University of California, Berkeley
College of Engineering
Computer Science Division — EECS

Fall 1999 John Kubiatowicz

Homework Quiz (HW #3)

September 29, 1999
CS152 Computer Architecture and Engineering

This quiz covers one of the problems from homework #3.
Good Luck!

Your Name:	
SID Number:	
Discussion Section:	

Total:	

Combinational Multipliers built with Carry Save Adders

In your homework, you learned about using Carry Save Adders to perform multiplication. Below we review using carry save adders to add four 3-bit numbers to produce a single 5-bit number:

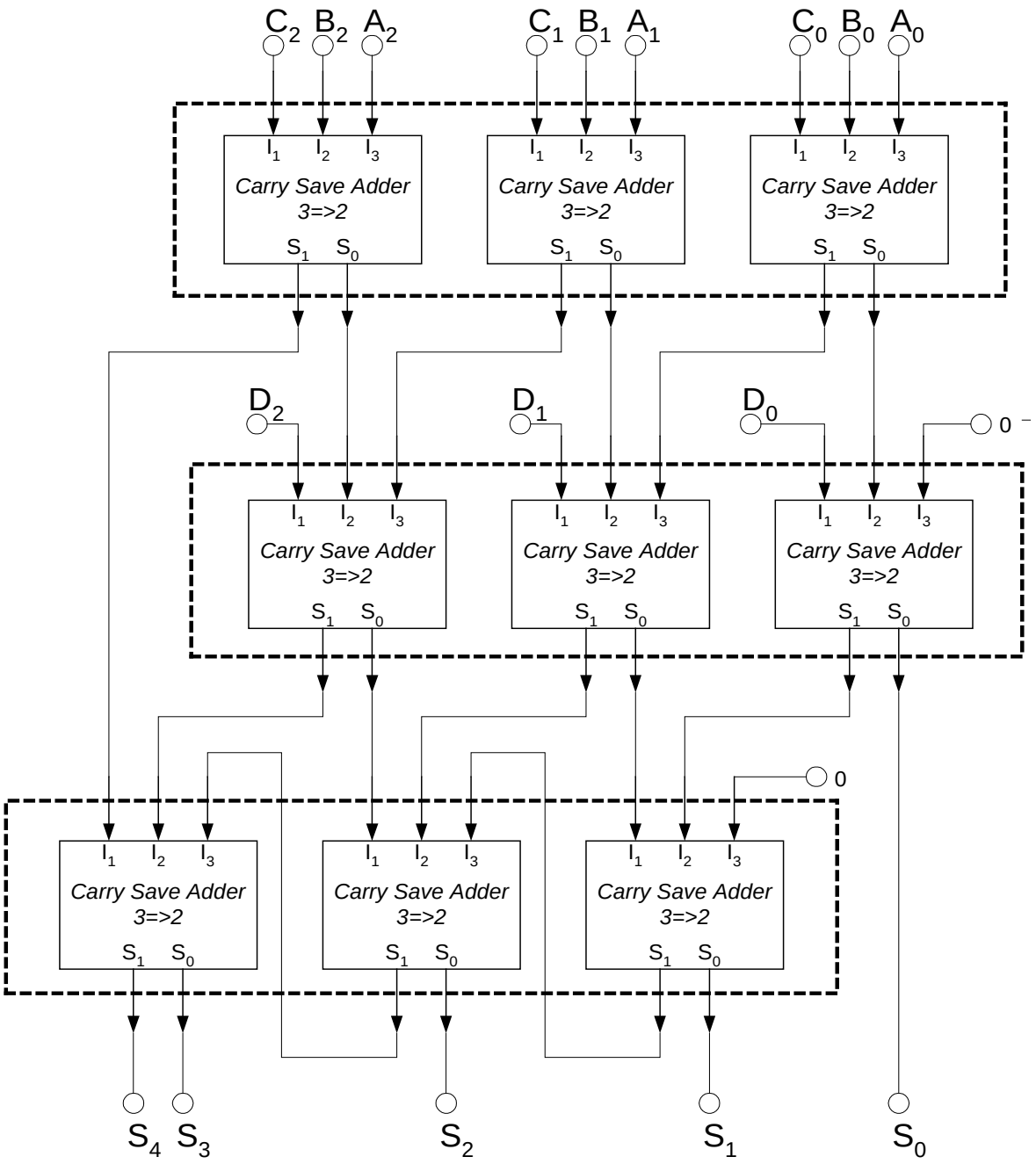

Figure #1: Adding four 3-bit numbers with carry-save adders.

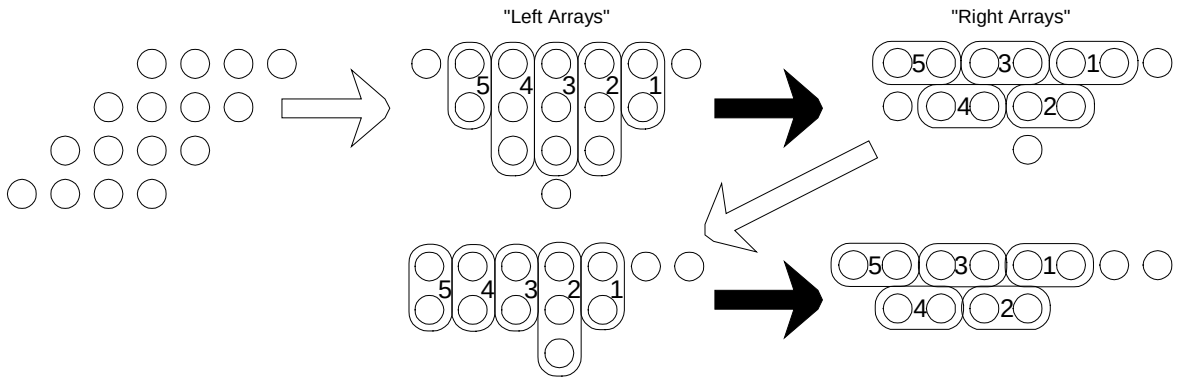

Figure #2: Carry Save Logic in a 4x4 multiplier

Figure #1 (previous page) shows a carry-save array for adding four 3-bit numbers together. You figured out how to use this type of addition to perform a 16x16 bit multiplier in your homework. One way to understand carry-save adders is that they add *columns* of bits together, just like people do on paper. Figure #2 shows this for a 4x4 multiplier. Each dot is a bit. The white arrows are renaming of bits, while the black arrows are carry-save logic.

The first array is the standard multiplication array. We rearrange the bits to get array 2. Each *pair* of arrays (called "left" and "right") represents one level of carry-save logic. The circled dots in a *column* of a left arrays are bits that are summed with a 3⇒2 adder. The circled dots in a *row* of a right array are the results. Thus, for instance, the two bits labeled "1" in array 3, are the result of adding the 2 bits labeled "1" in array 2. The long white arrow represents output of the first level going into inputs of the next. We continue until we have only two rows of bits, which will be added with a standard adder (possibly a carry-lookahead adder).

Representing each level by the maximum number of bits in any one column, we get a sequence that starts with the number of multiplier bits (4 in this case) and ends with the number 2:

$$4 \Rightarrow 3 \Rightarrow 2$$

1. Use a similar sequence to answer the following question: how many levels of carry-save logic would be needed for a 64x64 bit multiplier? The answer for 4x4 multiplier would be "2":

2. Use this number to *estimate* the maximum delay through this 64x64 multiplier, assuming that the last stage of addition uses a ripple counter, and that an adder requires 2 units of delay.