Carry and Overflow: A Short Primer

Whenever two binary numbers are added or subtracted, the result may not fit into the same number of bits. For example, if two 8-bit positive numbers are added and they are both near the top of their number range, nine bits are needed to express the result. If a ninth bit is not available, an error condition must be flagged.

Adder/subtractors often provide carry and overflow outputs to flag such out-of range errors. Which of these flags is used depends on whether signed or unsigned operations are being performed. Overflow is only used with 2's-complement signed numbers, and has no significance in unsigned operations. Carry, on the other hand, is only significant as an error flag in unsigned operations, but is used to cascade both signed and unsigned operations.

Unsigned Operations

Separate out-of-range flags are required since the number ranges are different for signed and unsigned numbers of the same bit-length. In an 8-bit unsigned numbering scheme, for instance, the decimal values 0 through 255 are represented by the hexadecimal codes 00 through FF. This is illustrated on the circular number line shown in Figure 1.

The result of an unsigned addition is out-of-range when it is too far around the number line, such that the 255-to-0 boundary is crossed. Since crossing this boundary causes a carry output, carry can be used to flag the error. The output bits remain valid, but only as the eight less significant bits of a 9-bit result. The full result can be recovered by using the carry output as the MSB of the result. Unsigned subtraction is slightly more complicated, but is rarely encountered.

Signed Operations

When using 2's complement notation for 8-bit signed numbers, however, the hexadecimal codes are assigned differently. The decimal values -128 through 127 are represented by the codes 80 through 7F, as shown in Figure 2.

In this case, a result can be out-of-range in two ways; it can be too positive or too negative. If it is too positive, it is too far around the number line in the counter-clockwise direction, and the 7F-to-80 boundary is crossed. If it is too negative, it is too far around in the clockwise direction and the 80-to-7F boundary is crossed.

Consequently, for 2's-complement numbers, the out-of-range flag must be asserted whenever the 7F-to-80 boundary is crossed in either direction, and this is how the overflow output is defined. For unsigned numbers, however, the 7F-to-80 boundary crossing has no obvious significance.

In an overflowing signed operation, the output bits are again valid as the eight less significant bits of the 9-bit result. However, the MSB of the 9-bit result is not carry or overflow, but the inverse of the eighth bit. This relationship can be exploited to obtain a 9-bit signed result from an 8-bit adder/subtractor. XORing the output MSB with overflow provides the additional bit, as in Figure 3. ◆

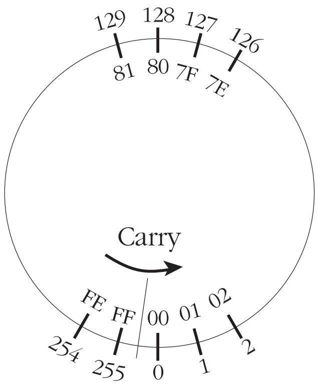

Figure 1: Unsigned Operations

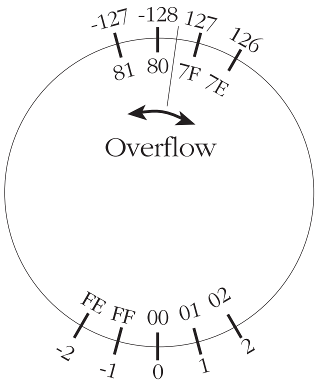

Figure 2: Signed Operations

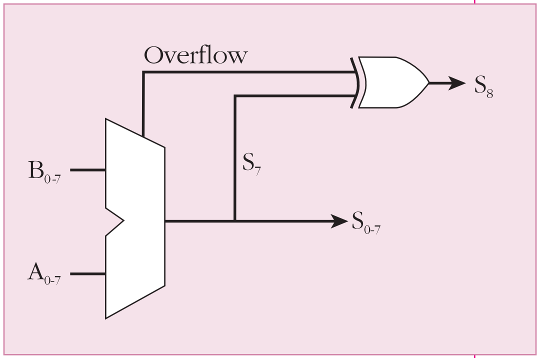

Figure 3: 8-bit signed operation with a 9-bit result.