THINKING DIGITALLY IN BINARY IS JUST PLAIN DIFFERENT

StentorVox dot BlogSpot dot com

Well, thinking digitally in binary is just plain different.

It is easy to picture a player piano roll with lots of holes, corresponding to a player piano:
Each hole represents a key press.

That is parallel transmission: 88 keys and 88 holes that tell each of those keys to make music as they turn on and off the musical mechanism. Well, actually, some player pianos have different numbers of keys, but there is a hole for however many keys there may be.

And it's also easy to think of Morse Code.

But that is NOT parallel. When Samuel Morse created a new alphabet to transmit messages, he realized it would not be easy to sell his invention if it required a parallel transmission of 26 separate wires designed to turn on and off individual keys to an alphabetic typewriter keyboard.

So he did the next best thing. Instead of wiring together individual, parallel, wires, he would send individual packets of messages on a single wire one at a time, in a serial stream. So the telegraph operator would send a clackety-clack down the wire, but some of those clacketys were long (dash) and some of the clacks were short (dot).

Therefore, a message comprising a series of dots and dashes, short and long tones, might sound like:
dit-dah (pause) dit-dah-dit-dit (pause) dit dit-dit (pause) et cetera.

His entire message was based on only TWO signals. This is a simple way to use binary code, a good start, but far from perfect.

For one thing, some letters of his alphabet were only one dit or dah long (.) and (-) while others may comprise two, three, or four blips, but it was a start. From here people could begin to understand there could be a way to send a series of discrete SERIAL messages down a single wire, rather than sending PARALLEL messages down a whole bunch of parallel wires.

So people began to string up miles of these serial wires from town to town on any vertical poles or trees or whatever available lumber might be handy. Sometimes the pickings were sparse, so it looked rather futile. It looked more like a grapevine than the straight, well-manicured utility pillars today that carry wires along our neighborhood streets and highways.

People who received an early message often said they "heard it through the grapevine."

When the manufacturers and musicians met to design a MIDI protocol before the end of the past century, they decided the digital idea was good as far as it went. Two choices: could be *up or down*, *heads or tails*, *black or white*, *on or off*, *dot or dash* ... well they didn't go for that one ... they decided on *On or Off* because it was more, well, computerish ... (we've all see the Star Trek re-runs, and where would those sets be without all their computer blinking lights that flashed on and off, but I digress) ...

Actually, binary is a great way for machines to count (and computers are machines that process signals, and MIDI is a computer signal), and signals can simply be ON or OFF ... TWO choices (so that's BI-nary, get it?)

I mean, when I learned to count, it was five fingers, then the other hand, for a total of ten fingers and that isn't hard to do. If I wanted numbers worth more, I could take off my shoes, and the little piggy that went to market became number eleven, and the little piggy that had roast beef was number thirteen, but here was another problem.

A computer wants to count binaural numbers: one choice among two, not a barnyard.

The other problem with counting was that we only had enough numerals to represent a single character for nine digits, and then we had to improvise by adding a second column.

(continued on next page)

That is to say, the system is only good from one to nine ... er, that is to say (in Arabic Numerals, named for the people who invented this system of counting) ... from 1 to 9, not counting the 0.

Not to be outdone by the Arabs, the Romans invented their own Numerals using alphabet characters like X's and V's, C's and I's ... but that's another story that has nothing to do with the way computers like to count, so I'll just drop it for now.

Back to *Arabic* counting: If you want a higher value than 9, you can use a "zero"

... oops, didn't mean to use *Cardinal* number (that's what we call the names of those numerals when they are spelled out one, two three, zero, et cetera).

Not to be confused with *Ordinal* numerals (First, Second, Third - or 1st, 2nd, 3rd), either ... best get back to *Arabic:*

Rather, we simply added a 0 ... and simply recycled one of those other numbers from 1 to 9, adding a whole new column.

One single column = ten's.

Adding an additional column works, as well. Two parallel columns = hundred's, etc.

(Well, actually, someone figured how to use something called hexadecimal to exceed the limitation after the "9" with single alpha characters that only required one column, but no time for that now).

Didn't mean to ramble on so. Just to say that there are several ways to count, and when the MIDI protocol was invented, they decided on Binary, and in a serial manner, marching along a single wire one at a time.

The Binary system is great for computers, because it doesn't need a new column to display something higher than nine, er ... 9.

If you have two choices (like flipping a single coin) you can have two choices:

HEADS or TAILS

.
But it gets more powerful, and quicker for those computer lights that flash on and off.

If you have **TWO** coins, you have FOUR choices:
Heads - Heads / Tails-Tails / Heads-Tails / Tails-Heads

When we try flipping **THREE** coins, we get even more choices with only three columns.
Heads-Heads-Heads / Heads-Heads-Tails / Heads-Tails-Tails /
Tails-Heads-Heads / Tails-Heads-Tails / Tails-Tails-Tails

Flipping **FOUR** coins ... This time, instead of saying "heads" let's say "0" and instead of saying "tails" let's say "1."

We how have sixteen choices:
0000 / 0001 / 0010 / 0011 / 0101 / 0110 / 0111 / 1000/
1001 / 1001 / 1010 / 1011 / 1100 / 1101 / 1110 / 1111

Well, so far, DECIMAL counting (the ten fingers from 1 to 9 plus a 0 with something in front of it) seems to be easier, because after nine, you have to add a separate column, but in Binary it takes four spaces.

When we flip **EIGHT** coins we have the potential of many more choices (a total of 156), and it gets better when 16 bits (that's what we nickname these ***BI***nary digi***TS***), we get a whopping 65,536 discrete choices.

(continued on next page)

And it is all from a total of just TWO CHOICES:
EITHER/OR,
ZERO or ONE
OFF or ON

Each message in a MIDI signal going down a serial wire is composed of a specific length (unlike Mr. Morse's dit-dah messages, comprised of one, two, three or as many as four separate tones to represent a character of the alphabet).

So these computer numbers are equivalent to many more choices that the original piano player roll we started out with, where one hole would cause middle C to play, and another would play an F or G.

That two-choice BINARY code is simply made up of digital signals.
Eight Bits is called a BYTE.
several BYTES can be joined together to create a digital WORD.

Now, instead of wasting the potential of digits to merely control 88 discrete piano keys, we have the potential of MIDI bytes and words to send all kinds of signals to a computer, or synthesizer processor.

Whenever you press a key, it sends a byte down the serial cable to say, "Hello, a note is pressed, stay alert and I'll tell you which key it was!"

Then in subsequent arrangements of these bits, the note is defined, spelling out which octave on the keyboard, etc., as well as much other information about that note.

In the early days of MIDI many of us purchased simple keyboard kits with just a single octave to solder together to send MIDI signals. Of course, we played more than an octave by simply designating which octave the notes should play on the playback equipment ... Try THAT on a player piano roll!

So a long answer, I know. We can overflow with praise of this musical tool. We use it for keyboards, EWI wind instruments, drum triggers and even to control theatrical lighting systems.

I wish an organist had been there in the design stage when a protocol was being formatted for the Musical Instrument Digital Interface (nicknamed MIDI), because it was poorly designed for combining and layering sounds on top of each other as organs like to do with all their drawknobs, stoptabs, pistons and studs. MIDI began with just 16 channels of MIDI that could be played one at a time.

But it was a start, since a universal language was established to play the same sounds on a variety of different synthesizer keyboards, since they were established with specific instrumental sounds on specific channels. Of course, a Steinway piano on one keyboard will sound different than a different piano on a cheaper keyboard, but it will always be a piano and it will always be on the "piano" channel.

It is referred to as a MIDI "Protocol," as it does not actually "create" the actual sound, any more than a punched hole in a player piano roll of paper "creates" the sound the piano makes ... it just manages the sound.

I was discouraged when I tried searching for compatible organ software and hardware solutions, since today's different inventors had designed incompatible programs, but there is hope as more users enter the field. I plan to eventually merge pipes and electronic MIDI on a simple instrument, where the MIDI signal will operate a pipe magnet instead of a sound font.

Thanks to computerized technologies of today, programs like Hauptwerk, Artisan, J-Organ, MidiTzer and other software packages … wow ... bring hope for us theatre organ folks, after all! Now if we can just pull theory into practice!

Paul E. Kealy

StentorVox dot BlogSpot dot com