

# The

## "Kill a Byte"

## Standard

Hal Walker  
National Multiplex Corp.  
3474 Rand Ave  
South Plainfield NJ 07080

*Hal Walker has some "interesting" comments concerning his competition (as a matter of fact, we'll all probably get sued). As you read on there's a very good chance you'll get the impression from Hal that he is somewhat biased toward his own unit. Perhaps we should set these pages aside for "The Cassette Situation Forum" and give the opposition a chance to reply. Anyway, I'm sure you'll find it interesting reading. (Furthermore, if you'd like to build one of his units, the construction plans and schematics are in this and an article to follow.) — John.*

**W**ayne Green and I were standing around looking at the great proliferation of equipment at the Atlantic City Show. Wayne said, "We're trying to figure out what should be a standard for data interchange among computer hobbyists; everyone seems to be going in different ways." I agreed and said, "It's because almost everyone is overlooking the only really acceptable standard." Wayne asked why and I put my foot in my mouth, promising to write an overall survey on recording methods and interfaces.

### Some Background

Recording on tape (or wire) goes back almost fifty years. At first they recorded clicks and then they progressed to voice. With the coming of the computer age, computer makers adopted the nonaudio "bang bang" zeros and ones. We at National Multiplex manufacture non-return-to-zero (NRZ) type recorders for the trade. So do almost all the other profes-

sional recorder houses. IBM, Univac, Honeywell — all the big boys — started using NRZ in the fifties and still do. NRZ recorders do not use the same recording method as audio machines, hence must be considered as totally different machines.

NRZ has some drawbacks, one of which is that it has a dc component. You must keep track of this "marking" condition during silent periods or your first character will be incorrect. It is also subject to distortion when long strings of zeros or ones occur together — all because of the dc component. NRZ is not self-clocking; therefore, a capstan drive and extremely accurate speeds are a necessity. It is not easily used with data error detection schemes and tape requirements are critical. Originally NRZ was limited to 800 bits per inch, but better heads and tapes have extended this to the 1600-2000 bits per inch range.

Next came modified non-return-to-zero (NRZI). NRZI still suffered from dc component troubles, but it did allow easier error checking and was a slight improvement. In NRZ recording there is a transition on the tape at

the start and end of a bit. NRZI makes its transitions in the center of a bit. Because both NRZ and NRZI saturate the tape, there is some loss of density of recording and ways to circumvent this were devised.

Coding methods under various names appeared, but they all seemed to converge into one method known as phase encoding (PE). Now wait just a moment. Despite what you may have seen in ads, or print, the so-called Kansas City standard (sometimes known as the Byte standard) is not true phase encoding; it is FSK, which we will discuss later. However, it could be called redundant phase encoding which we will also discuss.

Phase encoding can operate from unsaturated tape (audio recorded), thus effectively doubling the bit density, but it requires twice as many transitions, thus halving it again so you are back where you started. It has two big advantages: 1) It has no dc component, and 2) it is self-clocking. It is the proposed ANSI/ECMA standard and is the method used by the big boys on cassettes.

This article and a following one will tell you how to

build your own PE 2400 interface. While basically intended for NRZ recorders, phase encoding can be used on audio recorders if the baud rate is kept high enough. A hobbyist cassette board using it is marketed as the Tarbell interface. Several other interface kits also can use it.

### Cassette Problems

What about the present hobbyist systems? Well, I've heard enough discussion and cussing of most of them to fill a book. Most of the cussing is based on some good fundamental underlying problems. One of the problems is that standard cassettes are subject to uneven tape speed due to binding in the cassettes themselves and due to extremely critical clearances mechanically in the drive unit. Just the slightest amount of overpressure on the back of the cassette can cause a jerking motion that ruins the whole data string. This applies to your audio cassette also.

To get good data we have to spend a great deal of time adjusting these clearances and pressures. The same effect can be caused by the cassette itself. Cassettes unfortunately are not precision devices. They are molded plastic, and they do catch internally. Tape guides and pressure pads are critical items and you must often try several brands to get the right combination.

Not all cassettes are suitable for data because of dropouts on the tape and assorted mechanical binds. For example, too much pressure on the pressure pad in the cassette causes a skipping action that causes lost bits.

Not all audio tape is good for data. We had to buy dozens of samples of the available audio tapes and test them for data use. In the audio field only Memorex MRX2 and Scotch HE are really good. The *do not use* list includes Scotch Classic, Memorex Chrome, any other chrome or ferrichrome,

Maxell UDXL (but you can use UD) and Realistic. These *do not use* tapes are OK for audio, but not for data. Certified data tapes are best of course.

Two more things about cassettes. Use short lengths if possible to avoid binding. C30 is always better than C60, and you should avoid using the first 10 seconds of tape because that's where the stretches and glitches are.

Because of the various problems encountered with Philips cassettes, there is presently an industry wide changeover underway to the 3M Data Cartridges. These are free from most cassette problems, but are more costly to the user. Our own product line will be fully converted by Christmas of '76.

### The MITS ACR Unit

When MITS started the amateur computer craze, I managed to buy an early model with a weak power supply and an ACR cassette interface. The power supply wouldn't carry 16K of memory and the ACR wouldn't read their tape. Being of the "bigger hammer" type, I cured the power supply problem by adding an EICO battery charger to the 8 volt bus, then I threw ACR away.

The ACR was retrieved by the local computer store man who finally gave up on it and went to Tarbell cassette interfaces. I'm hurt that he hasn't even tried my cassette units, but you can't sell them all. He is stocking my new data cartridge units though.

In fairness to MITS, they did correct the power supply problem, but they have not really corrected the ACR problem.

The ACR uses a phase locked loop (PLL) to detect frequency shifts (FSK) and thus determine a zero or one.

I have manufactured a model LP-7 audio/visual machine using phase locked loop ICs to control slide switching and automatic stopping for several years. Both I and my

customers wish I had never heard of the things for use as tone detectors.

The ACR system is based on frequency shift between 2025 Hz and 2225 Hz for discrimination of 1s and 0s. At 300 baud each bit has 7 cycles. One or more of these is lost in the transition (change of frequency can't be instantaneous) and probably two are lost in lock time delay. PLLs have pull in times amounting to several cycles so that there is a delay in recognizing a one or zero. Since each cycle is 14% of the total, three lost cycles equal 42% distortion. The poor old UART can only take 45% before it gives an error. What do we do now if we lose a cycle or the tape drags? The tape MITS used has too many dropouts, causing the loss of one or more cycles, causing still more delay and distortion. With this system it is far too easy to exceed the distortion limit.

Let's talk about cassette motors for a moment. In the low-cost home recorder they cost about \$1.70 each in quantity. We use this type of motor in the CC-7 model (not the CC-7A) and we throw away 5% of the motors we buy as too noisy. They radiate like a spark gap into a TV set so that your data stream has some unwanted bits. They are mechanically speed regulated so that they will hold within about 1/3%. But — they are specified as having a spread of  $\pm 2\%$  from nominal speed. If you take 100 recorders and put a test tape through, you will find some as far off as 4%. That's not much you say. Well, in a 10-bit byte it is 40% distortion in the last bit. A plus 2% on the recording machine and -2% on the playing machine is 4%. Now add this to the loss of bits, PLL problems, occasional cassette catches and you have a bad data system.

You may accuse me of overstating the distortion in the PLL, but even if you cut my figures in half and allow for the speed spread above,

the miracle is that any ACR units work at all with MITS software. The same is true of any PLL — FSK system; I don't mean to single out MITS. Teletype data sets use this FSK system, but they do not use little 1 chip detectors. It is the PLL type FSK detector that causes the problem.

My solution was to play the MITS cassettes on a variable speed cassette player adjusted in speed so that it locked exactly at 300 baud on playback. The tones were then put through a 103 Data Set used on the Teletype line and RS232 data extracted. This was recorded on a reel-to-reel NRZ recorder which was used thereafter to feed the Altair (flawlessly).

Compared to the MITS-ACR board, the Kansas City standard (which is also FSK) was a real blessing. It is not the end-all, however, because it is still subject to the speed problems between audio recorders when operated asynchronously and it is too slow. In implementing this so-called standard, the circuit boys did a good job eliminating the PLL problem, but did not improve the low (300) baud rate. Fortunately, they made it self-clocking so that the varying cassette speeds could be accommodated in externally clocked operation. Unfortunately, this method of operation increases the complexity of computer to cassette interfacing.

### The "Standard"

I'll summarize my objections to this "Standard" as too slow and the use of stop and parity bits was not specified, i.e., fixed. Unless everyone uses the same number of stop bits and parity bit check, there is no standard. Checksums have been suggested as the answer. The checksum proponents claim that by using them you can forget about parity checks. OK, some fancy systems using 9-bit tape do both parity and checksum loadings. You can do that here too.

The trouble with checksums is that there is no standard and you must know the sum. There is an Intel checksum, a Motorola MIKBUG<sup>\*</sup> checksum, a MITS checksum, a Tarbell checksum, etc.. Let's keep checksums out of the standard. Let the standard record and play any format with a parity check. Then, if the tape is to be played via an Intel or MITS or Motorola or any other checksum, it is a software, not a hardware decision, and any special instructions for software can be included with the tape listing. The checksum thus becomes the second check for data integrity.

The HIT standard, a tone burst method, never caught on so we'll ignore it. It had little to offer anyway and IMSAI more or less confessed they made a mistake in bringing it out. They have subsequently more or less abandoned it.

Among those who have gone over to the Kansas City standard are SWTP, IMSAI, Pronetics, PerCom and Morrow. Pronetics/Percom offer a straightforward interface board. The others require that you use the computer to get coding and require elaborate software packages. I haven't tried any of these but I do own a Morrow. It comes as a fine looking kit, with ROM for software and absolutely no, none whatever, instructions on how to use it. If someone will tell me how, I'll try it. The Southwest Tech dual system has many dedicated users and therefore must be assumed to work well.

No matter who you buy your KC standard kit from, at present you are limited to 300 baud and every user of KC standard tapes has to have a recorder within 1% of standard speed or have a self-clocking I/O board. This gives you a fair chance of tape interchangeability and an even better chance if the program is recorded twice on

good tape. But notice in the ads for KC standard interfaces they say "can be operated at 2400 baud or 218 bytes per second" (PE 2400 is creeping in).

We are after a good tape interchange standard for hobbyist use. There really is only one suitable standard that will fit a variety of low cost audio tape recorders at respectable baud rates and that is PE — the ANSI ECMA standard. It is self-clocking, so you can use almost any audio recorder that is motor noise free (or a digital recorder for that matter), and it can be used with 8080, 6800, 6502 or any other system at 2400 baud. Speed range is such that you can accommodate Tarbell tapes at about 1500 baud and the Byte standard if you must.

Let me take off for a moment on the "Tarbell" method which some people call a standard. Don Tarbell deserves a pat on the back for his Tarbell board which was the first to use an audio recorder in the Phase Encoding mode. There are several things wrong with it as a standard, however. Unless you buy the Tarbell board, or own an IMSAI 2SIO or National Multiplex 2SIO(R) board, you can't use it.\*\* This immediately leaves out all those who own MITS or Processor Tech boards.

\*\*These boards use the Intel 8251 which can use the IBM bisync code which Don uses. In other words, you can program an 8251 for Tarbell.

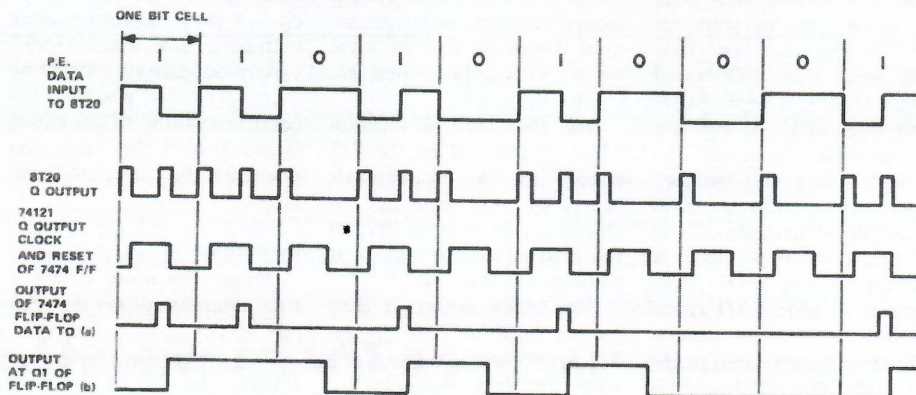
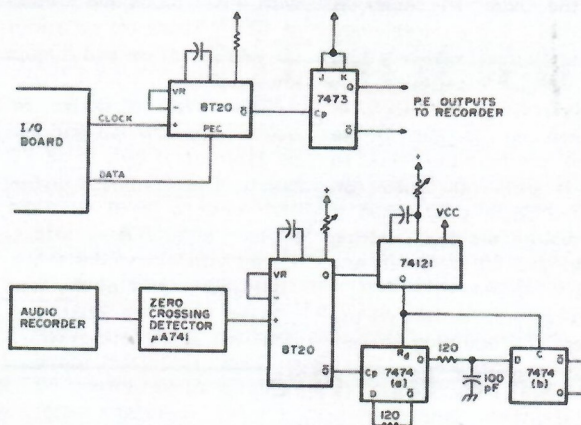


Fig. 1.



People with 6800 and 6502 systems are also out of luck. You can't have a standard unless everyone can use it, therefore the Tarbell system cannot be a standard.

I also object to the checksum loader which requires mental gymnastics to get it into the software. A simple 16-byte bootstrap will load KC standard tape if you use parity checks. The same is true of PE 2400 to be discussed. And, while I'm objecting, let me object to 1500 baud, eight bits, no start bit, no stop bit and no parity bit as well. For some reason, Don seems to have taken the 800 bpi figure of 1968 as gospel forever. That 800 bpi was about the limit of tape capability in 1968, but it is nowhere near it today. It also applied to saturated (NRZ) recording and not to audio. 1500 baud

is not a standard baud rate, which is a problem unless you have a free running or PLL baud rate generator. Lest Don throw a brick at me, let me say I know a lot of people who swear by his board. I do maintain, however, that a Standard must apply to all I/O boards and all microprocessors.

#### The National Multiplex PE 2400

The advantage of PE 2400 is that you do not need an I/O board made especially for the purpose, such as the Tarbell. The PT 3 P + S, the MITS — 2SIO or 88SIO or the IMSAI 2SIO will do. The National Multiplex 2SIO(R) has the PE control programs in ROM. For best results with any cassette system you should use a monitor/loader in ROM, but you can bootstrap it in. The monitor/

\*Registered Trademark

loader will format your data and allow you to start up cold. If you like, you can build your own I/O board using a UART, a 6850 ACIA or an Intel 8251. If you are the real clever type, you can use a PIA in a "bit banger" circuit, but the 8251 or ACIA are preferred because of the 1/1 clock. A 16/1 clock is required for the UART, but an easier to obtain 1/1 clock can be used in the others. 6800 system users will probably prefer the ACIA or PIA in a "bit banger" circuit.

How does PE work? See Fig. 1. We start with a clock and the data. The clock is doubled in a frequency doubling circuit which is used to drive a  $\div 2$  counter. This merely restores the clock, as it does on 1s, but if we inhibit the doubling circuit so that it puts out spikes on positive edges only, then it will restore the clock during 1s and give 1/2 clock outputs on 0s.

To restore the data we use a frequency doubling bidirec-

tional one shot. This drives a second one shot whose period only allows it to fire once every two pulses. This recreates the clock.

We then add a flip-flop which is clocked by the 2/1 output of the bidirectional one shot and reset by the clocking one shot to restore the original data and clock it through a D flip-flop.

The clock output is used by the 6850 or 8251 in the 1/1 synchronous mode along with the data. A 16/1 PLL and divider can be used to restore the 16 to 1 clock for UARTs but you are better off with a 1/1 clock and a 6850 or 8251. There are variations on this which we will discuss later.

The circuit is simple, requires very few ICs and can be built on a very small PC board. The PC board layout and how to build it is the subject of the next article. Then you need software. That also is part of the next article. Both the 6850 and 8251 use mode latches which

must be set in software to set up 1/1 clock operation along with parity bits, etc. A UART system can operate with minimum software if you have the cassette running to get a long clocking lead time and you manually start/stop the computer.

#### Summary

Phase Encoding offers a data interchange system that can accept a wide speed variation, operate at 2400 baud (hence, the PE 2400 name) with most recorders and does not require special I/O boards. That is 8 times as fast as the Kansas City standard.

Now, what about the Kansas City Standard. It is an FSK system (2 tone), but if you look at it another way, it is a redundant phase encoded system. It is just a long string of 2400 baud 1s and 0s. Therefore, you can operate this phase encoder with KC standard tapes at 300 baud or pure PE tapes at 2400 baud. Only your clock rate need be changed. You can

also operate at 600, 1200 and 1500 (why bother) baud. You can use the Tarbell 1500 baud system only if you have an 8251 in your I/O board and program it for BiSync.

While we're at it, let's resolve the whole standard problem. ANSI/ECMA/ASCII standards call for 1 stop bit at 2400 baud. Let's stick to it. Next, let's specify "even" parity and get away from checksums. Now we really have a standard. Even the name PE 2400 has all the features a standard needs because it says Phase Encoded, 2400 baud, Parity Even. It can be used on any I/O board or microprocessor. Now, if everyone uses 1 start, 1 stop, and even parity bits, all tapes can be interchanged and you don't need to know the checksum number. Phase is always the same — standard TTL Mark = (+5). Maybe we should call it the "Kill a Byte" standard. Anyway, you'll be hearing a lot about PE 2400 in the coming year. ■

## ATLANTIC CITY 1976

