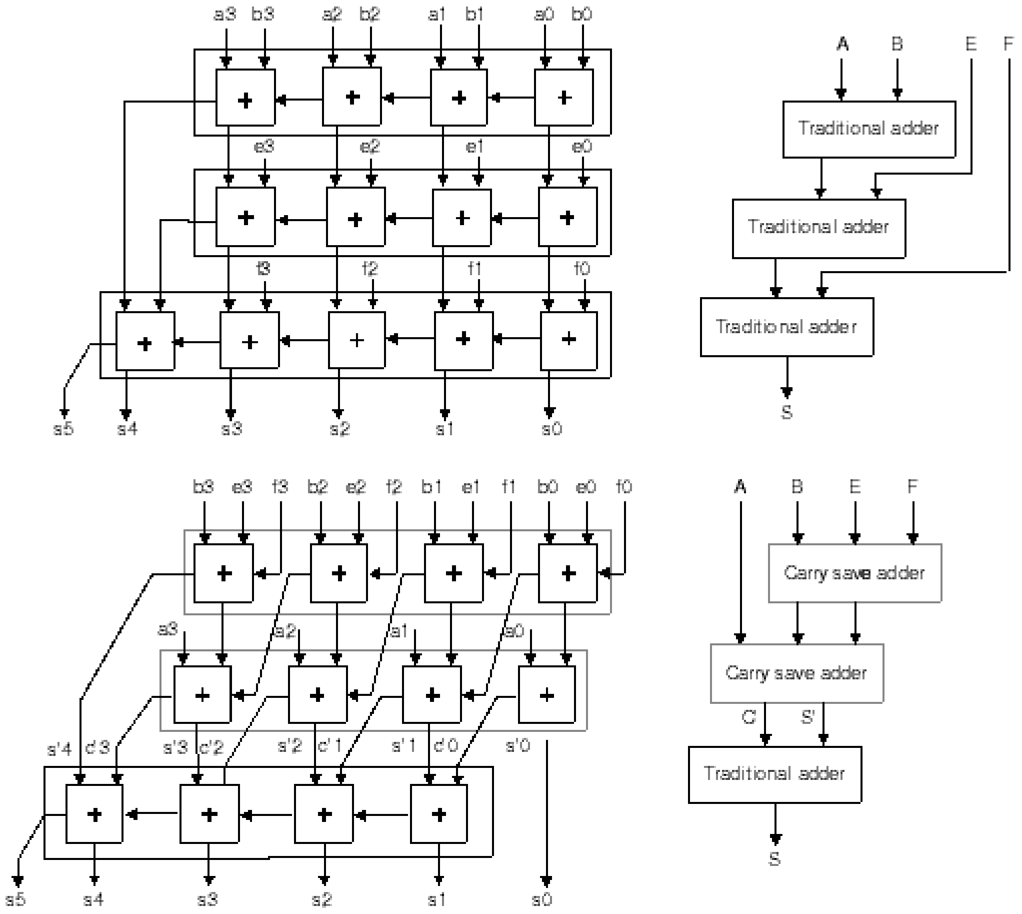
Homework 6

1. There are times when we want to add a collection of numbers together. Two methods can be used to add four 4-bit numbers (A, B, E and F) using 1-bit full adders, as shown in the figure. The first one uses the traditional method of ripple carry, and the second one, the carry-save adder, only uses ripple carry at the very last step.



   Once the three inputs x_i, y_i and c_i are available, it takes two gate delays for a full adder to generate the carry out c_{i+1}, and three gate delays to generate the sum s_{i+1}. For each method, find the number of gate delays needed to generate each one of the outputs from s_0 to s_5. (Hint: it is important to identify the “bottle neck,” or the critical path, i.e., the slowest path for generating each output.) How many gate delays does the second method save, compared with the first method?

2. Carry out 27 × 22 using the circuit shown in class. For simplicity, assume each register has six bits (instead of 16 or 32 bits, as in reality).

3. Carry out 17 ÷ 5 using the circuit shown in class and the restoring algorithm. Assume each register has six bits.

4. Do the same division 17 ÷ 5 using non-restoring algorithm and the circuit shown in class. Assume each register has six bits.

5. Use Booth’s method for signed binary multiplications to do (–22) × (–14). Assume you are doing this using the hardware including registers A, Q and M and the adder, etc., just the same way as (22) × (–34) is demonstrated in the notes.