

## ***Review Questions:***

# ***Digital Computer Internals***

© 1999 Charles Abzug

1. How can two Light-Emitting Diodes (LEDs) of contrasting colors (say, for example, purple and orange) be connected to an output latch such that the LED of one color lights up when the latch is a value of logical 1 and the LED of the other color lights up at a value of logical 0?

*Answer:* See Figure 5.4 in *Bebop Bytes Back*.

2. What is meant by the terms “common anode” and “common cathode” configuration?

*Answer:* See Figure 5.5 and the accompanying text on page 5-7 in *Bebop Bytes Back*.

3. Advanced texts on Computer Organization and Computer Architecture generally describe three kinds of shift operations, one of which has two variants. There are both the arithmetic shift and the logical shift (both described as such in *Bebop Bytes Back*), and also a circular shift. The circular shift comes in two varieties, one which includes the carry bit in the process (this is equivalent to the *Beboputer's* ROLC and RORC instructions), and the other which does not (not represented in the *Beboputer*). Be able to describe all of these instructions, and to recognize from the prior value of the operand and from the value of the results which instruction has been carried out.

*Answer:* See Figures 8.10 and 8.11 in *Bebop Bytes Back*, and accompanying text.

4. Explain what is meant when a flag is said to be “set” and when it is said to be “cleared”.

*Answer:* Set: its value is ‘1’.  
Cleared: its value is ‘0’.

5. Under what conditions is the *Z* bit set, and under what conditions is it cleared? Answer the same question for the *N* bit. For the *C* bit. For the *V* bit (or, as it is called in the *Beboputer*, the *O* bit). Does the answer to each of these questions vary for different kinds of arithmetic?

*Answer:*

Revised 25 Jan 99, 2350 hrs

## Review Questions: Digital Computer Internals

The  $Z$  bit is set when the last operation carried out produces a result whose value of Zero.

Note that there is only one representation of Zero in Unsigned numbers, Two's Complement numbers, and in Saturation Arithmetic. The conditions for the  $Z$  bit are therefore identical for Unsigned numbers and for Two's Complement numbers. Because of the special circumstances attendant upon the use of Saturation Arithmetic, no changes occur in any of the Status Bits when a Saturation Arithmetic operation is carried out. Ones'-Complement numbers include two zeroes, one which is called "positive zero" and is identical to the only zero of Unsigned and of Two's-Complement numbers, and the other which consists of all '1's and is referred to as "negative zero". Signed-Magnitude numbers also have two zeroes, the "positive zero" which is the same as that for One's-Complement, and the "negative zero" which consists of a single '1' followed by all '0's. Thus, in principle the zero bit is defined identically for four of the number representations, but the implementation may be different because the zeroes are differently defined in different number representations. And Saturation Arithmetic never affects any status bits.

The  $N$  bit is meaningful only for Ones'-Complement, Two's-Complement, and Signed-Magnitude numbers, where it is identically defined as being equal to the value of the leftmost bit of the number. Both Signed-Magnitude and Saturation Arithmetic have no representation of negative numbers, and therefore in these cases there is no meaning to an  $N$  bit.

The  $C$  bit represents a carry-out from the most significant bit of the magnitude of the number. For a number represented in  $n$  bits, in Unsigned Numbers, Ones'-Complement, and Two's-Complement numbers this refers to the carry-out from bit position  $n - 1$ , while for Signed-Magnitude numbers where the sign bit never figures into the addition, the carry-out comes from bit position  $n - 2$ . Of course, for Saturation Arithmetic there is never a carry-out because any would-be carry-out is handled internally by the hardware and is converted into a register-content of all '1's, while there is no significance to any status bits in Saturation Arithmetic anyway.

The  $V$  bit (*O in BeboP Bytes Back*) signifies that the result of the last operation does not fit in the register (i.e., it requires one more bit to represent it accurately than is available). The abstract definition is identical for all five forms of arithmetic, although in terms of implementation there are differences. For both Unsigned Numbers and Signed-Magnitude Numbers the  $V$  bit is always identical to the  $C$  bit (although note that the implementation of the  $C$  bit is different in these two cases). For both Ones'-Complement and Two's-Complement numbers the  $V$  bit is set when the carry-in to the MSB (Most Significant Bit) has a different value from the carry-out from it. And of course for the case of happy-go-lucky Saturation Arithmetic there is no such thing as an overflow, and anyway no such thing as status bits, either.

6.

7.