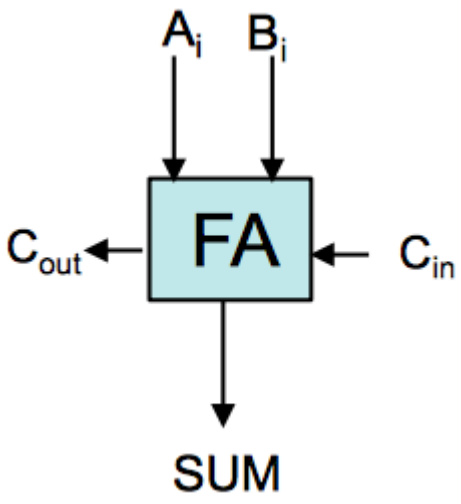
LE8.2

Bookmark this page

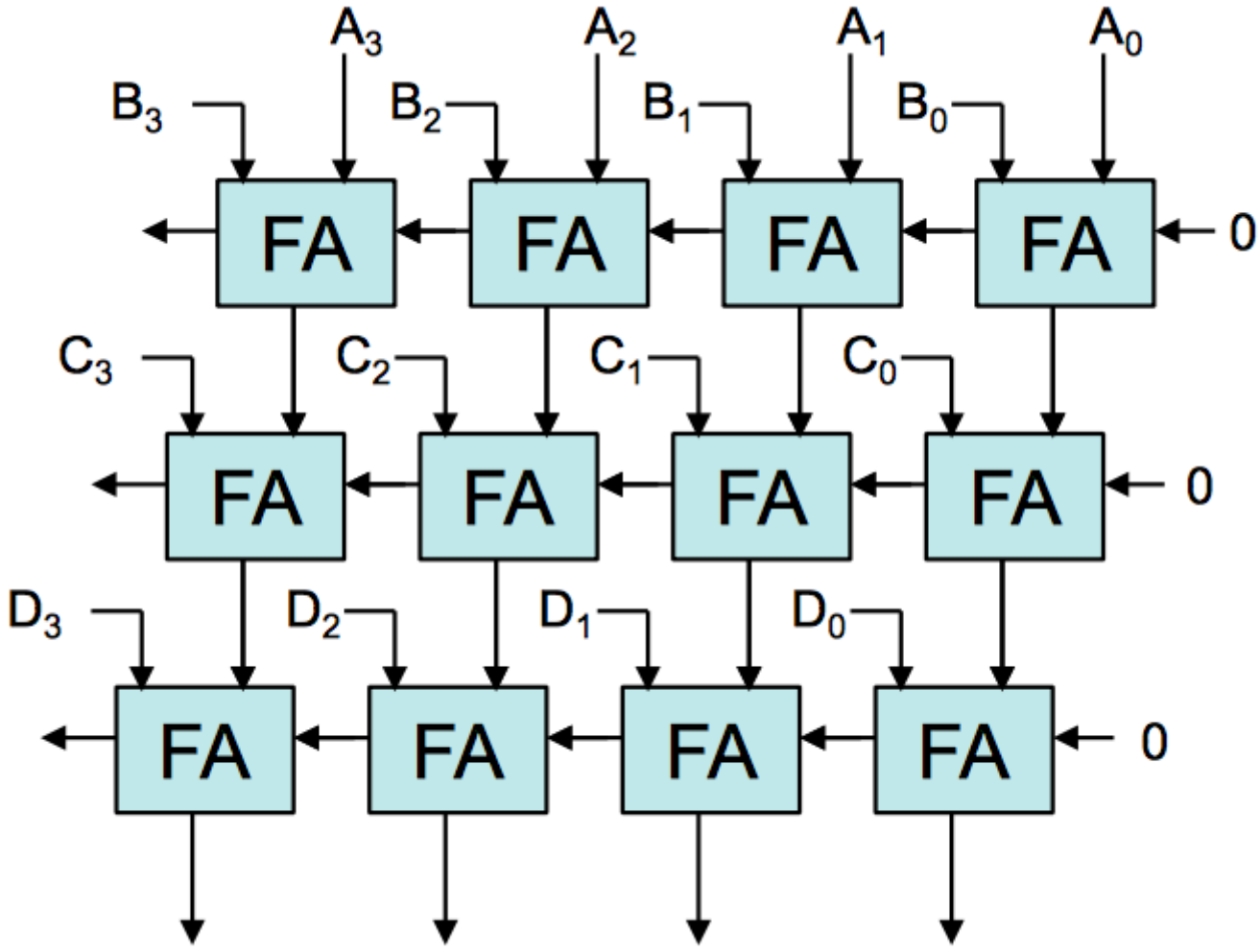
LE8.2.1 Pipelining and Design Tradeoffs

7 points possible (ungraded)

You've been hired by Addplex, a startup specializing in adders capable of summing an entire column of binary numbers as a single operation. Until you came on board, they had no engineers, but had a number of customers for combinational devices capable of adding N k-bit numbers together for various values of N and k. You need to quickly design products to satisfy this demand. You remember the full adder device you designed in lab 2, shown here. Fortunately, Addplex has a large inventory of these on hand.



You begin by tackling the problem of designing combinational circuit that adds four 4-bit binary numbers, producing a 4-bit binary result. Your approach is to combine three copies of the 4-bit adder you built in lab 2 and combine them as shown below:



You connect the low-order carry inputs to 0, and simply ignore the high-order carry outputs. Note that the four 4-bit inputs are designated A[3:0], B[3:0], C[3:0], and D[3:0].

You learn that the propagation delay of each full adder module is 1 nanosecond. What is the propagation delay of the above adder?

ns

You consider generalizing this approach to add N k-bit binary numbers for arbitrary values of N and k. You are particularly interested in the asymptotic cost and performance characteristics of the adder as N and k become large, and recall the $\Theta(\ldots)$ notation that abstracts out additive and multiplicative constants.

Using $\Theta(\ldots)$ notation, give asymptotic **latency** of an adder constructed as above capable of adding N k-bit quantities. Give expressions in terms of N and k.

$\Theta(\ldots)$

Using $\Theta(\ldots)$ notation, give asymptotic **hardware cost** of an adder constructed as above capable of adding N k-bit quantities. Give expressions in terms of N and k.

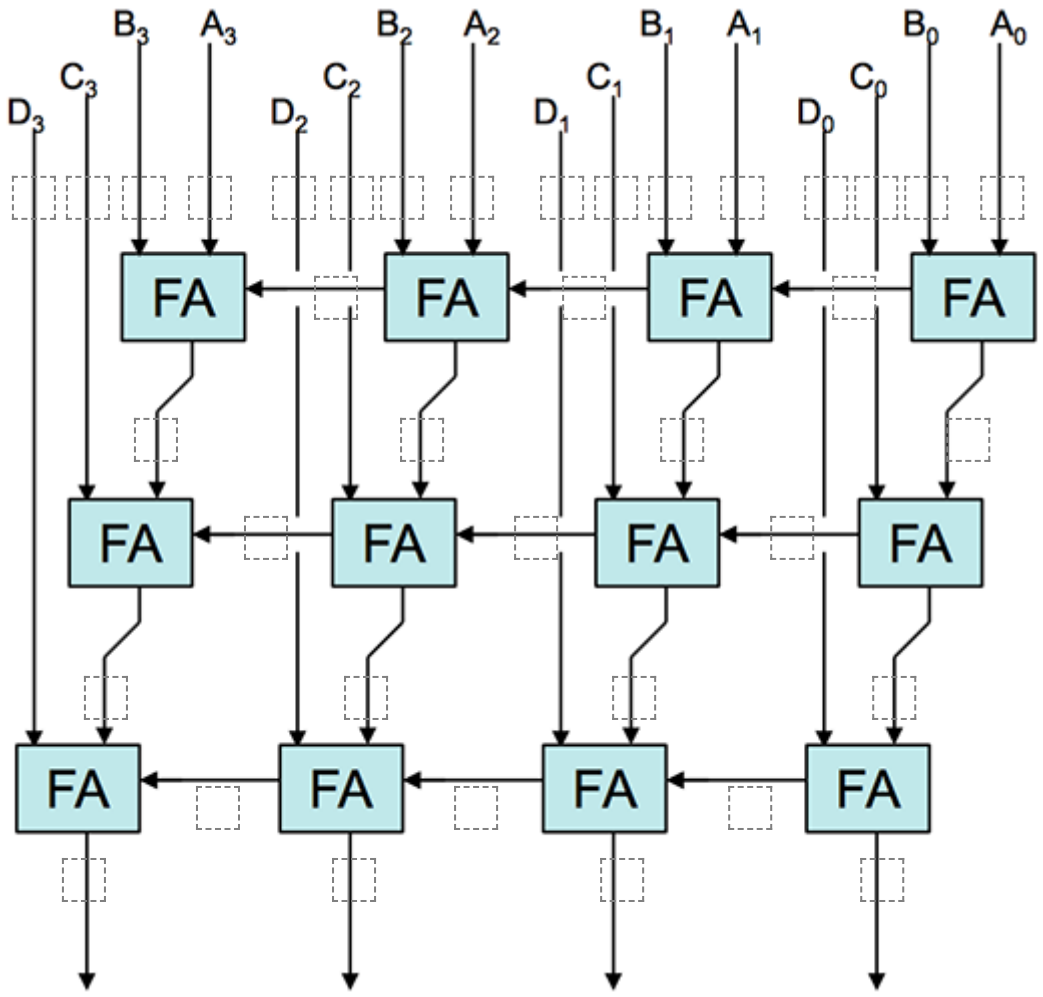
$\Theta(\ldots)$

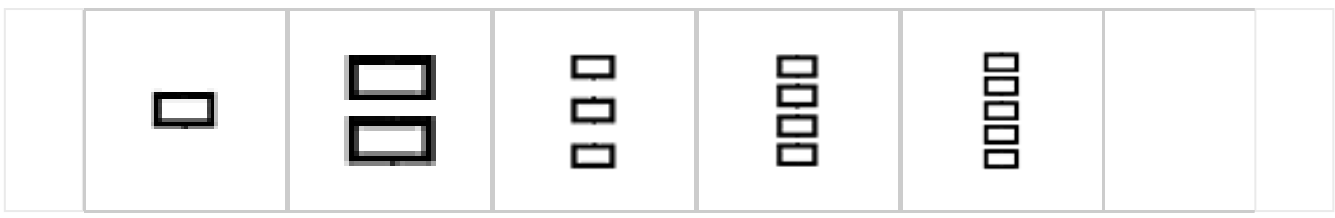
Calculator

Next, you turn to the problem of pipelining your adders for maximum throughput. You start by pipelining your 4×4 combinational adder, using ideal (zero-delay) registers inserted at strategic positions:

The diagram below has a single place available for registers on each wire. Indicate how many registers should be added on each wire by dragging the appropriate number of registers to each wire to produce a maximum-throughput pipelined implementation of the 4 by 4 adder.

You may consider printing the image and drawing contours to show the pipeline stage boundaries, and then counting how many pipeline stage boundries are on each wire to know how many registers to put on each wire. Use the minimum number of registers necessary to maximize throughput. Remember to put registers on all outputs. If a wire does not need any registers, do not drag any registers to that wire.





Finally, consider generalizing the pipelined adder to add N k-bit numbers.

Using $\Theta(\ldots)$ notation, give asymptotic latency, throughput, and hardware cost of a pipelined adder capable of adding N k-bit quantities. Give expressions in terms of N and k.

Latency $\Theta(\ldots)$:

Throughput $\Theta(\ldots)$:

Hardware Cost $\Theta(\ldots)$:

Submit

Discussion

Hide Discus

Calculator

Topic: 8. Design Tradeoffs / LE8.2

Add a Post

Show all posts by recent activity

- Grader too strict — 1
- Hardware cost of generalized adder — 5
  The explanation says "For every adder in the circuit, we need to add registers for every stage in the pipeline, making sure that each i...
- [Staff] How registers get counted in \Theta notation? — 2
  Hello, Please give us clear explanation of how registers get counted in \Theta notation for the last question about **Hardware Cost**
- Alternative pipeline — 2

Previous Next