No. 61, Dec 1998

THE COMPUTER CORNER

Parity: Odd But Necessary

- by Stan Kaplan, WB9RQR
105 Martin Drive
Port Washington, WI 53074-9654
(414) 284-9346
WB9RQR @ N9PBY.EN63BI.WI.USA.NA
skaplan@mcw.edu

Many of you have probably seen SIMMs (Single Inline Memory Modules), which provide the Random Access Memory (RAM) for your computer. SIMMs consist of a small circuit board about 4 inches long by ¾ inch wide. Typically they contain nine chips permanently soldered to the board, all in a row. Eight of the chips are involved in storing data; the ninth is for parity.

If you are at your keyboard composing a letter, the characters you type are echoed to the screen, but they also go into RAM. When you later save the file to disk, the contents of RAM are read back out and written to the disk. Parity (or more correctly, parity checking) is a scheme to check that the data written into RAM is the same later when it is read back out. Why do you need parity? Because errors abound in modern computers, no matter how well they are built.

As you might expect, errors can occur when reading data from or writing data to the magnetic surface of a hard or floppy disk. But errors can also occur due to the data being changed, even when no read or write operations are taking place. Cosmic rays are constantly bombarding us from outer space, and if they strike a magnetic domain on a disk, they can change a 1 to a zero, or the reverse. The same sort of change can occur inside a RAM chip if it is struck by a cosmic ray. It has been estimated that the average user of a modern computer can expect about one error per month just due to a cosmic ray striking something inside their computer. Thus, error checking is prudent, a Good Thing. While it cannot prevent errors, or even correct them, it can alert us to the fact that an error has occurred. We can then take steps to minimize the consequences of the damage.

Well then, how does this parity business work? In the June 1995 column (No. 21, SERIAL AND PARALLEL), I used a favorite analogy to describe how data is represented in a computer. The data is stored in 8 bits (one byte), and we can think of these 8 bits as a row of eight simple ON/OFF switches - all single pole/single throw. Let us suppose that you are working in your word processor and you type the capital letter A. Here is how the letter A is represented:

Switch number	7	6	5	4	3	2	1	0
Bit Value	0	1	0	0	0	0	0	1

Strangely, computer nerds start counting from the right instead of the left, as you can see in the switch number column. Another nerdy characteristic is that they call the first thing in any series of things the "zeroth" thing. Thus, the first switch is switch number zero, the second is switch number one, and so on. A bit value of 1 means the switch is on; a value of 0 means the toggle has been flipped to off. As you can see then, switch number zero is on, the next 5 switches are off, the next is on and the 8th (switch number 7) is off. This pattern of the on/off position of 8 switches represents the capital letter A. No other pattern represents A; it is unique. Another example, the lower case a, is shown below:

Switch number	7	6	5	4	3	2	1	0
Bit Value	0	1	1	0	0	0	0	1

So, here is how parity works. When you type the A, 8 switches (one in each of the first 8 chips on a SIMM) are set to the bit pattern shown in the first example above: 01000001. The system then counts the number of ones present. There are two. Two is an even number, so the system sets a switch in the 9th (parity) chip to 1. Now there are three ones, an ODD number. If you next type a lower case a, the next group of switches are set to the pattern shown in the second example: 01100001. The system counts the number of ones present. There are three, an ODD number, so it sets the parity chip bit to 0, thus maintaining an ODD number of ones. In every case, the system flips the parity toggle switch so as to make an ODD number of ones (hence the tongue-in-cheek title of this article).

Later, when the data is read back (such as when you save the file to the disk), the system checks the number of ones. If the nine-bit byte contains an even number of ones and the parity bit has been set to zero, there must be an error. Similarly, if the number of ones is odd and the parity bit is 1, there must be an error.

What happens if there is an error? The system generates a Non-Maskable Interrupt (NMI), which is a techie term that essentially means the Central Processing Unit is told to unequivocally quit doing whatever it was doing, RIGHT NOW! The system may then clear the screen and offer a menu: S)hut off NMI. R)eboot. Press any other key to continue.

If you ever see that message, press the space bar to continue. Then save your work, but don't save it with the same file name you were using before. If you use the same name, you may write possibly corrupted data over your older, but good, copy of the file. Save it using a new name, and check it later to see that it is good. Then reboot and see if everything seems normal. If it is, then perhaps it was only a cosmic ray coming through your roof and striking your computer. On the other hand, if it happens again and often, you may well have a bad memory chip or even a bad motherboard.

Now get this. In the mid 1990's, computer manufacturers began making motherboard that did not check for parity. Having thus disabled parity checking, they were able to install 8 chip, non-parity SIMMs, which were 10 -15% cheaper than the parity checking versions (anything for a buck!). They did not advertise this, and the public was unaware that any change was made. Although a number of these manufacturers have now gone back to parity checking motherboards and SIMMs, there may still be some systems that are being sold without it.

The message: if you are purchasing a new computer, get a statement in writing that both the motherboard and the installed SIMMs will support parity checking. Don't buy without the statement. If you do, you are likely getting a non-parity checking system, and will never know if parity errors are occurring. It is not good enough to just be sure parity memory is installed. A non-parity checking motherboard will work fine with parity SIMMs installed, but parity will not be checked. Happy computing!