Figure 4.11 shows an 8-bit adder formed by cascading eight binary full adders. Similarly, we build a 32-bit adder by cascading 32 binary full adders together. The carry into the 32-bit adder is zero, and the carry out will be saved in the carry bit.

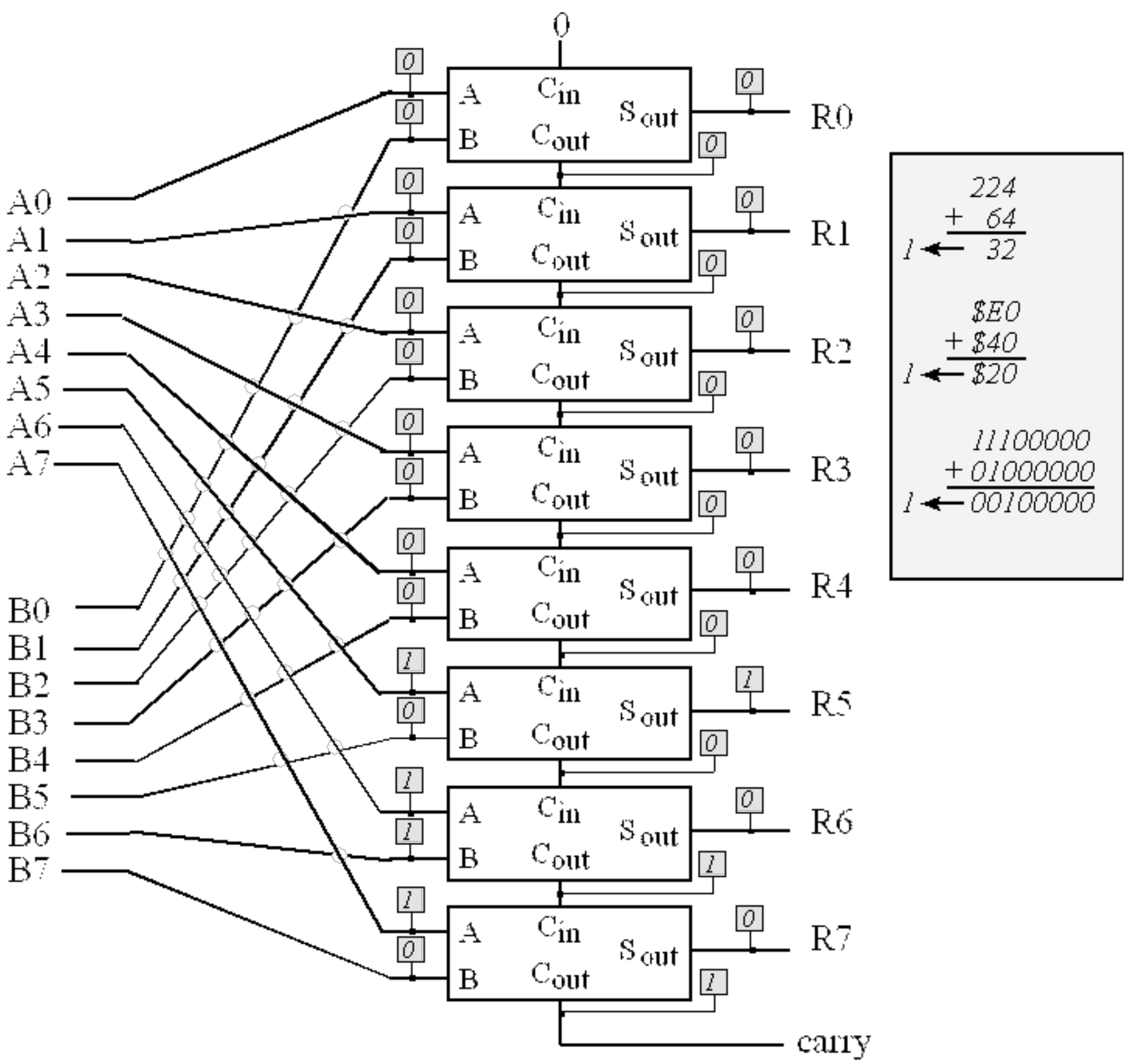

Figure 4.11. We make an 8-bit adder cascading eight binary full adders.

For an 8-bit unsigned number, there are only 256 possible values, which are 0 to 255. When we add two 8-bit numbers the sum can be any number from 0 to 510, which is a 9-bit number. The 9-bit result in Figure 4.11 exists as the 8 bits

R7–R0 plus carry.

We can think of 8-bit unsigned numbers as positions along a circle, like a clock. There is a discontinuity in the clock at the 0|255 interface; everywhere else adjacent numbers differ by ±1. If we add two unsigned numbers, we start at the position of the first number a move in a clockwise direction the number of steps equal to the second number. If 96+64 is performed in 8-bit unsigned precision, the correct result of 160 is obtained. In this case, the carry bit will be 0 signifying the answer is correct. On the other hand, if 224+64 is performed in 8-bit unsigned precision, the incorrect result of 32 is obtained. In this case, the carry bit will be 1, signifying the answer is wrong.

CHECKPOINT 4.9

If A has the value 100 (0x64) and B has the value 50 (0x32), what will be the value of the output (R7-R0) of the circuit in Figure 4.11? Also what will the carry signal be?

Hide Answer

100+50=150 with no overflow, so carry=0.

CHECKPOINT 4.10

If A has the value 255 (0xFF) and B has the value 2 (0x02), what will be the value of the output (R7-R0) of the circuit in Figure 4.11? Also what will the carry signal be?

Help

Hide Answer

255+2=257, so there is overflow. The result will be 0x01 and the carry=1.