No. 37, Dec 1996

SPEED #2: The Address Bus

Last month we saw that the data bus has some effect on computer speed. After all, 32 hams pounding keys can send more information than 8 in a given amount of time. However, there is another bus inside the computer that has an influence on speed -- the address bus.

The address bus is a set of wires, completely independent from those of the data bus, that are used to control where in memory the data should be sent for. You see, it is not good enough just to have sufficient wires to send data in the computer. You must also have a way of telling the CPU *where* to send it.

Remember the table last month? The XT had a *data bus* width of 8 (wires), the 286 had 16, the 386DX and 486DX had 32, and the Pentium had 64. Well, suppose our 8, 16, 32 or 64 hams sitting at keys have their wires not just going to one bank of light bulbs in a single building somewhere else in the city. Rather, their wires go to a big switch bank that can send their data to any single building located anywhere in the world. The address bus controls that big switch bank.

The address in the computer's memory to which the data is sent is a unique number, just like a house number. However, because computers use a base 2 numbering system (binary), a single digit can only represent two unique addresses (0 or 1). Two digits can represent four unique addresses (00, 01, 10 and 11). Three digits can represent eight unique addresses (000, 001,...), four can represent 16 addresses (0000, 0001,...), and so on. The math is pretty simple. The exponent indicates the numbers of data bus wires: $2^3 = 8$ unique addresses, which is the same as saying 2 x 2 x 2 = 8.

Well, it turns out that an XT has an address bus of 20 wires. To calculate the number of unique addresses 20 wires can provide (2^{20}), multiply 2 x 2 x 2 x 2 x 2 ... (20 twos, in all). The answer is 1,048,576. In computer lingo, 1,048,576 is one megabyte (since one kilobyte = 1,024 bytes). So a 20-bit address bus can send the data from 8 hams pressing keys to any one of 1,048,576 unique buildings scattered around the world. Somewhere, depending on which single building is selected, each of those 8 light bulbs will light up or not, making a unique pattern that can be decoded by anyone looking at them as a lower case letter a. Which building gets the data is the responsibility of the address bus.

What about the other Central Processing Units? The following table shows address bus width (number of wires) for several CPUs, along with a very impressive capability to address unique spots in memory.

ADDRESS BUS WIDTH AND MEMORY ADDRESSING CAPABILITIES OF SEVERAL CPUs

CPU	**BUS WIDTH**	**BYTES**	**KILOBYTES**	**MEGABYTES**	**GIGABYTES**
XT	20 bits	1,048,576	1,024	1	
286 and 386SX	24 bits	16,777,216	16,384	16	
386DX, 486 and Pentium	32 bits	4,294,967,296	4,194,304	4,096	4

The population of the world back in 1987 was 4,975,000,000. Interesting to think that the old 386DX sitting on your desk could address over 85% of all the men, women and children living at that time! The P6 (not shown in the table above) can address 64 gigabytes of unique spots in the computer's memory. Mind-boggling!

Well, then, what does this all have to do with speed? A good definition of speed in the computer world is the rapidity with which a machine can do a task. Given two otherwise equal computers, the one with a single megabyte of memory will take longer to multiply two 12-digit numbers than the one with 16 megabytes of memory. And that computer with 16 Megs can only use the full 16 megs if it has a 24-bit address bus. Similarly, when you are word processing a large document in an XT, it may not all fit into memory at one time. If you are working on Chapter 5 of your book, and you need to move back to Chapter 2, you may well have to wait for the word processing program to load Chapter 2 from the hard drive into memory, before you can begin to work on it. Not a problem in a 486 with lots of memory. It is likely that all the chapters will be loaded into memory when you first start your word processing session.

So the address bus is like a telephone book, with everyone's address. When the CPU wants to send some data into memory, it depends on the address bus to make the right connections so the data is routed to the correct spot, and so it can be found later, if needed. Simple enough. You can't send mail to someone if you don't have the address!

I really like the way computer guru Scott Mueller described it: "The size of the data bus is an indication of the information-moving capability of the chip, and the size of the address bus tells you how much memory the chip can handle." While both of these factors influence speed, that is still not the entire story. Some more next month. Happy computing!