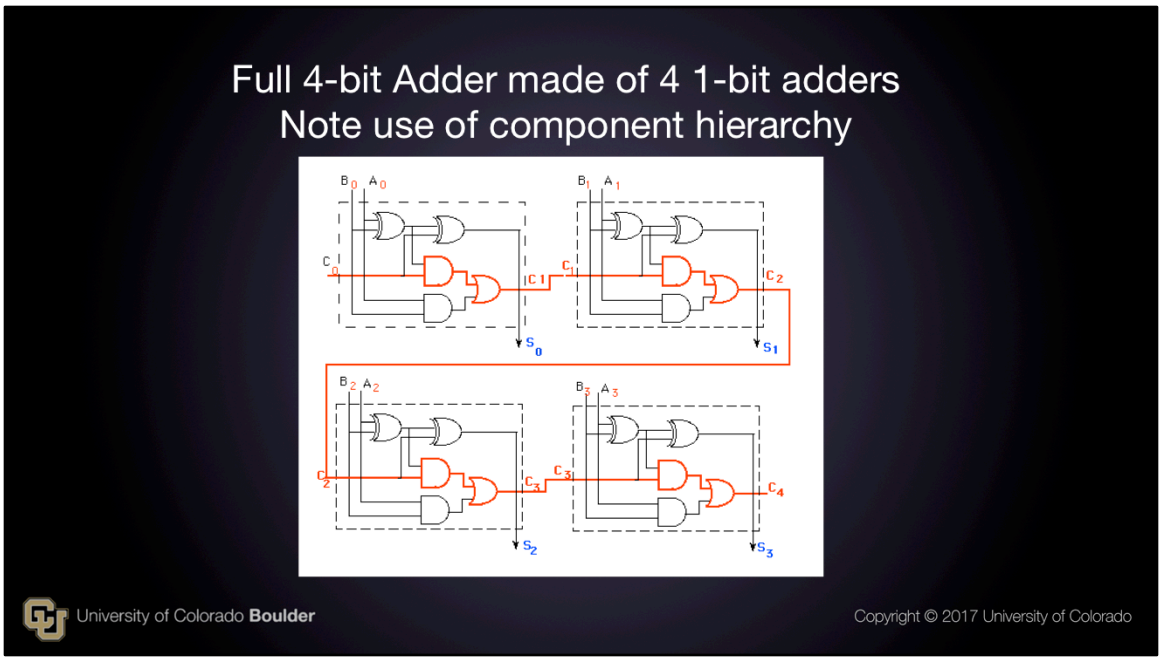
The is the result of the implementation of an adder in the Altera MAX10 FPGA, shown here using the RTL viewer tool which is part of the Quartus Prime FPGA design suite. It looks just like the gate diagram.

RTL stands for Register Transfer Logic or Register Transfer Level. RTL is based on the concept that movement of data through digital logic can be viewed as logic diagrams with expressions representing the data variables that are stored in registers. It is generally a higher level representation than the gate level.

Full 1-bit Adder implemented in FGPA , Technology View

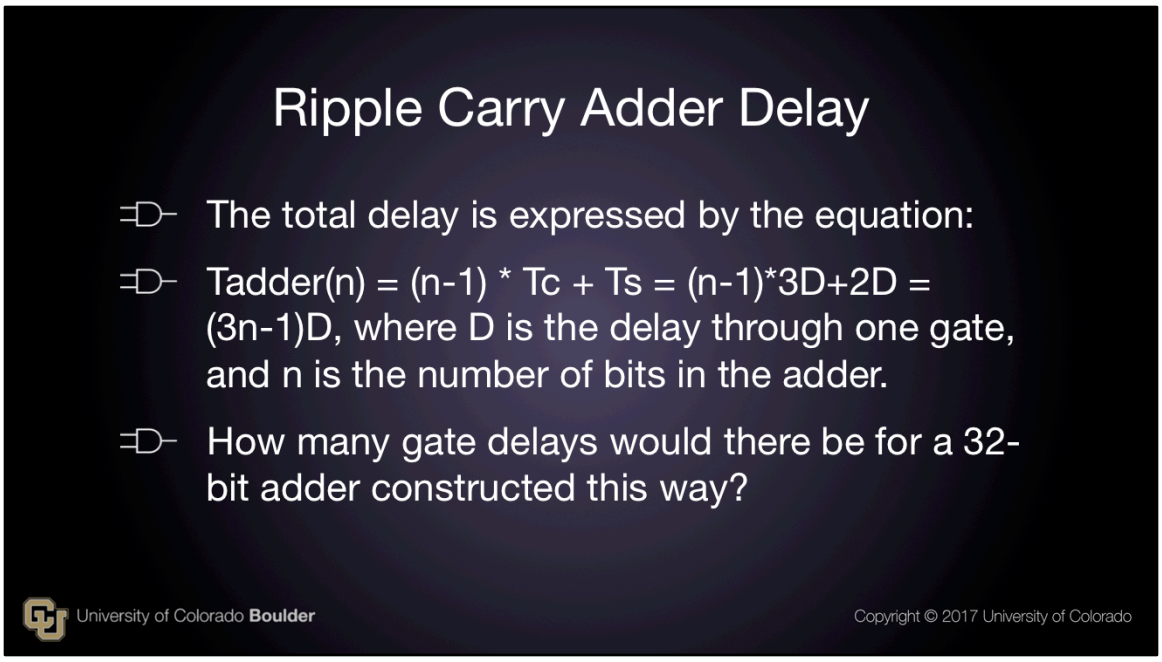


The is the result of the implementation of an adder in the Altera MAX10 FPGA, shown here using the Technology Map viewer tool. It shows the implementation is made using 2 3-input LUTs.



http://www.pldworld.com/_hdl/2/-seas.upenn.edu/_ese201/lab/CarryLookAhead/CarryLookAheadF01.html

We can use full 1-bit adders as components to create n-bit adders by connecting them together. Here we see the connection for a 4-bit Adder made of 4 full 1-bit adders. This is known as a ripple-carry adder.
It is fast to design, however the ripple-carry adder is relatively slow in performance, since each full adder must wait for the carry bit to be calculated from the previous full adder.



https://en.wikipedia.org/wiki/Adder_(electronics)

The total delay is expressed by the equation:

Tadder(n) = (n-1) * Tc + Ts = (n-1)*3D+2D = (3n-1)D, where D is the delay through one gate, and n is the number of bits in the adder. How many gate delays would there be for a 32-bit adder constructed this way?

[pause] 95.

Full 4-bit Adder made of using Carry Look Ahead (CLA) Adders to reduce delay

The Carry bit can be calculated as generate and propagate terms:

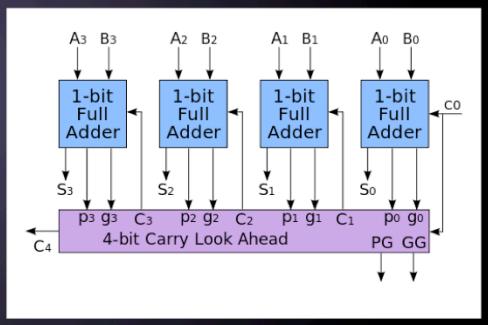
Ci+1 = Gi + Pi.Ci (1)

in which

Gi = Ai.Bi (2)

Pi = (Ai Å Bi) (3)

Now the delay is roughly equal to n



http://www.pldworld.com/_hdl/2/-seas.upenn.edu/_ese201/lab/CarryLookAhead/CarryLookAheadF01.html

We can use more sophisticated circuits to create better performing adders with less delay. One such possiblity is the Carry Look Ahead Adder shown here.

The Carry bit can be calculated as generate and propagate terms:

Ci+1 = Gi + Pi.Ci (1)

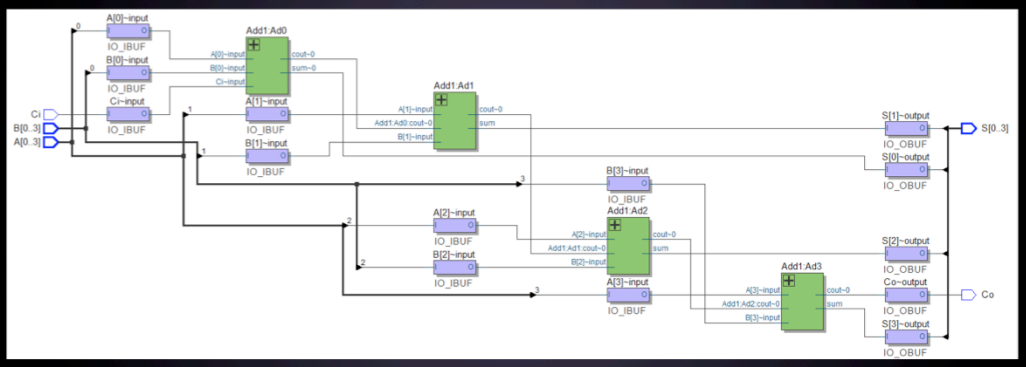
in which

Gi = Ai.Bi (2)

Pi = (Ai Å Bi) (3)

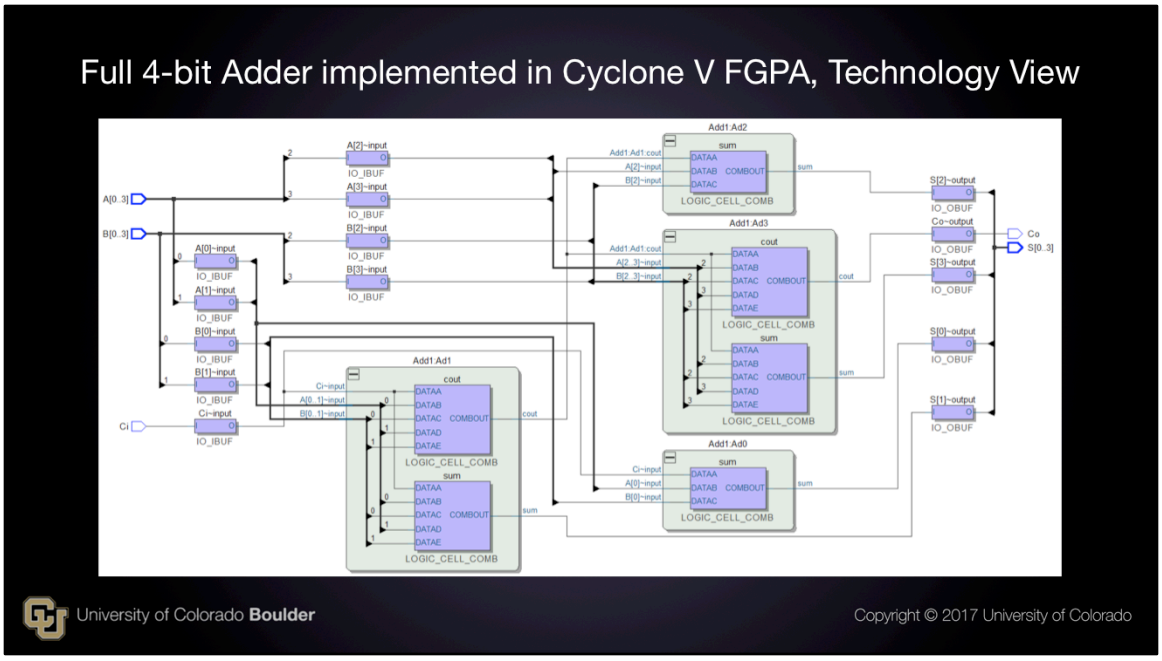
Now the delay is roughly equal to n, a big improvement over the ripple-carry adder, but at the cost of a much more complex carry bit calculation.

Full 4-bit Adder implemented in FGPA , Technology View



The is the result of the implementation of the 4-bit ripple-carry adder in the Altera MAX10 FPGA, shown here using the Technology Map viewer tool. It shows the implementation is made of a cascade of 4 pairs of 3-input LUTs. The delay through this circuit will be only 4 LUT delays, not 11 gate delays as based on the delay equation.

Furthermore, we may get more of advantage if we use a more advanced FPGA Architecture.



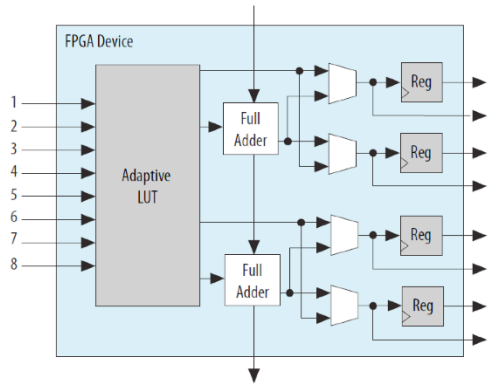
The is the result of the implementation of the 4-bit adder in the Altera Cyclone V FPGA, shown here using the Technology Map viewer tool. It shows the implementation is made using 3 6-input LUTs. Instead of 1-bit adders, the 6-input LUT is used to implement a 2-bit adder in 2 cases, for the 1 bit and for the 3 bit. The other 6-input LUT is broken into 2 3-input LUTs to generate sums for bit 0 and bit 2.

What is the delay through this circuit? Only 2 LUT delays, considerably better than we may have originally thought.

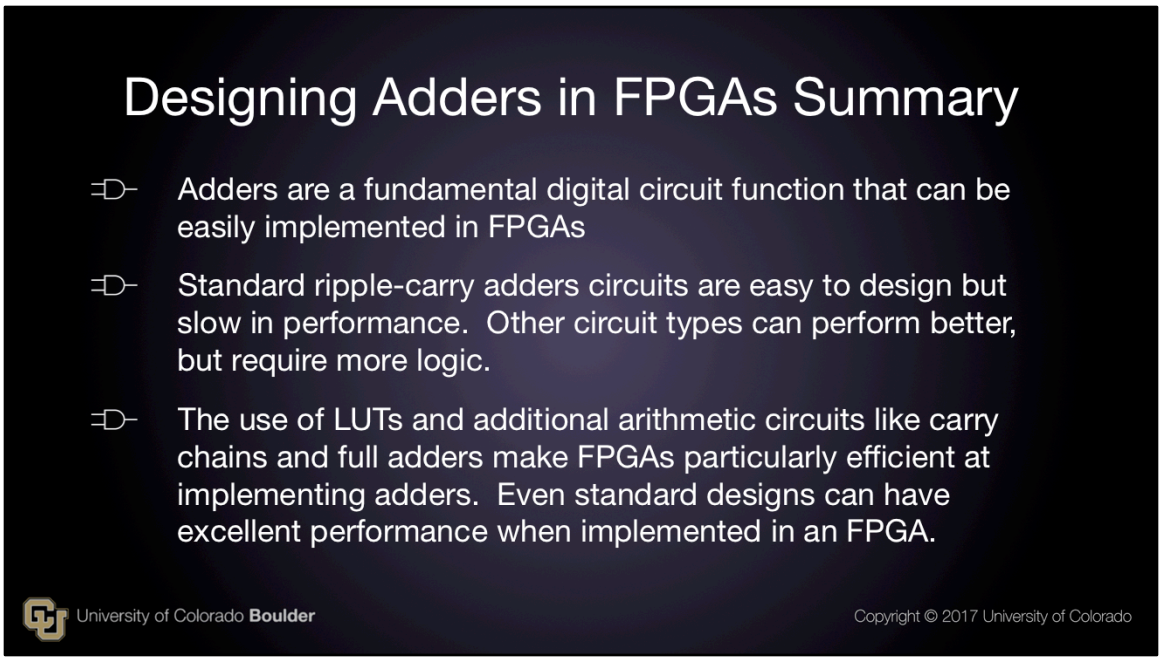
Although the use of Carry Look Ahead or other more sophisticated adder circuits may be beneficial in some applications, the presence of coarser grained FPGA architectures may make this unnecessary in many cases. The implementation you choose needs to be determined on a case by case basis, as there are many tradeoffs that may not be obvious at first. This is not unusual in FPGA design.

Altera

Figure 8: ALM for Cyclone V Devices



Furthermore, the Cyclone V logic cell, called an Adaptive Logic Module or (ALM) includes fast carry chains and full adder implementations as part of the logic cell, which makes implementation of adders particularly fast and efficient.



https://en.wikipedia.org/wiki/System_on_a_chip

In this video we have learned

- Adders are a fundamental digital circuit function that can be easily implemented in FPGAs
- Standard ripple-carry adders circuits are easy to design but slow in performance. Other circuit types can perform better, but require more logic.
- The use of LUTs and additional arithmetic circuits like carry chains and full adders make FPGAs particularly efficient at implementing adders. Even standard designs can have excellent performance when implemented in an FPGA.