Hughie's HOT TIPS c/o
Hugh Macdonald
Psychology Dept.
Stanford, CA 94305

Hughie's #2 HOT TIPS -PHI-

TIP # 2: GETTING PHI DECKS TO WORK AT 10^8 BIT ERROR RATE: Possible?

The most common conversation following the mention of a DG Phi Deck system is "Does it work?" or some other comment about its reliability. Alan Neel, primary designer of the Phi Deck system, told me that he can get 10^8 reliability through careful tuning and cleaning -- but admitted to some of the flaws in the system which I'll mention later. A San Francisco DG addict told me "Oh mine works fine, no problems, in fact, when the capstan roller spring broke, I just put in a block of wood and no problems!" He told me that his 50 amp power supply was the cause of his good luck. The Chicago group, and the San Jose group are all happy with their Phi Decks -- if they are carefully tuned.
... ... However,... .... The vast majority of responses, queries, and other references to the Phi Deck system are often unprintable. Suding mentioned throwing them in the ocean; one subscriber mentioned jumping off a bridge; and another sent his back immediately and ordered disks.

DISKS AND PHI DECKS BOTH SPEC'D AT 10^8 BIT ERROR RATE

This makes me theoretically happy. However, my Phi decks always reread and my disks never do. (Of course it is harder to see the disk do a reread since they are so fast; however, I have copied 250 KBytes from disk to disk and verified them without hesitation in the stepper motor which guarantees a rate which exceeds my phi decks.)

The VERY IMPORTANT QUESTION is: What is 10^8 bit error rate? This is a "soft error" rate that means that one bit will be misread in every 100million bits read correctly. 100 million bits is a lot! In the Phi deck format it takes 10 bits to form an 8 bit byte, so this means that you should be able to read 100/10 or 10 million bytes before seeing the Phi decks perform a reread. (A soft error is recoverable.)

	Size tape	Minutes@5ips	# of bits @8000/sec	#bytes@800/sec	
	C30	15side/2.66= 5.6	2.7 10^6	.27 Megabytes	(.25) 94% ADJUSTED
(300 feet)	C60	11.25	5.4 10^6	.54 "	(.51)
	C90	16.875	8.1 10^6	.81 "	(.77)
	C120	22.5	10.8 10^6	1.08 "	(1.02)

A commonly used tape is the C30 which holds 2.7 Megabits, or .27 Megabytes of data. However, the Phimon format requires 271 bytes to store 256 (quite efficient! 94.5%) so a C30 tape actually will hold .255 MBytes, i.e. the same amount as an 8" floppy (single density.)

10^8 = 37 READS WITHOUT ERROR OF ENTIRE C30 CASSETTE

Fill a C30 cassette with 5.6 minutes of data (250Kbytes) and read it 37 times without noticing a single reread. Impossible? This is what 10^8 bit error rate means. Of course this rate is an average so that if you read it 370 times you would only see a total of 10 error although some reads would have none and some would have several.

I wrote a small modification of the DG supplied phideck test routine to perform an actual test of my decks for bit error rate. My new routine records one header (all 1's followed by a SYNC character) followed by an infinite string of data bits. The data can be all the same character or the standard sequence of characters from FF thru 0 and repeating. The

first sequence tests the response to a certain pattern of bits whereas the second sequence tests the response to all combinations of bit patterns. The playback routine is changed to display the characters read from the tape on the TV screen, giving you a visual display of bit errors should they occur. Thus with the first pattern might show a continual stream of the letter "e" on the screen, changing to another letter like "I" if a bit should be incorrectly read. The likelihood is that if one bit is read incorrectly, the synchronization of the bits will be thrown off and the remaining bits will be read and interpreted as a different character. Thus is is easy to see a bit error on the screen in real time and to note about where it occurs on the tape.

RESULTS OF BIT ERROR RATE TEST

To make a long test short, I used a C90 tape that required only 12 complete reads to establish a 10^8 rate. My unmodified A version board rarely read the tape once without approximately 10 errors, most of them at the beginning or end of the tape. The initial error rate was thus approximately 8.1/10 or on the order of 10^6. This is 100 times worse than specd.

I made the Hughie's Hot Tips STILLSON PHI FIX modification and for the first time was able to read a whole tape without one error. I also changed to high quality computer tapes and observed even better response. However it was still rare that I could read the tape twice without an error, so the rate was improved to about 10^7. This almost respectable amount degraded when the tape was put in another drive or over time, temperature, and the mystical vagaries of the DG equipment.

I bought the version C upgrade, hoping the phase locked loop with 20% speed tolerance would solve everything. No soap. Without the STILLSON PHI FIX the rate was 10^5; with it, it was about the same --perhaps 10^6 -- but by this time I was tired and knew something was wrong with the system's heart or front end.

MANY THEORIES:

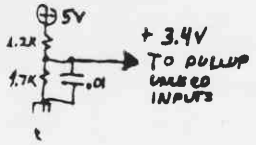
I am calling this writeup the FINAL theory and explanation of the causes and cures of Phi deck ills. Over the past two years I have tried so many fixes I've forgotten half of them. I finally generated a hypothesis that bit jitter was causing misreads. Alan Neil said that the bit jitter was very bad and seemed to corroborate my own theories. My theories of only ~~two~~ FIX months ago have changed dramatically so I must say that I was partly wrong and partly right. The following reconstructed drama describes the way I should have debuged the phi decks in the first place:

THE (?) TAMING OF THE PHI MOUSE:

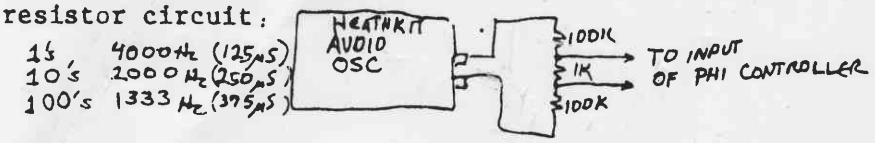
(The motors for the phi deck cost Triple I manufacturers $1.00. They are cheap. The mechanics of the deck are solid, cast metal and simple --reliability? The long cassettes don't seem to rewind all the way. Thus PHI decks are mouses. This explanation is an aside, an illusion, and doesn't really mean anything.)

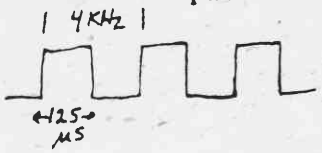
STEP 1: DIVIDE IN HALF, FIND THE SOURCE OF THE PROBLEM

1) For a long time I have suspected the TTL circuitry for glitches and other unresolved and nearly invisible errors. The STILLSON PHI FIX is aimed at this -- to terminate the unused TTL flip-flops and latches. A scope placed on the unused inputs to the 7474s shows a wiggling mess, so this seems like a good idea. I made a resistor divider with capacitor filter and installed it near the +9 volt regulator to supply the pullup for all of these chips (NOT Stillson just wired them directly to +5. However, this fix does not produce a miraculous improvement so I did the "real test."



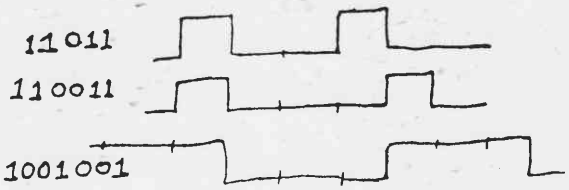
The REAL TEST is to eliminate the phi deck drives. I constructed a simple resistor circuit:





The audio oscillator produces a "jitterless" bit stream. Note that the frequency of 4 KHz gives a 125 microsecond square pulse, alternating positive and negative at 8000 per second.

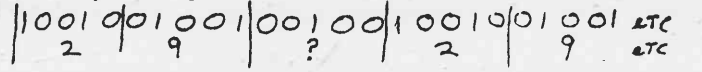
Remember that the phi deck uses GCR, or Group Coded Recording (invented by IBM) that records in NRZI format a bit stream that never has more than two 0's in a row. Thus there are really only three possible "frequencies" present in GCR: 1's (4000 Hz), 10's(2000 Hz) and 100's (1333 Hz.) NRZI could have anywhere from 0 to 4000 Hz on the one extreme and on the other, PE (Phase Encoding) would have only two frequencies 4000 and 2000. (GCR is very efficient and self clocking -- see your phi deck manual -- it is used on the modern 6250 bits-per-inch 1/2" computer tape drives.)



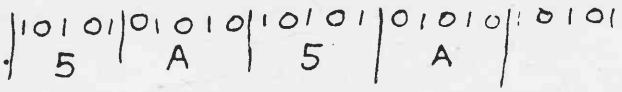
Example Waveforms GCR (similar to NRZI but two 0's max.)

RESULTS OF THE REAL TEST: No errors!

Read the phi manual to understand the encoding scheme if the following is unclear. Each block of data begins with all 1's (4000 Hz.) I loaded in the phi deck test routines, pressed "6" (for READ) and fed the controller 4000Hz. To simulate the "SYNC" character, I would switch the frequency of the oscillator down to 1 or 2 KHz a few times until the tv screen showed a display of characters -- meaning "sync achieved". Once locked in I could change the the oscillator to the three different frequencies to observe the character resulting from that bit pattern. Depending where the bits changed, the character would be any character that contained the correct pattern, i.e. at 1333 Hz, the bit pattern is 1001001001001001, each one being a transition and the two zero's an absence of transition. If you consult the 4bit to 5bit translation table in your phi manual (or in the back of this tip) you will find that this bit pattern can be 29_h or $9X_h$ where the "X" represents a code not possible in GCR but interpretable by the phi controller ROM.



The pattern for 2000 Hz is better: 5A or A5, so once locked in, the same character will repeat forever.



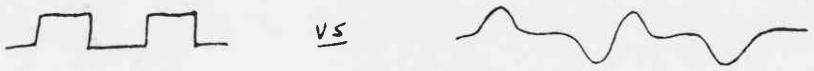
The 4000Hz pattern is all 1's and is not defined in GCR but results in one stable character(that I forget.)

No matter what frequency I fed the phi deck controller, I observed no errors. NO ERRORS! (I did observe errors until I adjusted the symmetry of the audio oscillator so that the positive and negative durations were equal.) If I adjusted the frequency to be just between the two normal "10" and "1001" BIT PATTERN frequency, then I observed errorsthat looked just like I had a tape in the phi deck drive. They were somewhat random --only occasionally "glitching" and producing an error.

The conclusion is that the TTL part of the controller board works just fine -- or at least to the 10^9 or better rate. 10^8 means 3.5 hours of error free reading; 10^9 is 1½ days of no errors!

STEP 2: THE FRONT END AND THE DRIVES

The above test verified most of the front end circuitry of the phideck controller board. It is true that I was sending it a square wave of high voltage whereas the typical tape produced signal is rounded and wiggly.



So my suspicion fell on the low level signal conditioner circuit and the tape drives themselves. I find it hard to separate these two since the problems I found in them and am about to describe interacted with each other.

THE ANSWER: BIT JITTER (TIMING) IS THE PRIMARY BUG AND IS CAUSED BY BOTH THE DRIVE MECHANICS, HEAD INDUCTANCE, AND A NOISY SLOW-SPEED FRONT-END CIRCUITRY.

Simple, eh? If the bits come in at exactly 125 us (micro-second) or 250 or 375, then there are no errors. If the bits exceed approximately plus or minus 20% then there are errors. (E.g. 250± 20% = from 200 us to 300 us.)

20% BIT JITTER IN MY PHI DECKS???

I was a bit surprised to note such deviant bits. But indeed, I was never able to observe an error (watching on my scope or using the micro software) that was not caused by a bit which exceeded its specs. I continuously monitor the bit times by a HP counter-timer and observe that 201 us gives a 10 ouput and 199 gives a 11 output.

The obvious question is WHERE IS ALL THIS JITTER COMING FROM? The Phi decks are spec'd at ±5% speed tolerance -- so this can only be part. The remainder of the jitter comes from sticky or too loose (yes too loose) tape cassettes and from the input circuit, i.e. "electronic jitter."

SOURCES OF JITTER (NOTE: SOLUTIONS MENTIONED BELOW ARE SUMMARIZED AT THE END IN CONSTRUCTION DETAIL.)

A: THE MECHANICAL DRIVE AND CASSETTE: The capstan appears fine, but since there is no clutch on the left-hand (rewind) spindle, mechanical stickiness on the supply reel can transmit vibration to the tape next to the head. Even the professional tape cassettes can cause problems: some are so loose that vibration in the supply reel transmits so easily as to make them unusable. I observed ± 5% jitter with Scotch AVC cassettes and a whopping ±10 % when I switched to a professional computer cassette. The AVC cassettes were so sluggish that they were almost impossible to fully rewind, whereas the professional cassette would zip at super high speed and finish with a resounding clunk. This jitter is especially bad at the very beginning and very end of the tape where the tension and leverage of the small and large diameter reels is most extreme. ±10 % mechanical jitter plus a possible ±5% from tape drive speed accuracy gives you ±15% already! This is beyond the capability of the unmodified phi controller which, according to the manual, can only sustain instantaneous jitters of ± 7%. (The long term tolerance is ±20%.) Since the version B and C board contain a phase locked loop adjusted for nearly instantaneous bit times, their tolerance may be larger, at least theoretically. See below under Electronic Jitter .)

SOLUTION A: 1) Use the little wheel with the timing marks on it and tune the speed of your capstan to standard.

2) Find high quality tape cassette with enough drag to work,

but not too much so that they don't rewind fully. (DAK EC 60's EC 90's (ADVERTISE IN POP ELEC) ARE NOT MADE ANYMORE "JAM PROOF".) (TDK D's are reported to work well. I suggest you buy from the same lot if you find one that works.)

The cassette should rewind easily to the very beginning and be dropout free. Certified computer cassettes are dropout free, but may have the problem of"super looseness." However, the loose cassettes will wear better than the conventional ones which may start to drag after a while and not rewind fully. THE TAPE SURFACE should BE VERY SMOOTH LIKE A MIRROR (CALENDERED).

3) Keep your cassettes in the cleanest possible environment. Dirt works it way into the cassette and increases the stickiness and drag. Dirt also gets ground into the surface of the tape and shortens its usable life.

4) HYPOTHETICAL: Modify the controller board to provide back tension, two capstans, or an erase head or other"tape steadier."

5) TOM COX says: much use mechanically vibrates the speed pot located inside the capstan motor; it can slow 5% a month -- so tune speed each month.

SOURCES OF JITTER

B: THE TAPE HEAD CONTACT:

Tape to tape head contact depends on the smoothness of the tape itself, the kind of pressure pad, and on the shape of the head. We are stuck with the head, but the smoothness and pressure pad depend on the cassette. See the above solution #2, 3, and 4.

A one thousanth of an inch separation of the tape from the tape head gap can reduce the signal output by about 50%. Specks of dirt can do this, as can a crease in the tape or vibration transmitted from the supply or takeup reel. Remember that the capstan -pressure-roller does not prevent small vibrations from passing through to the head.

If the head is misadjusted, the signal can also be reduced. Azimuth errors reduce high frequencies more than low frequencies and the resultant variation in amplitude can eventually appear as bit jitter. (see below "Electronic jitter.") If the head sticks out too far, it can press so hard against the pressure pad that it skews the travel of the tape, loosing its signal, or causing the tape to become creased. Some pressure pads are too thick and cause the same thing.

SOLUTION B:

1) If you can find an alignment tape (azimuth) adjust the tape heads of your decks. A HI-FI service center usually has one and if you find an audio deck that has been adjusted with such a tape, you can make a"second generation" alignment tape by just recording a 4KHz sinewave on the newly aligned deck. Then play this 4KHz signal on your phidecks and adjust for maximum output (put scope or sensitive AC meter at pin 8 of the LM3904 preamplifier, IC 1 A BOARD IC 13 C BOARD). You adjust the screw next to the head that adjusts the tilt of the head. Once adjusted, you should put a drop of LOCKTITE or DUCO or other non-permanent glue to prevent vibration from changing the setting.

NOTE: Azimuth adjustments should not be attempted unless you have the right equipment since you may make your deck worse. Azimuth does not generally cause serious errors but is worth a try if all else fails.

2) See solutions A above.

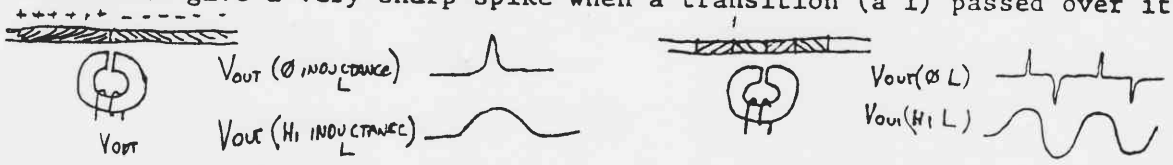
SOURCES OF JITTER

C: TAPE HEAD INDUCTANCE :

The Phideck manual proudly mentions that the VIKRON head used is a digital head with a frequency response of 100 KHz. I have two decks bought at different times and the heads seem to respond differently, but not significantly 50 A. However, there is a significant difference in the DC resistance of the head: one (older) is 40 ohm , the other is 20 ohms. Usually the smaller the resistance the lower the inductance and thus the higher frequency response of the head.

(One guy at the NORCAL (Northern CAl) meeting of DG users told me he had a head that measured about 5 ohms! He wanted to get a new one 'cause he felt it must be shorted. Maybe?)

I measured the inductance of the head and found it to be about 10 mH (5.15mH) (miliHenries.) This appears to be sufficient inductance to cause a bit timing shift although I am not truly sure. My reasoning is this: If there was zero inductance the head would give a very sharp spike when a transition (a 1) passed over it.

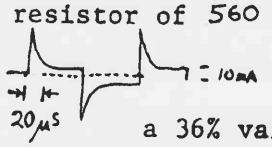


Note how the high inductance head rounds the spike. Inductance means that the head resists a change in the flow of current, so it takes time to get up there, and once there, it takes time for it to come back.

The primary effect of head inductance is to reduce the magnitude of the output signal which of itself is not too bad were it not for differential amplitude effects and saturation effects, mentioned below.

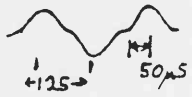
HEAD TIME CONSTANT: The hi frequency transitions have a repetition rate of only 8 KHz, which is far from the 100KHz limit of the head. However, the transition lasts for only a fraction of the repetition rate, so this is the time constant that is important. The head's 100KHz rating means that its 95% time constant is 1.6 x 5 = 8 usec. This is 6.4%. of the 125 us bit time.

The controller (modified) drives the head with a TTL square wave (approx ± 4 volts) through a resistor of 560 ohms. Observing the waveform across the head:



It seems to actually take 20 usec for current to reach a 36% value. 36% = one time constant and t = L/R or L=tR so head inductance L = 20 x 560 = 11 mH. In the modified controller the final current of 10mA is reached in about 5 x 20 =100 usec -- although this is the 99% level and for all practical purposes the current is final (90%) at 40 usec. As you can tell, although these measurments are far from accurate, they serve to show that the head's time constant is a rather large part of the bit-to-bit time. In other words, the transition recorded on the tape is not as sharp as I expected given a "100KHz" Nikron head.

While I was at first dismayed by the "slow" head, I later discovered that the typical magnetic tape was the limiting factor, not the head. Or perhaps it is the combination, I do not actually know. But no matter how sharp a signal I was able to record onto the tape, the playback always looked like this: It was rounded. It took 50µsec to rise to a peak and 50 µsec to return to zero, or 100usec for the pulse total.



MAGNETIC TAPE PARTICLE SIZE INDUCTANCE: Audio vs Digital

I have recently read that the best quality audio tape often combines super fine ferric oxide particles with big clumpy ones to obtain a well rounded audio response. The tiny particles are for the high frequencies, the big ones are for the base frequencies. Unfortunately, those clumpy particles serve to make the tape appear sluggish to a digital phi deck. The big particles are hard to erase and help the tape to retain images of supposedly erased data. In short, the big particles are for the frequencies not at all used in digital recording. This is why true digital tape has a thin (easily erased) coating of very fine (hi frequency) particles with no dynamic range (digital recording is "saturation

recording", i.e. no large dynamic range is required.) Digital tape is also super smooth. The surface of the tape is compressed by heated rollers so that the surface is smooth like a mirror. The smooth surface, besides reducing head wear, flakes less and is able to come into a much more intimate contact with the head. The closer the tape is to the head, the higher the output voltage. As I said earlier, a .001" separation from the head can reduce the output by 50%. In summary, digital cassette tapes are smooth running (precision guides, rollers, and lubricated and ribbed slip sheets), low static (graphite in the slip sheets and perhaps back and/or front-coated with a conductive agent), smooth surfaced (highly calendered mirror-smooth tape), and of low dynamic range (easy to erase, easy to saturate,i.e. high frequency.)

SOLUTION FOR C:

As you will see, it is possible to get good results with the heads supplied with the PHI deck, so changing the head is not needed. The head-tape interface seems to be slower than the effect of the head inductance.

Since the PHI deck user uses both audio and digital type cassette tape, it is recommended that the head be driven with more current in order to fully saturate the tape oxide. I have experimented with different drive currents and, just as explained in the books, I get the best frequency response (transition time) when I use low current drive, but this success is at the expense of excessive retention of previous bits, i.e. insufficient erasure. The residual signals are devastating to the rerecorded signal, sometimes adding together with the present signal and sometimes cancelling the present signal. The poor erasure more than destroys the gain in frequency response.

I suggest a drive of 7 mA final current. (During the transition, the inductance of the head initially resists.) This is achieved by changing the driver resistors from 1K to 560 ohms. Since the TTL 74125 drivers have a hi voltage of about 4 volts, the current through the 40 ohm head would reach i= E/R = 4/560+40 = 4/600 = 6.7 mA. This is the same current specified by the PHI deck manual! Unfortunately, the controller board delivers only i=4/1040 = 3.8mA final current because of the 1000 ohm driver resistors.

The 7 mA drive is capable of fully erasing any signal recorded on the tape, since 7 mA is a saturation level, i.e., it exceeds the magnetic force required to remagnitize all magnetic particles. Lesser current magnitizes the tape within its dynamic range, i.e. another recorder could possibly record a stronger signal that could not be erased by rerecording. The 7mA drive exceeds the dynamic range of the tapes I have tested: Scotch AVC, Memorex MRX2, Computer grade, and DAK ML audio. (I THINK... NOT SURE ANYMORE.)

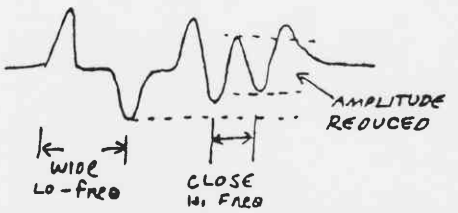
Disclaimer: If someone has recorded signals with a 10mA drive on their own phi deck and you rerecord it with 7mA, it should fully erase the tape and rerecord without error -- according to my limited tests. Nevertheless, it is best to standardize so disparate drive level recording is not encountered.

SOURCES OF JITTER: "ELECTRONIC JITTER"

D: DIFFERENTIAL AMPLITUDE EFFECTS AND BIT SPREAD:

Above I have stated that I observe a ± 10% bit jitter. The VERBATIM ™ data sheet for digital cassette states a ±20 micro-inch bit-jitter due to the mechanical limitations of the cassette. Since phi-deck bits are 625 μinches apart, this represents a 20/625 = 3.2% bit-to-bit jitter. (This value is smaller for the more widely spaced bits: '101'=1.6%, '1001'=1.0%).

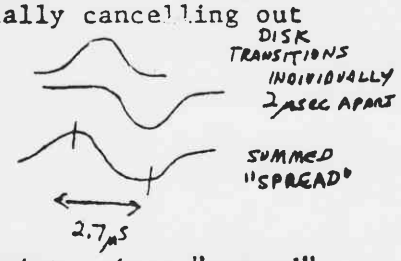
The tape head is slow and the tape is slow; the output pulses take 50 usec to rise, 50usec to fall. But if these pulses are constant, then the jitter is not increased. As you might have guessed, the pulses are not constant. Some tape (especially the most expensive audio tapes) is so "slow" that the highest frequency pulses don't even have time to peak before being pulled down again.



This results in differential amplitudes which depend on the bit frequency. High frequency bits (1's) are reproduced with a smaller amplitude than the lower frequency bit patterns (101 and 1001). The diagram above shows how the closely spaced bits do not have enough time to peak before reversing to seek the peak of the next bit.

Effect on bit shift: A string of 1's stabilize and the peaks of each bit are stable and constant (typically 125 usec.) However, if the string of 1's are preceeded by a 101 or 1001 pattern, the time to reach the first peak will be different than the rest. The same applies to the first leasurely pulse following a string of 1's. I have figured this out on paper, but now I forget which way it goes -- but it is also potentially confounded by the effect of "bit shift spread".

Bit shift spread is caused by a previous pulse partially cancelling out the following pulse. The tails of each pulse cancel each other, resulting in pulses which appear wider apart. (see picture and see Computer Design, Feb 80, page 130.) The non-zero tail of pulse #2 pulls down the peak of pulse #1 , shifting the peak to the left. In disks, the 2 usec pulses can be shifted to appear 2.7usec apart, a 35% error!



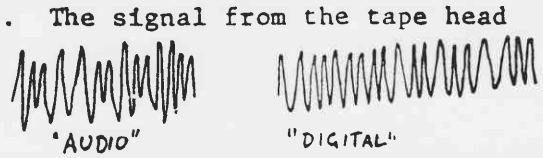
The interactions on the phi deck are not so easy to pinpoint, since "spread" depends on the head and the media, as does the differential amplitude effect. Suffice it to say that I have observed on the HP counter, times of 140usec instead of 125 usec for the 1's, especially after a 101 pattern has occurred. Bit spread is not supposed to occur at the bit density (1600) used by the phi deck, but if the head has a large enough gap, it will. So I don't really know why, but bit shifting does occur and does depend on the bit pattern being recorded.

DAK in LA area. see JAN Feb. 80 Pop Electronics

SOLUTION D:

1) Use high frequency digital tapes. DAK makes a tape for high speed duplication that contains only small particles and is optimized for high frequencies; it is called EC60 or EC90. DAK is a mail-order company. The EC tapes cost about $1 each. They have screws, rollers, graphite ribbed slip sheets, and smooth surfaced tape. I have had good success with these tapes.

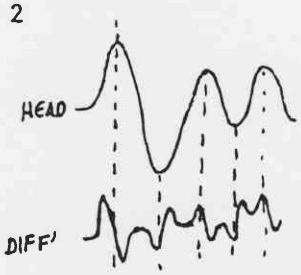
The DAK tapes are for audio, but they work. True digital tapes are wonderful, however, and the difference is quite apparent. The signal from the tape head (after amplification) shows the superiority of the digital over audio. The audio tape has a much increased differential amplitude effect for the differing bit patterns.



1's are noticably lower in amplitude than 101 and 1001's. The output of true digital tapes is very constant and independent of pattern.

Digital tapes are constant and higher frequency, but the difference between them and audio tapes is highly reduced by the decoding method used by the phi deck controller board. It is even more highly reduced by the modified controller board to be described. But, if you can afford it, use digital tapes (providing they are not too jittery when used on the phi deck drives.) If the digital tapes do not work, use the duplication type cassette mentioned above. In any case use one with rollers. I have been able to use almost any C30 tape, but many c60 and all c90 tape cassettes which do not have internal rollers will not fully rewind on the phi deck. The DAK C90's do fine, at least, when new. Haven't had them long enough to find out how they age.

Why is the differential amplitude effect reduced by the decoding method?



A differentiator circuit detects the peak of the signal and the peak is relatively independent of the instantaneous amplitude. (Bit shift spread is not reduced by the differentiator.) The differentiated signal goes thru zero at the peak of the input wave. The modified controller circuit further eliminates bit shift by providing a wider tolerance of ±50% for bit patterns that are close together.

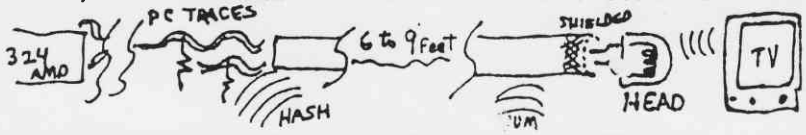
The differentiator circuit works best to eliminate the effects of slow changes in amplitude or changes from drive to drive or machine to machine. The instantaneous change from one bit to the next would be greater if the decoding sensed the zero-crossing rather than the peaks. Again, I could try to describe why, but you can work out your own trigonometry or geometry as I did, and then you will still be somewhat confused, as I am. An expert on these matters might be able to correct my rationalizations, but I am sure about my observations: → bit timing depends on the bit patterns, tape, and decoding method.

Summary: Use high frequency tapes and clean up the controller (see below.)

SOURCES OF JITTER: "More Electronic Jitter"

E: DIFFERENTIAL INPUT AMPLIFIER:

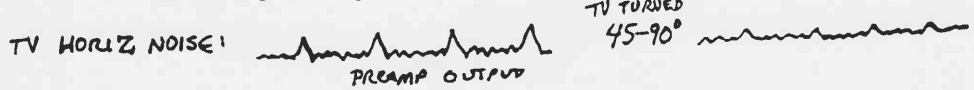
When I first read the Phi Deck manual, I was very impressed to read that a "true differential" amplifier was used. A differential amplifier has the property that common mode signals (noise) are cancelled whereas the differential signals are unaffected.

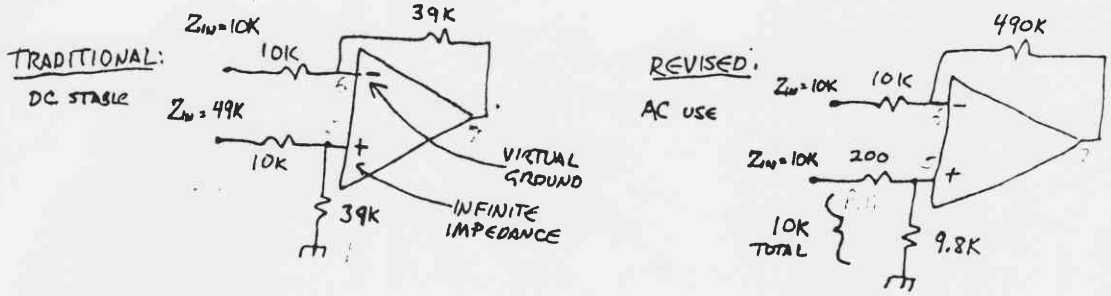


An example of a common mode signal would be power-line hum. Since both wires leading from the head are together, both should pick up the hum signal in equal amounts. The differential amplifier subtracts the two, leaving zero. If the hum signal is picked up more by one lead than the other, then the difference is not zero, and noise will appear with the signal. This is why AC leads and signal leads are always twisted: the two leads are very close together ensuring equal pickup of external noise. (High power AC leads will also not radiate as much as the RF cancels itself.) However, the main desire is that the noise signals be equal in both input leads so that the differential amp can difference them and yield zero. The actual input signal from the head is differential, one side plus, one side minus or vice versa, yielding a difference which is not zero.

THE PROBLEM: The 324 op amp on the phi deck controller is wired as the traditional "differential amplifier" but it does not act as one. Thus common mode signals are not cancelled and a higher noise is output.

THE TV PROBLEM: A main source of common mode noise is the video signals which drive the TV as well as electrostatic and magnetic radiation of the video -- especially the horizontal sync pulses. THe TV yoke seems most responsible for the radiation of the magnetic noise, although the horiz flyback might be helping. I have watched the scope as I turned my TV to a 45 degree angle with the phi decks and have seen the horizontal sync pulse noise reduce to one-fifth of its previous jagged height. As you might have already guessed, the magnetic radiation affects the phi deck head , not the wires, and the resultant noise is differential, not common mode! Thus this noise cannot be cancelled by the differential amp. The electromagnetic RF video can be helped by the differential input amplifier.





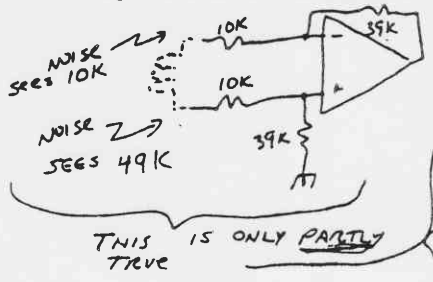
The TRADITIONAL vs. REVISED DIFF AMP: The traditional amp (above left) appears to be symmetrical. The ratio of each leg is the same 39/10 and the bias current coming from the op amp sees the same resistance, 39//10. (39K in parallel with 10K.) As temperature increases, the bias changes, but since both + and - inputs have matched values of bias, the difference is zero and the output voltage is stable with regard to temperature. (The above discussion assumes the inputs are connected to the same resistance to ground.)

What is not obvious in the traditional amplifier is the difference in Z in. THe input Z (input impedance) is essentially a resistance comprised of the resistors shown in the diagram. If you do not know already, the op amp works on the principle of negative feedback which keeps the - input at "virtual ground". Thus if 1 volt is put on the 10K input resistor, the op amp will adjust its output voltage so that the - input stays at 0. In the example above, -3.9V at the output will, by voltage divider principle, result in 0 volts appearing at the - input. (This discussion can also be phrased in terms of current injected and extracted from the - input so that the net current is zero.) Whether or not you understand the reason, you should know that the - input is always at "virtual ground"; and the signal input to the 10K resistor attached to the - input sees only the 10K resistor, not the 490K.

The traditional diff amp circuit is fine as long as the input device (in this case the tape head) has an impedance which is low compared to the 10K input resistors. If the head impedance is high comparatively, the common mode rejection ration (CMMR) will degrade because the signal that finally reaches the op amp + and - inputs has to travel through different impedances.

THE HEAD IS AN INDUCTOR: An ohmmeter on the head will show about 20 to 40 ohm DC impedance, i.e. resistance. The impedance shown to a 4KHz signal (such as the phi deck bit pattern) will be greater since the impedance (AND NOISE) of an inductor increases with frequency. For a 10mH head, the impedance at 4Khz will be $X_L = wL = 2 \text{ pi } f\ L = 6.28 \times 4000 \times 10 \times 10\text{-}3 = 251$ ohms. At the horizontal frequency of the video (15KHz) the $X_L = 942$ ohms. As you can see, the head's impedance is rising to a significant proportion of the differential op amp's input impedance. 942 ohms is 9% of 10K, meaning that perhaps 9% of the noise signal could be uncancelled.

For a simple example, assume the frequency of the noise is so high that the head's impedance is infinite. The noise then sees the op amp as such:

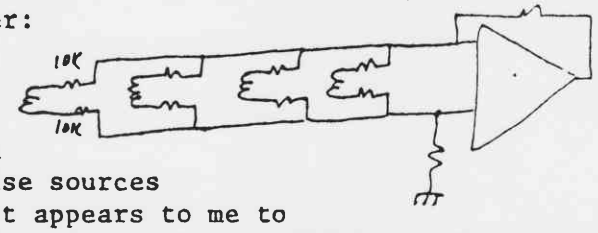


On the negative leg, the noise has to drive 10K whereas on the positive leg, the noise has to drive 49K. It is usually easier to drive 49K --resulting in a differential noise signal, not common mode, that appears at the output.

PLEASE IGNORE THIS RATIONALIZATION

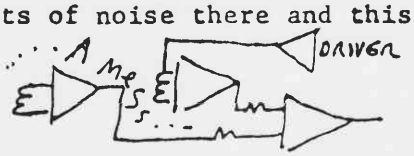
A revised circuit, shown on the previous page, is optimized for the high frequency situation, since both legs appear to be 10K at this point. However, this circuit has its own failings when the common mode signal is large or the frequency is low, but since the noise is mostly video and TTL hash, and the common mode signals are usually low, I have chosen this circuit as my ~~second~~ third favorite input configuration. Perhaps an op amp designer-reader can correct my reasoning above, but I must say that the circuit has been tried with the phi decks and it works fine. If three op amps are used, a true differential instrumentation amp can be constructed, but this is a waste of op-amps considering the resulting gain because of the kind of noise and the kind of signal source and signal level.

PHI DECK HEAD CIRCUIT: Look at what the common mode signals see when they connect to the phi controller:



The heads are paralleled without preamplification to the LM324 diff amp. I reason that if there are common mode noise sources they would love this circuit. It appears to me to be quite unbalanced, with the noise of four heads adding together adversely, since each extra head degrades the common mode rejection even further. In addition, the head lines travel at least 6 feet , connect to Hi-Z TTL outputs (tri stated), and travel up one side of the controller board unshielded. While the impedance of the head is quite low, making it quite insusceptible to noise, the signal it produces while reading is also very low (approx $\pm$ 3 millivolts) and does pick up large amounts of video and digital hash,

I do not really know how much the noise I have measured on the outputs of the LM 324 comes from the video, the TTL, the long leads, etc. or if the noise is large because of the multiple heads in parallel, or because of the poor differential cancellation. But there is lots of noise there and this noise should be causing inaccurate bit timing.



SOLUTION E:

I have put each head on its own amplifier with a pot that adjusts CMMR for maximum noise cancellation. The pot does have an effect, although I have a feeling that the extra powersupply filtering was a prime effect. I don't have enough time to find out, but I do know that the circuit I detail later produces clean hash-free output and that it is engineeringly correct, i.e. it is not wise to put 4 low level tape heads in parallel.

In addition to having its own preamplifier, it is advised that the head driver TTL (74125) be put adjacent to the preamps for minimum noise. However I have obtained 2.5 x 10^{-8} bit rate error without this mod and it requires more changing of the controller board than perhaps desired, but this is what I would find most engineeringly sound.

I also would advise putting two 270 ohm resistors in each output of the TTL driver instead of the one 560 ohm resistor since this would keep the lines travelling to the head more balanced -- at least theoretically. I have not done this to my own setup, but if I were rebuilding the board, I would add this change. I would also put the drivers with the preamps inside of the Phi deck cabinet.

As mentioned above, I have put my head preamps in the same box as the phi deck drives. I also plan to put the $\pm$12 power supply, +5 v supply for the motors, and +5 for the preamps, and +9 for the remaining motors in the rear of the 2 deck phi cabinet and in a wide narrow box under a 4-deck cabinet.

I have not tested putting the phi preamps in the mainframe box, nor do I plan to. Sending low level signals that far is quite stupid, despite the fact that it can work as well as some people report. Fortunately for my opinion, most people consider the phidecks unreliable and erratic, yet with my modifications, the phi decks appear to be quite stable and dependable.

My Favorite Preamp: ~~second~~ 3rd perhaps you noticed earlier that the suggested preamp was my "~~second~~ favorite." My first favorite takes an extra package (chip) for each head, and, while elegant, is probably not worth the extra $2 or so. My favorite circuit is a monolithic NPD5566 (National) dual JFET input followed by an op amp current-to-voltage converter. THe circuit produces extrememly low noise, and has the best CMMR response and highest frequency response. THe input impedance seen by + and - inputs are equal.

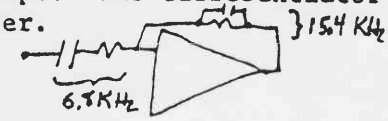
However, the op amp solution takes one fourth of the IC packages by using dual or quad op amps, and seems to work just fine with only slightly more noise and even slightly better CMMR. Thus it is my choice although only my second favorite.

Summary E; Preamp for each head located in phi deck cabinet.

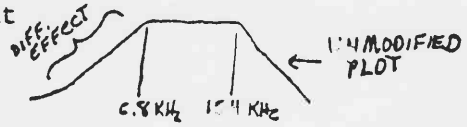
SOURCES OF JITTER:

F: -DIFFERENTIATOR

The phideck controller board uses the LM324 as an differentiator and a comparator in order to recover "raw data" from the tape. The differentiator is standard design but the time constants are not proper.



A differentiator is simply a high-pass filter, i.e. it will pass the input signal while it is changing rapidly but will inhibit it when it goes slowly. This is how it detects the peak. While the input signal is rising to the peak it is changing fast -- the differentiator output goes positive. When the peak is reached the input slows and the differentiator output goes to zero.

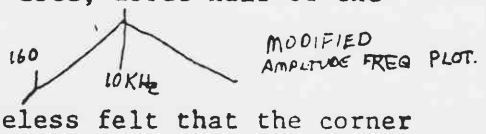


Differentiators have the problem that they pass high frequencies and noise is often high frequency. Feedback loops around the op amp are more unstable at high frequencies because of stray capacitance and other internal limitations. This is why every differentiator rolls off at frequencies higher than those in the "range of differentiation."

The differentiation effected by the differentiator occurs only in the region of the amplitude-frequency plot in which amplitude increases for increasing frequency. In the plot above for the unmodified controller, the differentiation occurs from 0Hz to 6.8KHz. The flat region from 6.8K to 15.4K is improper since it allows frequencies of no use to pass and increases the noise. THe three frequencies used in GCR are 4KHz (125 usec per pulse), 2KHz (250 usec per pulse), and 1.33KHz (375 usec.). While the 6.8KHz corner might seem high enough to include the aforementioned three frequencies, the actual frequency of interest is not one of these three. Instead it is the rise time of the pulses coming in from the head and all of these pulses are equal. THere are not three different pulses; there is only one pulse coming at three different spacings. The job of the differentiator is to detect the peak of the pulses. The pulse looks like this:

it has a 50 usec rise (approx.) and 50 usec fall.

The unmodified differentiator time-constant is 24 usec, about half of the rise time.



SOLUTION F:

Not being an expert in differentiators, I nevertheless felt that the corner frequency of the differentiator should be twice as high as the pulse frequency. The pulse freq is 1/125usec x 2 (there are 2 pulses per full wave)= 4 KHz. Thus I set the corner frequency for the differentiator at approximately 8KHz. Since 100usec pulses come when the bits are 20% fast, I increased this corner to approximately 10KHz. While I lack in full theoretical understanding for this decision, I have observed on the scope a more accurate delineation of the peak and less sensitivity to bit amplitude effects since the differentiator output returns to zero sooner. However, my decision might make the differentiator more susceptible to noise, but in practice, the modified circuit shows less noise than the original.

Since the low frequencies are not necessary for pulse detection -- in fact they can even cause objectionable bit shifting by altering the symmetry of the input wave, I raised the low frequency cuttoff to 160 Hz. The differentiator already ignores the low frequencies at -6db/octave, but I put in an extra high pass filter that passes 160 on up. Thus 160 and below fall off at -12 db /octave, giving further protection from hum (60 and 120 Hz.) and from slow dc shifts from the tape, tape-to-head contact variations, and dropouts.

Since the frequencies higher than 10KHz are deemed to carry no information, this frequency is chosen as the corner for a low pass filter. As shown in the plot, all frequencies above 10KHz are decreased at -6db/octave by the preamp and another -6db/octave by the high frequency limiting of the differentiator. The result is a -12db/octave filtering out of high frequency noise,(mostly composed of the TV horizontal at 15KHz and TTL hash.)

Summary: tune the differentiator corner frequencies to suit signal and noise.

SOURCES OF JITTER:

G: COMPARATOR:

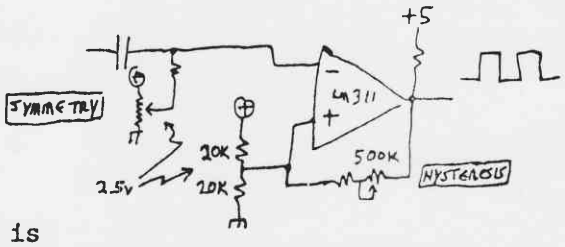
The LM324 acts as a slow comparator to detect the zero-crossings (representing the peaks of the input signal) output by the differentiator. The LM324 is a low frequency op amp and takes a wopping 18 μsec to rise and fall. 18 μsec is 14.4% of the bit time and noise on this line could appear anywhere on this signal resulting in a pseudo "early" bit or pseudo "late" bit. 18 usec is also much longer than suggested for input to TTL without danger of oscillation in the transition region. In addition the output of the 324 swings between points that are non symmetrical with the "1"-"0" points of TTL resulting in a strong assymmetry in detected bit lengths. Thus positive going input pulses that start as 125 usec end up perhaps as 140 usec, and negative going input pulses end up as 115 usec. (Numbers chosen for illustration; observed effect is in the range of 8 usec.)

Not only is the LM324 slow and assymmetrical, the negative input is connected to "V ref", obtained by voltage divider and filtered by a 1 uF cap. (corner freq at 212 Hz.) This voltages depends on the 5volt load and is surrounded by TTL hash. I have observed lots of noise on this line on my scope.

SOLUTION G:

I substituted a real comparator for the 324. A LM311 is compatible with TTL in speed and can accept a ground reference for output while retaining a ±12v supply if necessary. (I have set the Vcc to +5 and the V- to -12.) The output of the comparator is a speedy square wave.

I also added a Hysteresis control and a Symmetry control so that detection of the differentiated pulses could be optimized. In addition, the input to the LM311 is AC coupled so that the low frequencies are further reduced, and the threshold is independent of previous stages. The SYMMETRY pot sets the threshold voltage, and HYSTERESIS determines the amount of positive feedback. The amount of noise observed from the differentiator is about 15 millivolts, so the Hysteresis is set to about 50 to 100 mV to prevent the comparator from triggering due to noise alone.



Summary: true comparator design gives reliable and symmetrical triggering on the zero-crossings of the differentiator.

SOURCES OF JITTER:

H: DIGITAL DETECTION AND DIGITAL PHASE LOCKED LOOP:

The A board used digitally stepped correction in order to place the data window in the middle of the incoming data. The clock was divided by three different values giving a range of adjustment of ± 7%. However, the adjustment was based on the previous transition time and could be fooled into a much worse speed tolerance. The A board needs super well-adjusted Phideck speeds to work well.

The B and C board introduced a digital phase locked loop to permit a ±20% speed tolerance. However, in my board this loop kept losing the signal -- implying that the bit jitter was greater than 20%!? In addition, it takes a significant amount of time for the phaselock loop to reach synchronization with the input data, especially at the beginning where there is no signal due to the leader. Part of the problem appear to be the fact that the PLL sits at one end of its speed tolerance capability until data comes in, increasing the time it takes to reach center. A better PLL might sit at the center.

Probably the major flaw in the PLL is its digital nature. Digital PLLs have problems when used as data detectors. To quote Computer Design, Feb 80, p 132,

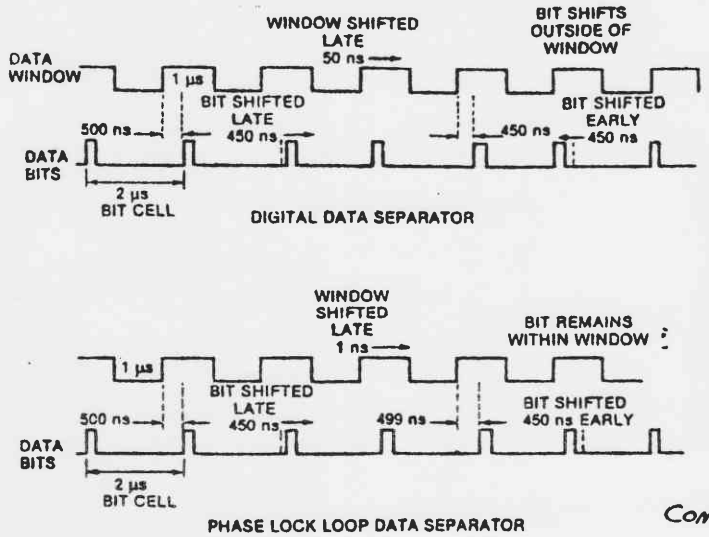


Fig 7 Data separator timing diagrams. Unpredictable bit shift is handled differently by digital and analog data separators. Typical digital data separator, using 20-MHz clock, provides 50-ns resolution. Assuming two worst case bit shifts in opposite directions, data bit would appear outside data window. Analog data separator, however, provides 1-ns resolution. As result, it can handle unpredictable bit shift more reliably than digital data separator

Computer Design p. 133 Feb. 80

Continued from Computer design:

the opposite direction, that bit will appear 50 ns outside the window, resulting in a misread.

An analog phase-lock loop separator, on the other hand, has tighter resolution (±1 ns) and handles bit frequency changes more reliably (Fig 7). With this method, a phase-lock loop locks onto the basic frequency of data bits read off the disc, and determines nominal bit positions for data and clock bits by sampling every bit (clock and data). It uses the phase relationship between a bit and its window to vary the position of the window. By sampling each bit, the phase-lock loop determines the phase error between a bit and the frequency being generated.

Changes in the data window position depend both on an integration factor and on the amount of bit position error. For an integration factor of 100:1, the data window would move 1 ns per bit cell for each 100-ns change in bit position, until the entire error had been compensated.

Using the same example as described for the digital data separator, the phase-lock loop would detect the data bit shifted late by 100 ns. Then, the data window would be adjusted so that it appeared 1-ns late. If the next bit is shifted early, the analog separator would also detect that shift and position the next data window, not late, but early. In this manner, the phase-lock loop reliably tracks frequency changes. A block diagram of a typical phase-lock loop circuit is shown in Fig 8.

Alan Neil (who designed most of the Phi deck) told me that he has achieved better results with a revised PLL in which the 555 frequency is doubled, increasing the resolution by a factor of 2.

Finally, as noted in the "Stillson Phi Fix" by Mr. Stillson, all of the "D" flip-flops (7474's) have inputs which are not terminated to ground or +5 and are left floating. A scope probe shows these points as noisy and near the TTL trigger point of 1.5 volts. A basic rule of TTL is to pullup all unused lines, preferably connecting these unused inputs to a TTL output which remains at "1" all the time. If tied directly to +5, a spike on the +5 could potentially turn on the gate.

SOLUTION H:

The first step is to terminate the 7474's. All unused inputs are connected to a pseudo TTL "1". While you could connect this to +5 and not worry, I felt like doing it "right" and connecting the lines to a 3 volt "1" produced by a resistor divider and filtered by a .01 capacitor. A scope probe put on this line looks nice and quiet. The resistors fit nicely in a spot near the voltage regulator on the right side of the B and C boards. On the A board, I just wired directly to +5, but today I would do as above.

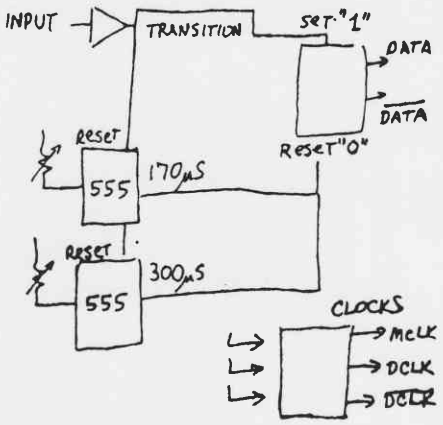
Part two: fixing the PLL? This was hard. It was hard enough to figure out how the damn thing worked -- or should work -- and when I found out, I wasn't happy.

The proper fix would be to redesign an analog PLL, but since the phideck frequencies are fairly low, and the decoding circuitry is all TTL, i.e. fast, I chose a much simpler solution. I used the simple "time a bit" scheme typical of most cassette decoders and like that used on Shugart disks.

Each bit transition begins the timing window. If enough time elapses a one-shot fires indicating that the empty space is to be considered a "0". Since GCR has a possible double zero "00", a second one shot notifies the controller that the dead space contains a second zero. If a transition comes soon enough after the previous one, a "1" is sent out and the cycle repeated. Thus two one-shots are continually timing out, awaiting completion, and being continually reset by each transition (or "1".)

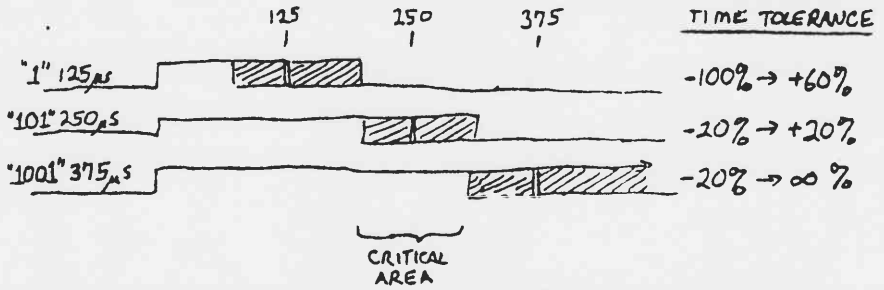
The circuit I developed contains two 555's for one-shots and miscellaneous gates and TTL one-shots. I chose 555's since they are so common and because they have a VCO input which could potentially be used to alter the average speed of decoding --i.e. a range greater than ±20% could be accomodated! I have not developed this VCO approach --that's a future tip? However, you might be interested in the timing diagram and speed tolerance accomodated by my little PLL replacer.

PLL REPLACER TIMING DIAGRAM:



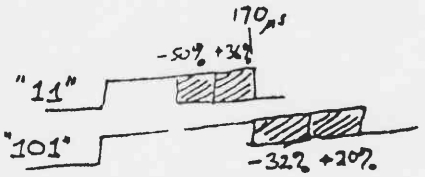
"TIME-A-BIT" DECODER
CIRCUIT BY HUGHIE

BIT TIMES	SPEED	TIME
SLOW: 150μS	-20%	+20%
NORM: 125μS	0	0
FAST: 100μS	+20%	-20%



The three possible bit spacings are shown for GCR (Group Coded Recording). Only two 0's can appear in a row. Pattern 1 is "11", next is "101" (one zero), finally "1001" (two zeros.) The shaded areas indicate tolerable alterations in the nominal bit times.

REVISED TIMING GIVES BETTER TOLERANCE:



REVISED TIMING

Two years ago I noticed that the phi deck encoding method (GCR) allowed a greater variability in the 1's compared with 101 and 1001. The limiting condition is the boundary between the 101 and the 1001, i.e. being able to discriminate between a fast 1001 at 301 usec and a slow 101 at 299 usec. This limiting condition means that the maximum speed tolerance allowed will be ±20% (bit to bit jitter). I.e. the 250 bit spacing (101) can be as slow as 300 usec, or 300/250 = 1.20 or 20% greater than normal. Similarly for the 375 usec"1001" , 300/375 =.8 or 20% less than normal.

However, the limiting 20% does not apply at the other ends of the bits. My particular self-clocked circuit can receive 1's as fast as one every 2 usec! (This represents 2/125= .016 or 98% faster.) The 1's can be 60% slower than normal. The 101 can be faster than 20% if the 1's 60% tolerance is proportionately reduced, i.e. you increase the tolerance of the 101 pattern at the expense of the 11 pattern. Finally, the 1001 can be much longer than 375 -- as much as 675 with my circuit, or 80% slower than normal.

The revised timing I selected is shown in the diagram. The "11" pattern tolerance is reduced to 36% slower in order to allow the "101" pattern to arrive 32% faster. From observation of the various effects of bit shifting and noise I have determined that the "11" pattern is often slower by approximately 12% due to bit shift. Thus adding an extra 16% on the speed tolerance for this bit pattern would retain the ±20% tolerance desired. Similarly, the extra 12% on the "101" pattern gives a safty margin.

Despite the above tolerances for "11" and "101", remember that the limits are set by the slow "101" and the fast "1001" so the extra tolerance gained by the "revised timing" is quite small on the average (unless you are recording all 1's or fast 101's!).

As an advance warning, the Hughie PLL-replacement circuit is indeed theoretically capable of tracking data with a ± 20 % bit jitter, i.e. bit to bit, a -20% bit followed immediately by a +20% bit. (This is not even possible for the digital decoder.) In actual practice the circuit achieves ± 18.8 % by itself, which is degraded to approximately ±15% bit to bit when the data travels first through the preamp, differentiator, and comparator. Moral: theory is not reality. (A few percent are gobbled by the finite length of each clock pulse, 1.3usec out of 125, and a few more percent are gobbled by the inability of the differentiator to accurately mark the peak of each transition.) (AND ONE-SHOT - CAPACITOR JITTER, ETC.)

Summary H: Replace PLL with one-shot timer circuit which is simple and stable.

SOURCES OF BIT JITTER:

I: ARE THERE ANY MORE? IS IT FIXED?

Phew! I have degutted the whole front end of the phideck and controller in order to obtain as close to 10^{-8} bit error rate as possible. Each part of the solution works fairly well by itself: the new preamps and differentiator and comparator clean up the input signal and make the PLL decode much better. The Time-a-bit circuit can accept the LM324's preamplified head signal with great improvement. But I did all at the same time and vice-versa, ... I do not know what part of all of my fixes is truely responsible for the achievement of nearly 10^{-8} error rate. Speaking as an engineer who makes theoretical rationalizations, I have explained everything (see above 16 pages.) Honestly, now that it is all done and purring along, I feel a little unsure about all of my glib theory.

Nevertheless, it works! I have tested the modified phi deck with a cheap Scotch AVC type no-roller non-shiny-surfaced tape. This C-90 tape as recorded with a standard Phi-test pattern of 00,FF,FE,....03,02,01. I was able to read this tape, crammed with more than 7,542,720 bits, a total of 5.5 times without error, i.e. 41,484,960 bits without error. Adjusting for the 94% efficient phideck format, this means a bit error rate of 2.55×10^{-8}. This value is about 100 times better than the rate I was getting with the same tape before I made my modifications.

With my revised circuitry, the HP counter-timer shows a high degree of bit timing accuracy on the "101" and "1001" bit pattern 250±10 and 375± 5 ! The "11" is still bit-shifty, often showing up as 140 if it follows a 101 or 1001 pattern.

I now keep my TV far away from the phi deck tape heads. However, with the new circuitry's low noise and high common mode rejection, I don't mind if it comes close.

Every day I power up, the phi decks work. Reliable. Alas, slow indeed compared to disks, but given the proper debugged PHIMON or Tom Cox's HI PHI II (preferred) , the phidecks do their job flawlessly... well...

REMAINING PROBLEMS: SOFTWARE AND HARDWARE

J: I'm done!

The delays should be greater so the head guides don't crease the tape every now and then. A more clever search algorithm would not wear out the machinery. A resident PHIMON would not have to overlay so much, would take only about 6K, and could retain the directory in memory until the tape is "closed". Alternatively on that last point, the directory, properly "dated" or numbered could be written following any saved file as well as saved in memory for each deck being used. A pointer for each deck could specify the most recent directory location and allow this directory to be saved at the beginning at the end of the session or when the tape is "unmounted". This latter idea would safeguard the tape in the event of a system or tape crash. Tapes with extra directory blocks would be safe, but slower (possibly not), and smaller (because of all of the directories everywhere.) Perhaps a short systems tape on drive 0 could store all of the directories presently "open".

I suppose if I took another 2 years I could clean up ole Phimon to the modern standards of speed and cheap memory.

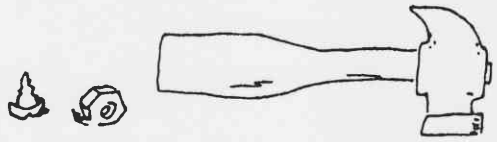
If you are doing this, have you tried the Disk-type of formatted tape, with lots of gaps for writing "in place" without danger? OPUS I believe does something like this. But does OPUS work?

Now that everyone is going to disks, why keep the PHI? Well, a reliable

interchangable phi cassette is much easier to send in the mail, and much much cheaper per byte of storage for archival use. It is slow. But you can back up those expensive disks (if you write the program.) But it is slow. It is cheaper!

Previously storing archival data on a phi deck was equivalent to throwing your data into a slot machine -- or gently placing it in the trashcan. It gave you the feeling of storing all of that good old stuff away for a rainy day -- and subconsciously you were happy that you would never be able to recover the information on that cassette. The compulsive's delight?

Well, now that is all changed. You are going to have to decide voluntarily to throw your data away, because if you keep the cassette , you are going to be able to recover the data -- barring nearby vacuum cleaners and duststorms.



CONSTRUCTION OF HUGHIE'S MODIFIED PHI DECK FRONT-END AND DECODER:

Disclaimer: These construction details are being written after I have finished my construction. I have not used the following notes to rewire someone else's controller so there may be some bugs herein. Please try to understand what you are doing before blindly following my notes. 25¢ credit for anyone who reports bugs found in this tip. Check memos published after Feb. 1980 for updates on this tip.

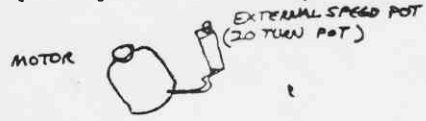
Individuality: I built my modification in 19 different stages. I describe below a combination of what I have done with what I would do for the next one. There are a number of options at your disposal, so please read and reread carefully to plan out your course of action. Perhaps someone will produce a PC board to simplify all of the mess of remodeling.

Compatibility with Unmodified Phi: A large effort was made to minimize substantial changes to the phi deck hardware. I wanted people to be able to return their system to "unmod" should they desire, and I did not want people to have to make many cumbersome changes, especially to the wiring from deck to mainframe. While I know that more extensive changes would produce a cleaner phideck performance, I will also present below a plan of changes that results in slightly greater susceptibility to noise in order to make the phideck cabinet assembly plug-compatible with all other "standard" units.

Things the new circuitry WONT do: Mechanical vibration from the clunk-clunk of head bar motion will take its toll, as will dirt, as will well-used worn-out cassette media. You will still need to adjust the speed of your capstan motor, clean the heads and keep the deck in a clean spot, and will need to eventually replace well-used, nicked, worn-thin cassettes. You will still need to make backup copies of all files to cover you during power failures and hard crashes.

[TOM COX SAYS: DON'T TRUST AN ALIGNMENT TAPE!!]

Those who use their phi decks a lot might think about replacing the capsta. motor with a 115V 60 Hz synchronous motor, available from Triple I, manufacturers of the phi deck drive. Or you might do as Tom Cox, author of HI PHI (upgraded PHIMON), and mount the speed pots outside of the motor housing. You might also drive the capstan motors with 6-8 volts instead of 5 volts. I suggested to Tom Cox that the internal pot be put to one extreme and an external circuit be used (ease of access) since Tom's mod to externalize the speed pot is "tricky."



I have never tuned the speed of my phi deck drives. The modified circuitry will stand a large deviation in speed, but eventually you will experience problems, and will have to tune the capstan speed. See Bridge and DG2.

SUGGESTIONS

STEP ONE:

SUGGESTIONS: Use a line filter (EMI filter) on your CPU. Unplugging a scope was able to glitch my phideck reading.

Add a separate power supply ± 12 V and 5V for the phi deck to use alone. In some cases this step alone cleared up most phi problems. However, if your CPU power supply is hefty or not fully loaded, feel free to add my circuitry without worry. My circuit has its own power supply filtering at the low level stages. EXTRA

NON-COMPATIBLE SUGGESTIONS: Put all phi deck boards and power supply in the cabinet with the phi decks (or in a cabinet underneath) and send only bytes to the CPU I/O ports. You only need 20 lines TTL. Connect these to the backplane like any peripheral (printer, modem , etc.) and drive it with 1½ ports input and 1 port output. This is my favorite solution. You could keep the head leads short, isolate the power supply loading problems, and simplify wiring. yes!

However, it would cost for the extra power supply (and extra box or expansion hardware) and the results would work wonderfully but not be compatible with other phideck cabinets. Yet it could then interface to S100 and other mainframes -- and you could sell it!!!

NOTE: Do not orient the powersupply transformer anywhere near the phi deck heads without considering the magnetic radiation. Usually a certain angle of placement will considerably reduce hum pickup. (The modified circuit is much less susceptible to hum than the old circuit, but no need to push it.)

Use safe wiring of 115VAC with all exposed 115VAC insulated with heat shrink or tape and install fuses.

READ IT OVER

STEP TWO:

Read over the whole construction and try to do it in your mind before beginning. There are several decisions -- like do I add a power supply, do I move the drivers, etc -- and you can make these better after reading the whole thing. Take your time. Use good construction technique -- and if you drill any holes, keep the metal particles out of the rest of the equipment!

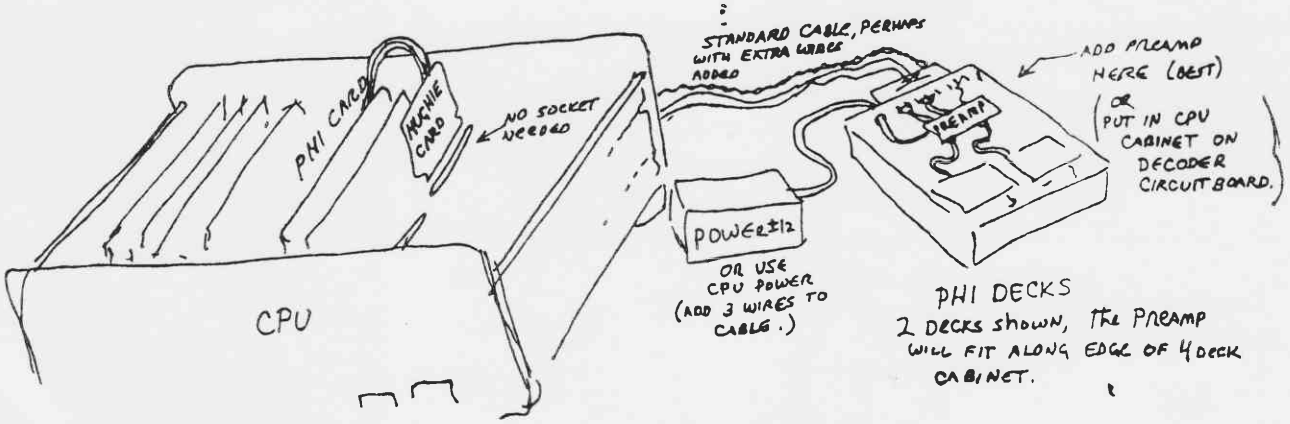
BLOCK DIAGRAM OVERVIEW

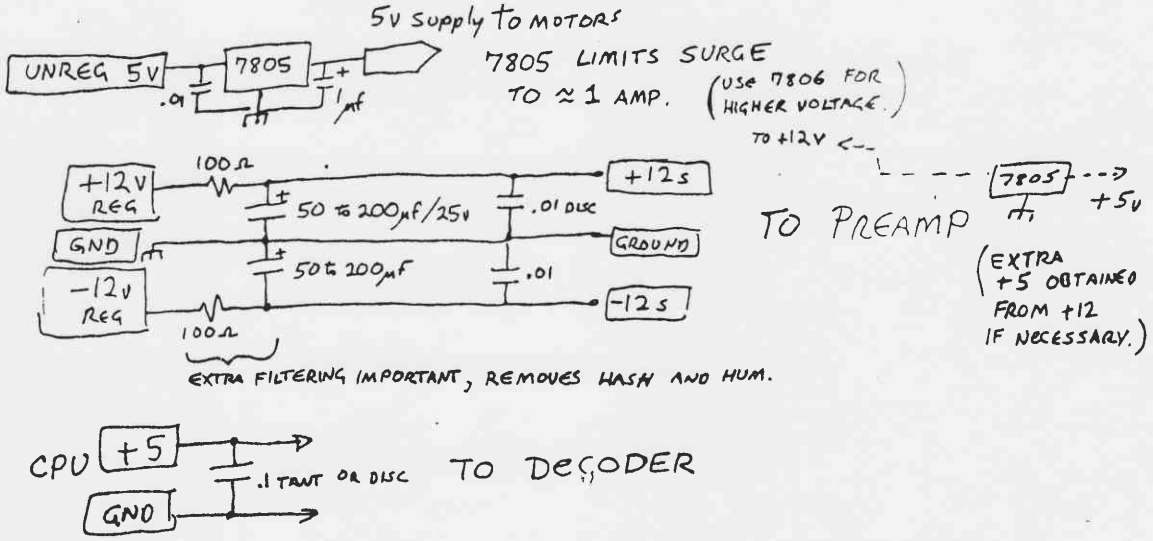
STEP THREE:

Have you read it over?

Construction involves three main parts: Build the Preamp circuit; Build the Decoder circuit; Modify the Controller. The preamp and decoder will work fine and vastly improve the performance of the "A" board -- but if you have an"A"board you must ignore the details of Part 3, and see the appendix after you read about the general theory. Part 3 is written for the"B"and"C"boards.

LET I CAN FIND MY SCHEMATIC!!

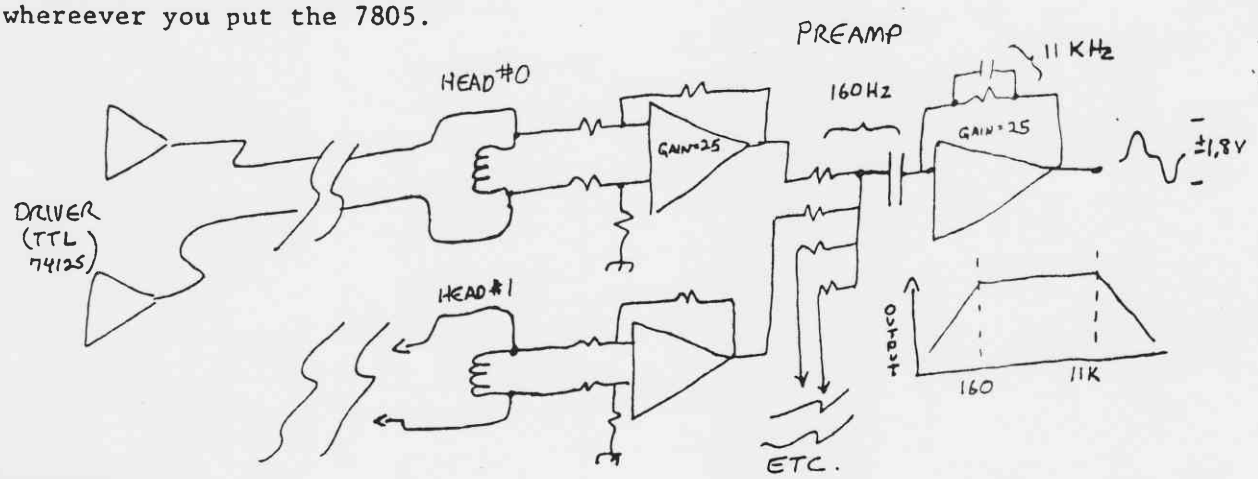




POWER SUPPLY OVERVIEW: Above shows how ±12 v is doubly filtered whether or not the ±12 comes from the CPU or from a separate supply. This double-filtered supply is only for the Preamp and reduces noise by a very noticable amount.

DECIDE: If you put the preamp in the PHI cabinet, you will need a supply of +5 for the comparator. I got this supply via a 7805 attached to +12. Alternatively you can send a high level analog signal from the preamp into the CPU cabinet and do the schmitt triggering there.

OPTIONAL: At the top you can see how a 7805 can be used to limit the turn-on surge (that can be as great as 4 amps for four drives) to about 1 amp. The 7805 should be attached to the UNREGULATED 5 V (i.e. about 8V) and will cool down your +5 regulator by about 1 amps worth (3 watts), and put this heat whereever you put the 7805.



PREAMPLIFIER:

Each head has its own differential preamplifier with a gain of 25, raising the input head voltage of ± 3mV to ± 75mV. Common mode noise is reduced about 40-60db (100 to 1000 times). The head preamps are summed by a band pass filter with 6 db/octave rolloffs at 160Hz and about 11,000 Hz. Hum and low frequency noise from dropouts and dirt are reduced and centered about zero by the AC coupling. TV and TTL noise are reduced by the high freq rolloff.

[My favorite circuit gates the unused heads fully off so that the noise from the unused heads does not affect the performance of the remaining head during a read. However, to gate the signal off you need a higher level of signal (say 1 to 5 volts) and this takes an extra amp for each head, plus some CMOS gates. I used LM356 low noise op amps for the preamps and, feeling protected, gave up the idea of gating each head.]

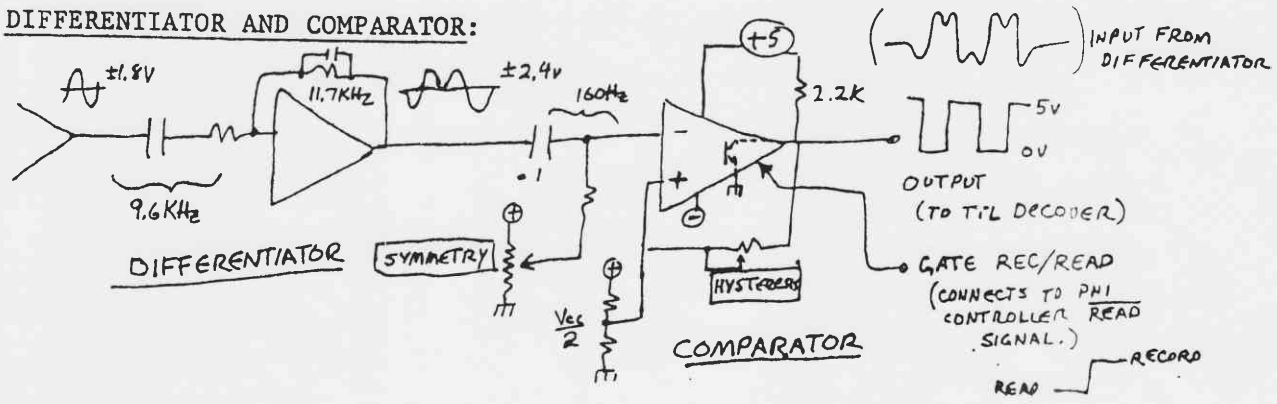
Note how the DRIVER TTL gates have a long way to go to connect with the head. If you DECIDE not to put the TTL drivers next to the Preamp in the Phideck cabinet, then you might as well put the preamp next to the TTL and hope it works. I'm going to try this on my next modification.

Noise values for the preamp: Referenced to the input (i.e. equivalent input noise) the preamp produces the following noise vs input condition values:

	NOISE VOLTAGE	
Shorted input	4 uV peak to peak	(estimated from scope face)
Head alone	8.3 uV	(all values referred to input)
Head with Driver TTL	20 uV	
One side shorted	33 uV	(shows response with no CMMR, i.e. non-diff amp.)

This shows how the long leads which extend from the phideck driver TTL in the CPU cabinet more than double the noise output of the preamp. Nevertheless, since the typical output voltage of the head is 6mV peak to peak, the S/N ratio (signal to noise ratio) is fairly good, ranging from 6000/8.3 = 722 (57 dB) with no Driver TTL to 6000/20 = 300 (49 dB) with Driver. These figures do not include noise from tape hiss or dropouts or tape misalignment and so are a bit high.

DIFFERENTIATOR AND COMPARATOR:



The differentiator measures the slope of the input wave. When the wave peaks the slope = zero, so the differentiator output goes to zero at the peak (actually just after it.)

The comparator is AC coupled to the differentiated wave and detects when the signal crosses zero. Hysteresis is added in the amount of about 5% to 10% in order to prevent false crossing due to noise, i.e. once the input goes positive, it needs to return to its previous voltage plus 5% lower in order to trigger the comparator negative. 5% of ±3 mV (head voltage) is 150 uV. Thus 5% hysteresis eliminates the 20uV noise, the tape hiss, and still leaves a safety margin. However, if noise is still present on the head signal, it will cause timing variations as to when the comparator triggers. The hysteresis does not improve timing -- it makes it a slight bit worse, depending on transition times and the noise. The hysteresis does prevent the comparator from double or triple triggering due to small quick ± excursions of a noise signal superimposed on the desired head signal.

(The old circuit used 20 to 40% hysteresis, resulting in timing errors due to amplitude variations.)

Symmetry is important. The unmodified circuit was off by about 3 to 8% and lowered the maximum possible speed tolerance by this amount. AC coupling in Hughes's Mod and the SYMMETRY control permit accurate adjustment of symmetry so that the positive and negative excursions of the input wave are treated alike.

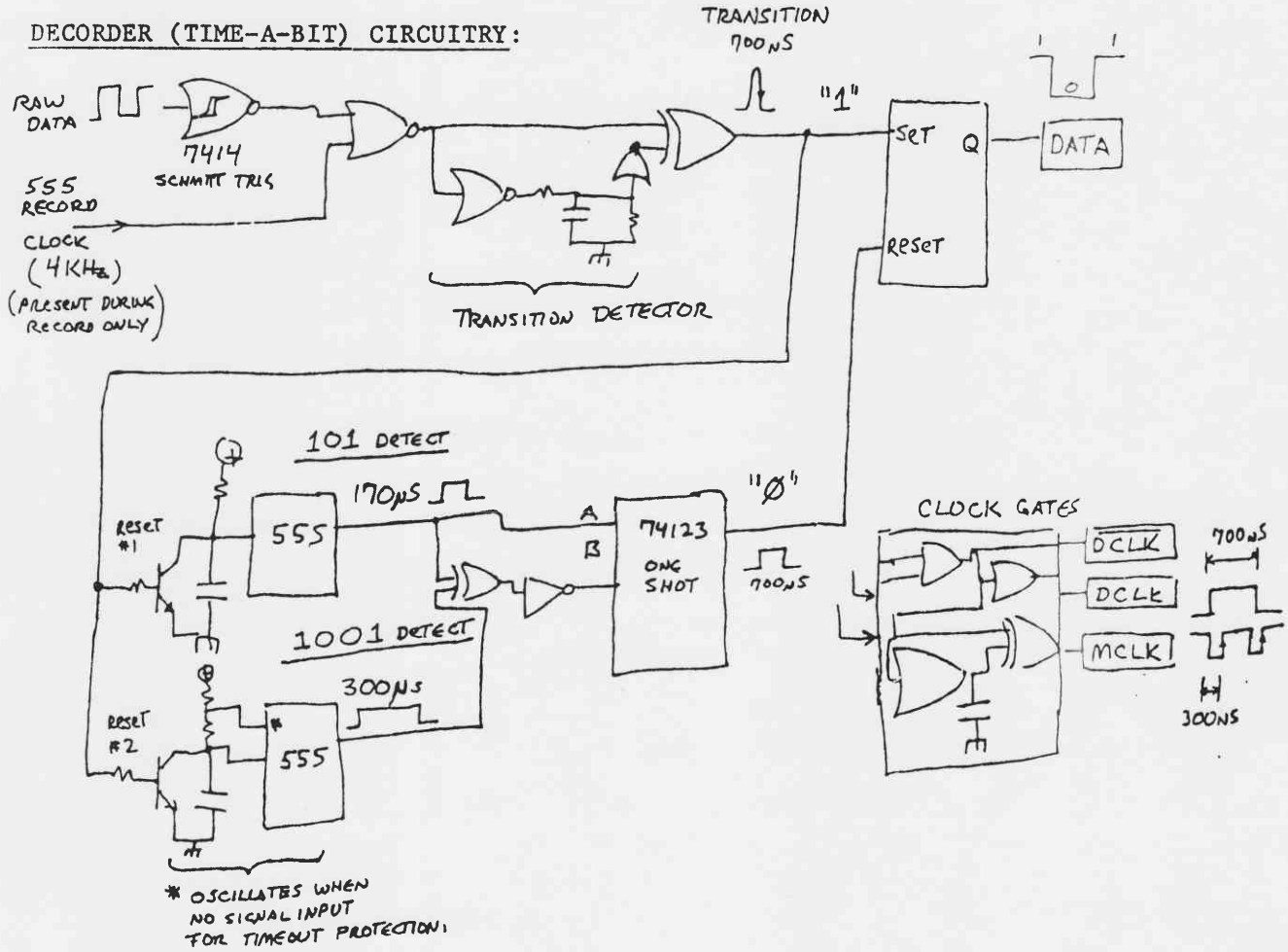
The LM311 comparator has an open collector output. 2.2K provides a proper speed risetime for TTL of about 1/2 usec . The emitter of the output transistor is brought out and connected to ground, giving 0 to 5 volts output compatible with TTL. The negative supply of the 311 is connected to -12 so that large negative excursions of the input wave do not invert the proper operation of the 311. Positive excursions can reach +18 volts

INPUT

and cause no problems even though the V+ supply is only 5 volts.

To adjust symmetry, set hysteresis to minimum and adjust the symmetry pot until you obtain nice square waves on the smallest possible signal. (Generate a 4KHz sine or square wave by CPU or Audio generator -- or short the head with a 1 ohm resistor!) See Construction Details.

DECORDER (TIME-A-BIT) CIRCUITRY:



A schmitt trigger shapes up the input signal (necessary if you do not build the preamp part). An OR gate passes either the record clock or the RAW data depending on the READ signal obtained from the Phi deck board. The READ ("not read"), when low, forces the RECORD CLOCK to 0. When hi (in RECORD) the LM311 is gated off by its "strobe" input. This gating is required in order to be compatible with the rest of the phi deck TTL decoding scheme.

THe decoder is simple. An exclusive OR and inverter detect the transitions of the input wave. Note that both negative and positive transitions result in a positive output pulse of about 700 nanoseconds. The RC circuit sets the delay for pulse width, and 7414 schmitt triggers are used to clean up the RC signal. (7404's might oscillate at the transition.)

By definition in GCR, every transition is a "1", so the transition pulse sets a flip-flop output DATA to "1". The transition pulse also resets the two 555 one shots which are attempting to time out.

If too much time elapses between transitions, first one and then the other 555 times out and triggers a one-shot (74122 or 74123) to reset the DATA flip-flop to "0". The first zero occurs after 170 usec elapses, the second zero occurs after 300 usec. If the signal disappears entirely, as it might after the last block on a tape, the second 555 begins to oscillate with a 300 usec period in order to clock the last byte through the phideck controller card ROM and into the output port.

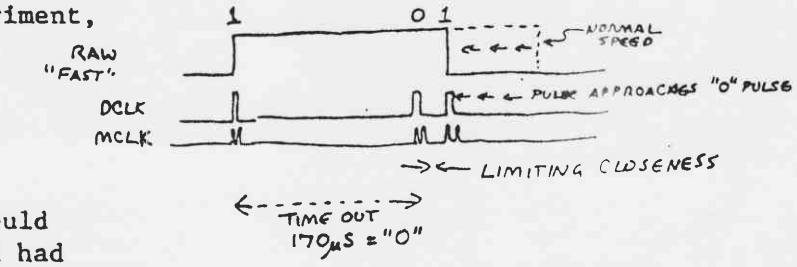
The CLOCK GATES are not detailed completely. These gates comprise inverters and another transition detector which outputs a 300 nSec pulse. The clock gate circuit is required in order to provide the phi controller card with DCLK and not-DCLK and MCLK. Since the original design synchronized a eparate "read clock" to the data it was easy to produce the DCLK and MCLK; however, my new design is self-clocking and produces the clock from the data itself, so the CLOCK GATE circuit reconstructs or synthesizes the appropriate clocks.

DCLK is pulsed every time a data bit is detected, i.e. it is the DATA CLOCK. Either a transition ("1") or a time-out ("0") pulses the DCLK output.

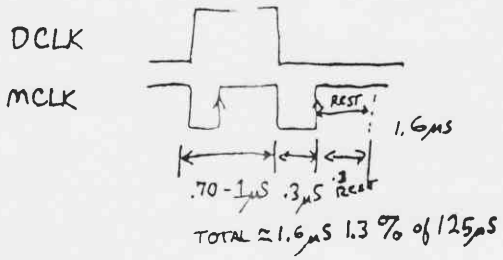
MCLK is twice the DCLK and is used by the phideck controller to OR and AND together with DCLK in order to time and synchronize the ROM and other decoder circuitry. My cheap and dirty way of creating MCLK is to double the frequency of DCLK. I invert DCLK and detect its positive and negative transitions. At each transition a pulse is sent out which is smaller than the DCLK from which it is derived, 300 nS versus 700 nS.

Note that the DCLK and MCLK must be short. The DCLK+MCLK time limits the maximum possible speed tolerance. 20% is possible, but since the DCLK and MCLK take 1.6 usec out of a bit-to-bit time of 125 usec., 1.3 % speed tolerance is lost. The limiting condition is when a transition "1" immediately follows a time-out "0"; each event results in a 1.3usec clock pulse stream and since they cannot overlap, that 1.3 usec (plus a bit) is lost.

It is possible to reduce the time of DCLK and MCLK, but I was afraid of them being lost on the lines which connect the Hughie board to the Phideck board. Just being "safe" and I could see the 1.3usec pulses on my #2(slow) scope! (If you want to experiment, just reduce the value of the capacitors in the transition detectors and in the one-shot 74123 by a proportional amount, say 1/2.)



[My time-a-bit circuit would be much more complicated if I had to recover high speed disk signals which come every 2 usec. This would require MCLK pulses at about 5 nS!]

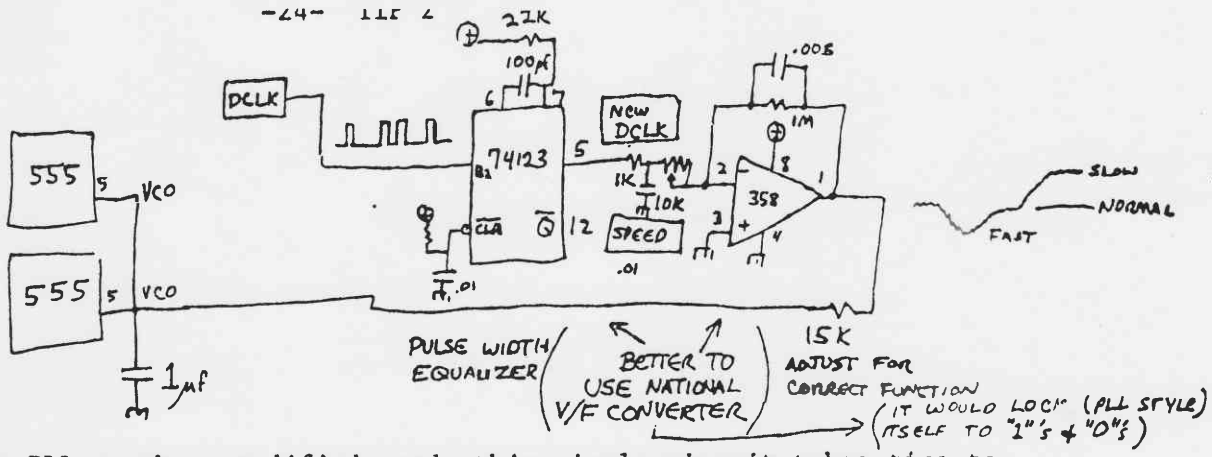


SLOW SPEED VARIATION ACCOMODATION and SPEED ADJUSTMENT.

Since 555's are used for the record clock and are used to time-out the 0's in the read data stream, a single voltage can be applied to all 555's VCO input in order to change their timing. The voltage at pin 5 is normally 2/3rds Vcc ; if it is externally lowered, the frequency increases and the one-shot timing decreases. I have tried attaching a variable resistor from pin 5 to ground in order to change all three 555's at once and have observed fairly close matching, i.e. over a small range, the Record Clock changes in proportion to the two time-out 555's.

[The close tracking of the 555's led me to conceive of a phi-deck controller that COULD READ ANY TAPE ±100% speed! A frequency-to-voltage converter (like the one National sells for $2.50) would convert the DCLK pulses and integrate over a 100:1 period to obtain a voltage equal to the average speed of the phi-deck in use. This voltage would adjust the timing of all of the 555's so that they properly decoded the incoming data. Sort of a backwards PLL? At any given average speed, the decoder circuit would accept bit-to-bit speed variations and jitter of ±20% (or ±15% in practice.)

A simple version of this circuit is shown for your interest.

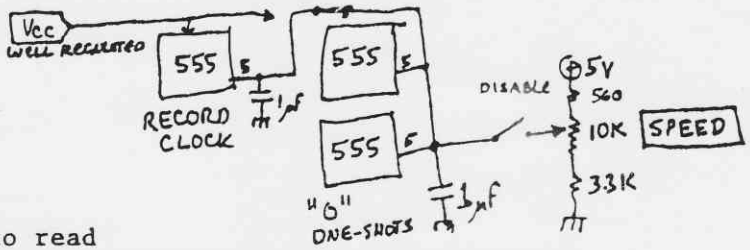


[Like the PLL on the unmodified card, this simple circuit takes time to synchronize with the data. Alan Neil said that his PLL circuit sometimes cannot synchronize during the first 70 "1"'s since there is no data which preceeds the first block. I suggested writing a "dummy" block at the front of the tape before the directory begins -- but this fix is not the correct way to solve the problem. The correct way is to have a "normal" setting at which the PLL rests with no input or with normal speed input. Only when fast or slow data is received would the PLL begin to adjust. Such a PLL would be more complicated than the one on the phideck board.

[The circuit above, with no input, rests at one extreme, depending on the particular opamp and powerup conditions. When data arrives, DCLKs (corresponding to each transition "1" of the data or timed-out "0") are first equalized in width by the 74123 (so "0" and "1" are equal in ability to change 555 speed). The equal width pulses are filtered to slow the rise time so the 358 can follow them. The 358 is connected as an integrator with a slow decay. Approximately 100:1 ratio, i.e. averaging time-constant equals 125us x 100 or 12.5milliseconds. The 358 inverts the signal so that its output voltage decreases for increasing input frequency. This voltage is applied to the 555's VCO PIN.

[PROBLEMS: The simple circuit is non-linear; it simply integrates pulses to obtain the average value which does not change linearly with frequency at the extremes. In addition, the 555's VCO input is non-linear; equal steps in voltage results in different sized changes in frequency. The 15K resistor needs to be adjusted for optimum results. And, last and worst, the circuit can latch up at the high speed extreme by getting the "0" detectors to supply the majority of pulses without regard to data from the tape! But it is a simple circuit; I built it; I disconnected it.]

SIMPLER STILL: A SPEED POT YES? NO?



It seems a shame to put in 555 one-shots because they have a VCO input and not to FULLY use them. Nevertheless, I recommend not using the VCO PIN. The "speed pot" mod shown here is good if you expect to read wildly varying tapes or wish to read those old tapes you wrote with a slow or fast motor. Unfortunately, this mod makes the 555's more sensitive to Vcc voltage variations, and since the Vcc comes from the CPU and is usually quite loaded, you might increase the basic 1% jitter of the 555 to 2 or 3% or more. This jitter will degrade your overall speed tolerance capability . If you need "speed adj", why not put in a dip switch or jumper so that it can be disabled. (Reason: the internal resistor divider of the 555 is well matched; adding external voltage to this divider via pin 5 alters the matching so that the 555 becomes much more sensitive to voltage supply and temperature changes.)

If you do not use the speed pot for experimenting with recording at a

higher density, i.e. you use it only for reading, then do not connect the wire to pin 5 of the record clock 555. Leave it alone. Differences in matching of the 555's make such a connection a little risky; i.e. the internal resistors of one 555 might degrade the matching for another 555.

(The purist might connect each 555 pin 5 to its own capacitor to ground, and skip the idea of speed change entirely -- or might put in a separate dip switch for each pin 5.)

When it comes to reliability and archival data, I am a purist. I want my phidecks to be reliable. I don't care how slow they are -- I want my program (or data) back once I've put it on the cassette!

SUMMARY OF THE OVERVIEW:

Simplified schematics have been shown and described for each stage of the modified preamp and decoder. The mods for the phi-deck controller have not been covered. Additional ideas have been mentioned.

PHI DECK CONTROLLER BOARD B and C: Modification overview

As I mentioned earlier, DATA , DCLK, not DCLK, and MCLK signals are required by the ROM decoder section of the unmodified controller board. It is necessary to cut the traces at 4 places and to connect the above signals in replacement.

The preamp-decoder needs two signals from the controller board, a $\overline{\text{READ}}$ signal and a RECORD CLOCK. The $\overline{\text{READ}}$ signal gates the RECORD CLOCK on and off and simultaneously gates the RAW DATA off or on so that during record, the Hughie-decoder produces DCLK and MCLK from the 4KHz record clock, while during reading, the Hughie-decoder produces DCLK and MCLK from the RAW DATA. DCLK and MCLK are used to clock bits through the Group-Coded-5-bit-pattern-to-Hex-4-bit-pattern-converter; they also clock the bits from the deserializer into the data output port -- in fact, they clock everything through everything!

GCR (GROUP CODE): What?

On the right is a table showing how the 4 bit Hex data patterns are converted into 5 bit GCR patterns for recording on tape. The GCR conversion is quite logical. Start with the sixteen hex values (shown in binary) and put a "1" on the far left for the first 8, and put a "0" on the left for the last 8. You now have 5 bits for each of sixteen values. But since GCR requires no more than two zeros in a row or two zeros at the end, you must delete all such bit patterns. This wipes out the 0,1,4,8,and C. Replace these with left-over but acceptable 5 bit patterns.

WHY NOT FM OR MODIFIED FM? LIKE THE DISKS?

FM is not efficient, but double density MFM is slightly more efficient than GCR and much easier to decode and detect. Given what we know today, it would probably be the best choice for an encoding scheme. I myself have not been able to find out in what ways GCR is better than MFM (please tell me) but GCR is used on all of the

GCR: Group Code Recording is a recording technique in which whole groups (or fields) of a user's data (called "data groups") are converted into groups of tape data (called "storage groups"). The conversion ratio is 4 bits of "data" to 5 bits of "storage" using a look-up table as illustrated in Fig. 1. A decoder at the read output converts the data back to its actual value. No more than two consecutive '0s' can be recorded, eliminating the need for clock pulses.

Data Group to Storage Group Conversion

Data Groups (4 bits)	Storage Groups (5 bits)
0000	11001
0001	11011
0010	10010
0011	10011
0100	11101
0101	10101
0110	10110
0111	10111
1000	11010
1001	01001
1010	01010
1011	01011
1100	11110
1101	01101
1110	01110
1111	01111

high speed IBM 6250 bits-per-inch tape drives, so it must have some merits. Perhaps it is more self-clocking than MFM? More tolerant? Maybe the big companies use GCR because DIGITAL GROUP used it? ho ho ho. Commodore in their PET disk drives uses it (or at least they were going to use it; one of their disk board designers was trying to sell his phidecks .)

PHI DECK CONTROLLER BOARD A:

My modification circuits will cost less than the upgrade board DG sold for $70 to convert the A board to a B or C board. In many respects, the Performance will be the same after Hughie-mods are complete. Some motor control functions and logic have been modified in the B and C boards -- but I am not an expert on this. I will say that my OLD A board worked just fine around a very narrow speed tolerance range.

DISCLAIMER:

My A board has no chips on it. All have been moved to the C board. So the changes recommended for the A board are theoretical. Check a Memo or Tip published after Feb 80 in order to catch any bugs. There is enough similarity in the two boards that the changes should work fine. Most of the changes (A to C) were to the phase lock loop circuitry and other front-end decoding -- most of which has been replaced by Hughie's decoder.

CONSTRUCTION DETAILS:

READ THIS OVER AT LEAST ONCE BEFORE STARTING:

The following construction details are for the most simple Hughie-mods and require the least amount of construction. The preamp and the decoder are on one board and reside inside the CPU cabinet, connected to the phi-controller card by about 10 wires.

Better performance will be obtained by separating the preamp, differentiator, and comparator and placing them in the same cabinet with the phi deck drives. You will have to run ±12v to the cabinet, as well as the RECORD CLK and the not READ signal. (DIAGRAMS AND LAYOUT SUGGESTED BELOW.)

Best performance will be obtained by having all phi deck electronics installed in or under the phideck cabinet together with a separate power supply.

The "better" and "best" options are not detailed; you might wish to build the preamp and decoder on a perf board that can be cut down the middle in case you later desire to move the preamp into the phideck cabinet.

MOTIVATION:

Say a prayer. My phideck mods should improve the bit-rate-error by a factor of 10 to 100. Estimated bitrate error is 2.5×10^{-8}.

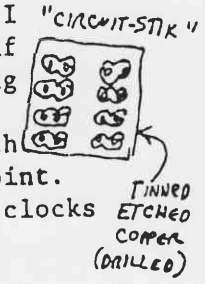
THREE STEPS:

1) build preamp and decoder; debug with scope, audio oscillator, and counter.
2) cut traces and otherwise modify Controller Board.
3) connect results of step 1 to results of step 2. DEBUG and enjoy.

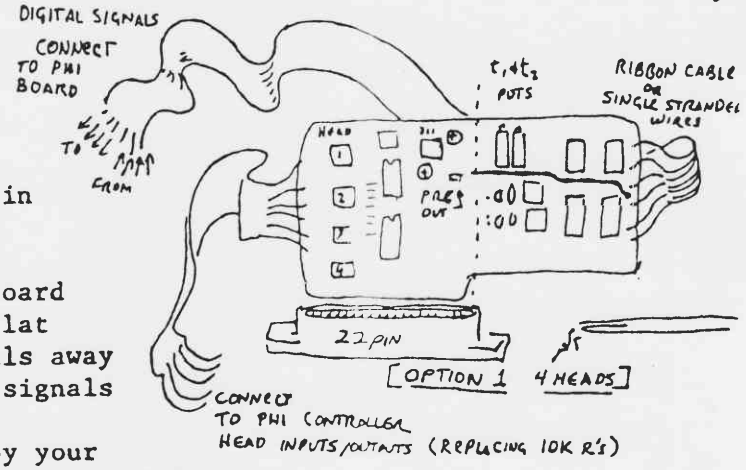
BE BRAVE, TURN THE PAGE →

STEP 1: BUILD PREAMP AND DECODER.
(refer to schematic placed at the end of this Tip.)

Follow the schematic closely and build circuit on a .1" hole perfboard. I prefer glass-epoxy Vector board. Use wire-wrap for digital sections if you must, but a neater, thinner job will result with point-to-point wiring and "Circuit-Stik" type tinned pc subunits for each IC socket.



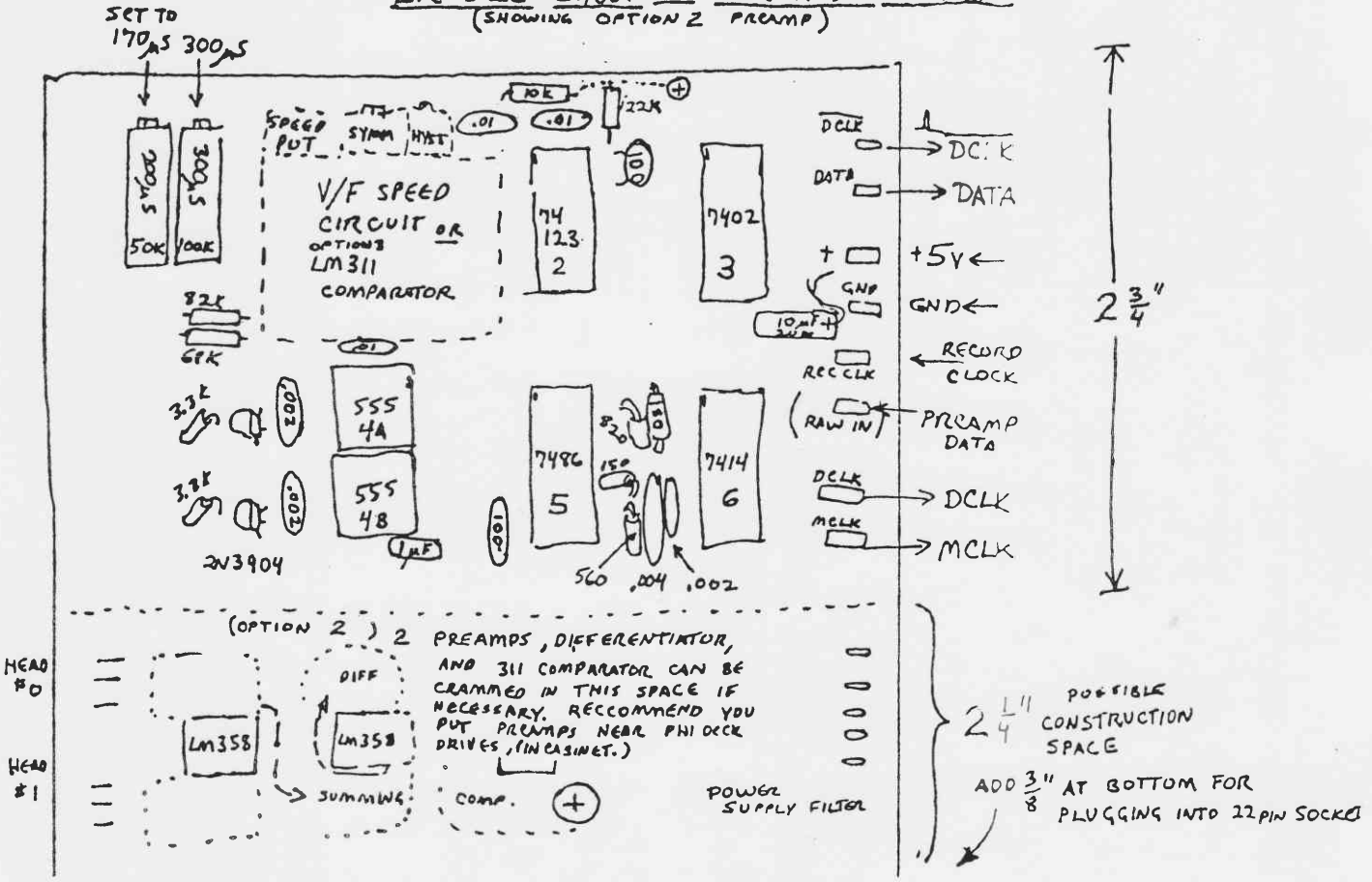
Lay out the location of each socket according to the diagram. Attach the Circuit-Stik at each location. Solder in sockets. Wire point-to-point. Be neat. Do not use sockets for the transistors. Bring out outputs and clocks to Vector terminals or wire loops that you can solder to.

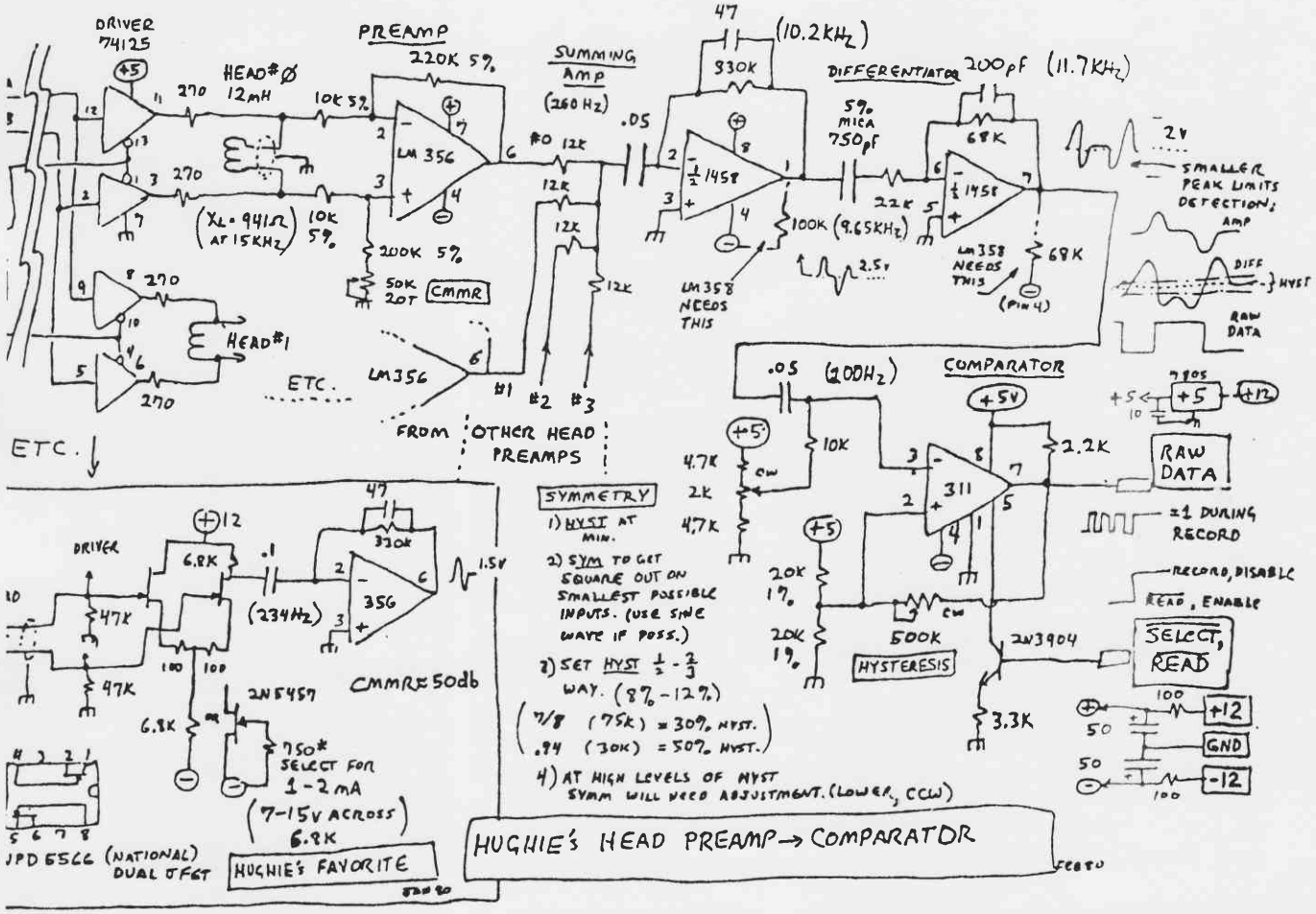


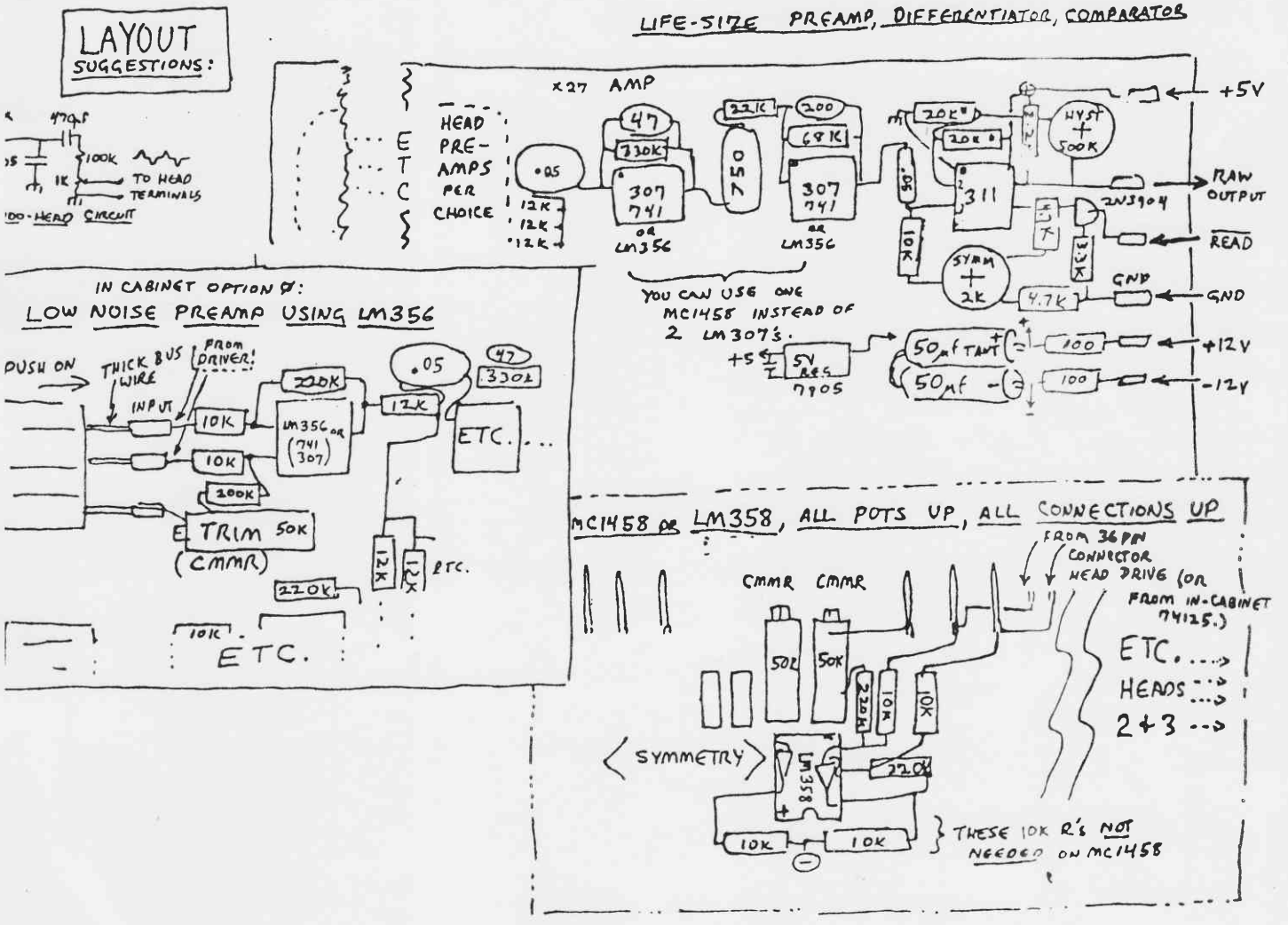
The Hughie-board I built plugs into an unused 22pin connector near the phi board. No power is taken from the 22pin connector so any kind of mounting will work. +12 and +5 are taken from the controller board. Connect the Hughie-board to the controller with stranded wire; flat cable will do, but keep low level signals away from digital signals. Separate analog signals from the digital by two grounded wires. Keep the length as short as permitted by your CPU cabinet layout, but not so tight that they caress nearby boards and capacitively couple in stray signals. Note that all pots are oriented pointing upward for ease of adjustment.

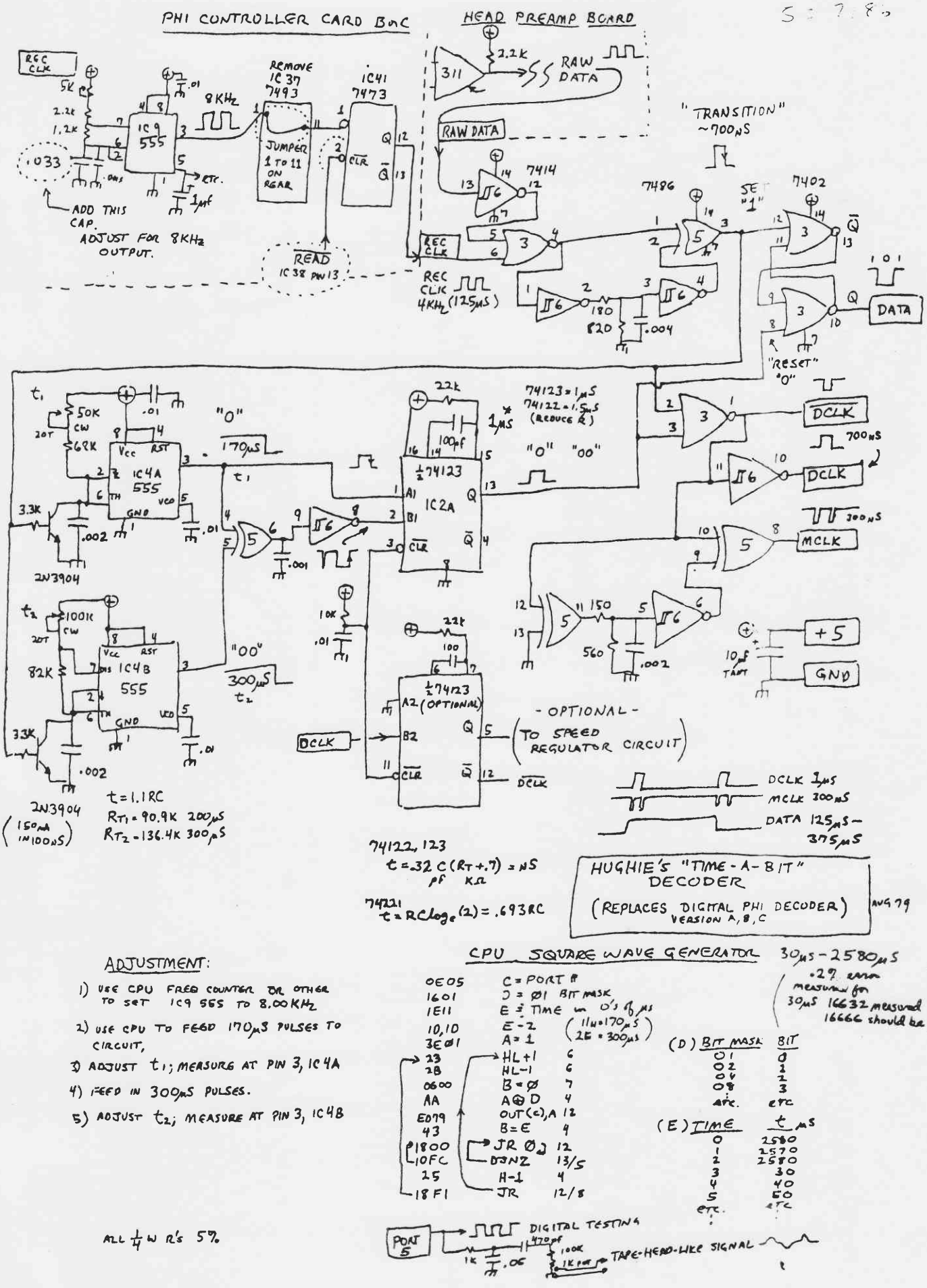
The preamp section is to the left, the decoder to the right. The board can be cut down the middle for translocation. Other arrangements are suggested below.

LIFE-SIZE LAYOUT OF "TIME-A-BIT" DECODER
(SHOWING OPTION 2 PREAMP)




DRIVER 74125
PREAMP
220K 5%
HEAD#Ø 12mH
10K 5%
LM 356
SUMMING AMP (260 Hz)
.05
47
330K
(10.2KHz)
DIFFERENTIATOR
200pf (11.7KHz)
5% MICA 750pf
68K
1/2 1458
22K
100K (9.65KHz)
LM358 NEEDS THIS
2.5v
LM358 NEEDS THIS (PIN 4)
68K
2V
SMALLER PEAK LIMITS DETECTION; AMP
DIFF
HYST
RAW DATA
(XL = 941Ω AT 15KHz)
10K 5%
200K 5%
50K 20T
CMMR
HEAD#1
ETC.
LM356
#1 #2 #3
FROM OTHER HEAD PREAMPS
ETC.
.05 (200Hz)
COMPARATOR
+5V
+5
+12
7805
10K
4.7K
2K
4.7K
CW
311
2.2K
RAW DATA
≈1 DURING RECORD
20K 1%
20K 1%
500K
HYSTERESIS
2N3904
RECORD, DISABLE
READ, ENABLE
SELECT, READ
3.3K
+12
GND
-12
SYMMETRY
1) HYST AT MIN.
2) SYM TO GET SQUARE OUT ON SMALLEST POSSIBLE INPUTS. (USE SINE WAVE IF POSS.)
3) SET HYST 1/2 - 2/3 WAY. (8% - 12%)
(7/8 (75K) = 30% HYST.
.94 (30K) = 50% HYST.)
4) AT HIGH LEVELS OF HYST SYMM WILL NEED ADJUSTMENT. (LOWER, CCW)
DRIVER
6.8K
.1
(234Hz)
47
330K
356
1.5V
47K
47K
100
100
2N5457
CMMR≈50db
6.8K
750* SELECT FOR 1-2mA
(7-15V ACROSS 6.8K)
HUGHIE'S FAVORITE
UPD 5566 (NATIONAL) DUAL JFET
HUGHIE'S HEAD PREAMP → COMPARATOR


LAYOUT SUGGESTIONS:
470pf
100K
1K
TO HEAD TERMINALS
100-HEAD CIRCUIT
LIFE-SIZE PREAMP, DIFFERENTIATOR, COMPARATOR
x27 AMP
HEAD PRE-AMPS PER CHOICE
ETC
.05
12K
12K
12K
47
330K
307 741 OR LM356
22K
200
68K
307 741 OR LM356
20K*
20K*
HYST 500K
311
+5V
RAW OUTPUT
2N3904
READ
10K
SYMM 2K
4.7K
GND
GND
YOU CAN USE ONE MC1458 INSTEAD OF 2 LM307's.
+5
5V REG 7805
50μf TANT
50μf
100
100
+12V
-12V
IN CABINET OPTION Ø:
LOW NOISE PREAMP USING LM356
PUSH ON
THICK BUS WIRE
FROM DRIVER!
INPUT
10K
10K
200K
220K
LM356 OR (741 307)
.05
47
330K
12K
ETC.
TRIM 50K (CMMR)
12K
12K
ETC.
220K
10K
ETC.
MC1458 OR LM358, ALL POTS UP, ALL CONNECTIONS UP
FROM 36 PIN CONNECTOR HEAD DRIVE (OR FROM IN-CABINET 74125.)
CMMR CMMR
50K
50K
220K
10K
10K
ETC.
HEADS
2+3
SYMMETRY
LM358
220K
10K
10K
THESE 10K R's NOT NEEDED ON MC1458



ADJUSTMENT:

1) USE CPU FREQ COUNTER OR OTHER TO SET IC9 555 TO 8.00KHz
2) USE CPU TO FEED 170μS PULSES TO CIRCUIT,
3) ADJUST t_1; MEASURE AT PIN 3, IC4A
4) FEED IN 300μS PULSES.
5) ADJUST t_2; MEASURE AT PIN 3, IC4B

ALL ¼W R's 5%

CPU SQUARE WAVE GENERATOR

30μS – 2580μS
.2% error
measured for 30μS 16632 measured 16666 should be

```
0E05   C = PORT #
1601   D = 01 BIT MASK
1E11   E = TIME in 0's (11=170μS, 2E=300μS)
10,1D  E-2
3E01   A=1
23     HL+1      6
2B     HL-1      6
0600   B=0       7
AA     A⊕D       4
ED79   OUT(C),A  12
43     B=E       4
1800   JR 02     12
10FC   DJNZ      13/8
25     H-1       4
18F1   JR        12/8
```

(D) BIT MASK / BIT

BIT MASK	BIT
01	0
02	1
04	2
08	3
etc.	etc

(E) TIME / t mS

TIME	t mS
0	2580
1	2570
2	2580
3	30
4	40
5	50
etc.	etc

CONNECTIONS:

HUGHIE-MODS IN/OUT LEADS (SEE PICTURES)

1) +5 HOLE IN +5V BUSS
2) GND GND TRACE (REAR, TOP OF BOARD)
3) $\overline{\text{READ}}$ IC 4 PIN 8
4) REC CLK IC 41 PIN 12
5) DCLK IC 50 PIN 2
6) $\overline{\text{DCLK}}$ IC 18 PIN 1
7) MCLK IC 50 PIN 3
8) DATA IC 30 PIN 2

POSTSCRIPT: I strongly recommend you put the preamp-thru-comparator into the phideck cabinet. If you leave the phideck controller board in the CPU cabinet, you must leave the Hughie-time-a-bit decoder in there also. The MCLK and DCLK signals are very (QUICK) short and should not travel far. I do not like to save chips but like the front end of my phidecks to be very high quality. Thus I chose to use one LM356 for each head in my system. The LM356 is super low-noise, low bias, and fast, and not too expensive. Nevertheless, the dual 358 or 1458 can substitute for the op-amps shown in the schematics. The LM358 requires a load resistor tied to V- to prevent crossover distortion when using dual supplies. (The LM1458 requires none.) Sample layouts for the preamp are shown in the previous pages.

While the head drivers can remain on the phi controller board, you will probably get a reduction in noise (especially transient pickup) if you place the driver 74125 in the phideck cabinet. If the head drivers stay with the controller board, make sure the leads to the preamps are evenly twisted or shielded like the head leads themselves.

STEP 2: MODIFY THE CONTROLLER BOARD (see diagrams on next pages)

Above is a list of the 8 signals which will connect to a modified controller board. It is relatively easy to reconvert the board back should you wish to (horrors!). (In fact, reconversion-ease was the reason I bend pin 3 of IC 26 out of its socket instead of cutting a hard-to-get-at trace.) Refer back to the above list when the following modifications are complete:

2A: STILLSON FIXES: PULLUP THE UNUSED INPUTS

On the next page is a diagram of the Stillson fixes, (extracted from Tip 2.5 "Quick Fix".) All unused 7474 and 7473 pins are pulled up to 3.5V formed by a resistor divider of 2.2K and 4.7K with a .01 cap filter. See diagram.

B,C BOARD CONNECTIONS		A BOARD CONNECTIONS	
IC	pins to pullup	IC	pins to pullup
3	1,10	4	1,10
4	1,4,13	6	1,4,13
21	1	14	1
42,55	optional 1,4,10,13	34	1,4,11,12
41	3,14 (2 to IC38p13)	41,42	1,4,10
50	1,4,13 NOTE	43	3,7,14

OTHER CONNECTIONS

ADD A 2.2K 1/4w R from pin 29 to pin 31 on front of board in holes of gold fingers.

HEAD DRIVER R'S= 560ohms
B,C: R6,9,20,23
A: R17,20,23,26

2B: FIX THE SCHEMATIC:
Using a fine pen and the enclosed schematics and changes, change your own schematic so it agrees with what you have done or will do. NOTE: B,C schematics have an error: IC28, 74125 pin 3 should be pin 13 on the control gate.

2C: REMOVE, CUT, AND JUMPER: (see diagrams!)
REMOVE IC 10, 11, 13, 37,42, 55. (7400,358,324,7493,7474,7474).
(REMOVE 10K head-preamp input R's, optional; see note on diagrams.)

CUT WITH XACTO KNIFE: on board bottom: IC18, pin 1 ($\overline{\text{DCLK}}$); IC 41, pin 12 (Record clock), and on the top of the board, IC 50 at pin 10 and near pin 4 (see diagram on this one!). FOUR cuts should have been made. Use lens to check for clean cuts.

JUMPER: (rear of board) IC 37 pin 1 to pin 11 (IC37 has been removed.)
IC 38 pin 13 to IC 41 pin 2 ($\overline{\text{READ}}$ resets flip flop.)
Diode D19 cathode (bar) to +5 trace. (DISABLES FET.)

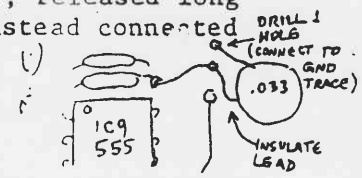
2D: Remove IC 26, bend pin 3 outwards and replace with pin outside socket. This disconnects the old "RAW DATA" driver.

2E: Attach 2.2K 1/4w R from pin 29 to pin 31 of 36pin connector fingers. Solder resistor in the plated-thru holes connected to the gold fingers.

2F: Optionally attach 10 to 50 uf tantalum or electrolytic to +5 and GND to provide additional storage and protection against transients.

[2G: During TESTING; you can override "overrun" by putting pin 8 and 10 of IC 21 outside of socket and pulling pin 10 to + thru a 10K resistor.]

2H: 555 TIMER MOD: attach a .033 cap in parallel to the existing timing (C12) capacitor of IC9. Ideally it should be a low drift type (mylar) but I used a disc capacitor. You will have to drill a #68 size hole to accomodate the extra capacitor or else be more clever or precarious than I was. With the new capacitor, the 555 will oscillate in the 8 KHz range. [You can also remove the .0015 C12 and replace it with squenched-up leads from the .033 capacitor.]



2I: If you have already made the Stillson fixes from my tip 2.5, released long ago, then note that pin 2 of IC 41 is no longer pulled up but instead connected to pin 13 of IC38.

STEP 3: CONNECT, DEBUG, AND ENJOY: (adjustments too.)

Make the connections shown on the previous page using stranded wire, about 26 to 30 guage or so. Bundle them loosely or use ribbon cable. Attatch to Hughie decoder leaving enough room to reach from mounting spot to controller board.

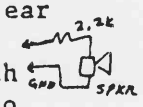
Following diagrams and schematics, connect your PREAMP-COMPARATOR circuit to the rest. ±12 volts will be needed to power the preamp. The not-READ signal must be supplied to the board; the RAW DATA line is returned to the decoder circuit: FOUR lines total. Before connecting RAW DATA, you might want to debug the decoder board. Connect with stranded cable (see diagram.)

After the preamp is connected (via cable or direct) to the decoder, plug the decoder and phideck boards into the CPU slot and begin the adjustment procedure. BE SURE TO DOUBLE CHECK ALL WIRING, ESPECIALLY THE +5 and ±12V POWER CONNECTIONS. Also check for properly inserted ICs. A reversed IC acts like a diode: ZAP!

ADJUSTMENT AND DEBUG PROCEDURE:

1) Power up system. Check voltages if possible (they are hard to reach sometimes.) Connect scope or freq-meter or suitably program the computer to measure frequency. (The scope is very very helpful since this circuit conglomerate is waveshape dependent. Borrow one if you can.)

2) Adjust 555 (IC 9) Record Clock to 8 KHz (125 usec.) (If you have a good ear you can hear the difference between a CPU generated 8KHz tone and the IC 9 generated tone. Listen to both signals with a 2.2Kohm resistor in series with a speaker. IC 9's signal is not square, so if your ear is not good, listen to the 4KHz signal which should be present on IC41 pin 12 (or decoder IC 3 pin 6).)



3) Adjust t1 and t2. Use the computer program listed on page 29 to generate a 170usec square wave. Feed this into the decoder in place of RAW DATA. Connect scope (or speaker circuit) to decoder IC2A, pin 13. The "0" output pulse (700 nS to 1.1 usec) will be seen (heard). (The speaker may have to be attached to the 555 IC4A pin 3.) ADJUST the t1 pot until the "0" pulse just disappears (or the sound stops.)

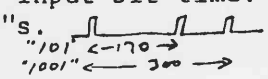
(IF YOU HAVE 4 MHz calculate YOUR OWN!)

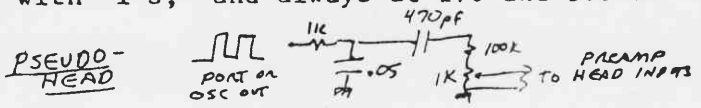
Change the TIME byte in the computer program from 11_h to $2E_h$. The output will now be 300usec. Both the speaker and the scope should be attached to the 555 IC 4B. ADJUST t2 until the output pulse just disappears.

(Note that the 555's remain at +5 until they time out, resulting in negative-going pulses. The 74123 output will be positive pulses. Allow circuit to come to operating temperature before calibration, especially if you used ceramic disc capacitors in the timing circuits. Temperature can cause ±3% drift in discs, versus ±1% in mylar and ±.4% in polystyrene over the same range.)

4) Optional: adjust program to produce 100-160 usec pulses. The DATA line should stay high, indicating "1"s are being decoded. If 180 usec or greater is fed in, the DATA line should go low, indicating "0".

If your scope can see the short DCLK pulses, you will find them correspond to the input bit time: synchronous with "1"s, and always at 170 and 300 usec for "0"s.





HEAD PREAMP ADJUSTMENT:

5) Feed the squarewave output of your CPU port (a la program mentioned above) thru the "pseudo-head circuit" shown on page 28 (and also above) This simple circuit produces pulses at each transition of the squarewave, simulating the signal which comes from a phideck head. (Alternately, you can use the PHI TEST software to start the phidecks loaded with a recorded tape and just use the actual signal coming from the head.)

6) Adjust the signal level on the pseudo head circuit to give you ±2.5 volt (5 v peak to peak) signals at the output of the summing amp. Check to see that the Differentiator is working and producing about ±2volts out. The COMPARATOR should be producing a train of squarewaves if and only if the "select" or "read" signal is low (0v)

7) Turn the 1-turn 500K HYSTERESIS pot to minimum. Adjust the 1-turn SYMMETRY pot until the comparator output is perfectly square. This is best observed by feeding in the smallest possible signal and adjusting for squareness. (A good ear can hear even this on the comparator output using the speaker circuit, but a scope is best. If you do not use the "pseudo-head" circuit, you will have to short the head with a 1 ohm or so resistor --good luck.)

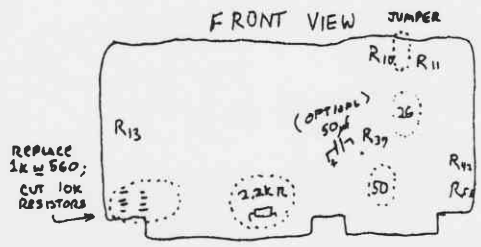
When the SYMMETRY pot is correctly adjusted, turn it a bit lower by a tiny bit (or you can leave it alone.) This adjusts for the slight interaction the HYSTERESIS pot will have on SYMMETRY when it is returned to normal.

Adjust the HYSTERESIS pot to 1/2 to 2/3rds way positive. At about 1/2 rotation, you will get about 8% hysteresis for a strong 2.5volt input signal, and proportionately more as the signal becomes weaker.

8) CMMR (Common mode rejection ratio): Short the two input pins of each preamp together. Attach the Pseudo head signal from the input pins to ground. Both pins are raised high or low at the same time, in COMMON. (An oscillator or other generator can be used.) ADJUST the input level until you can see it being output from the Summing Amp (you should not exceed ±10 volts.) ADJUST the CMMR 20-turn pot until the output of the Summing Amp is minimum. Each head preamp is adjusted in the same way.

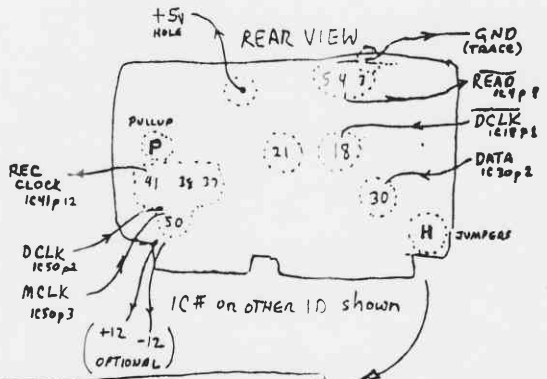
9) DONE! Keep your TV monitor away and tryout phi-test or PHIMON. You might also time your motors (see DG2 or BRIDGE.) And you might want to trace a signal through the preamp-decoder from head to toe, just to see how it works..

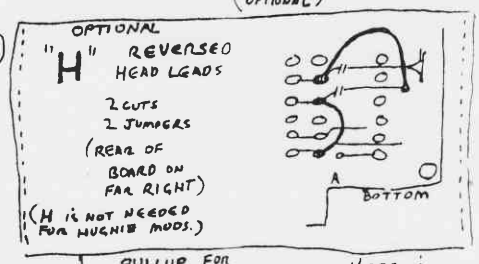
10) If it doesn't work, don't cry. Read on.

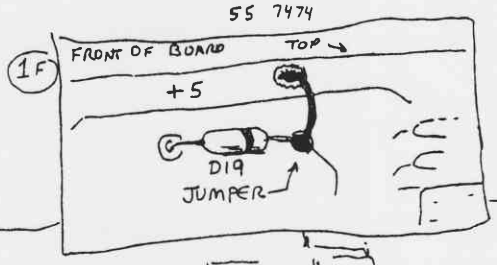


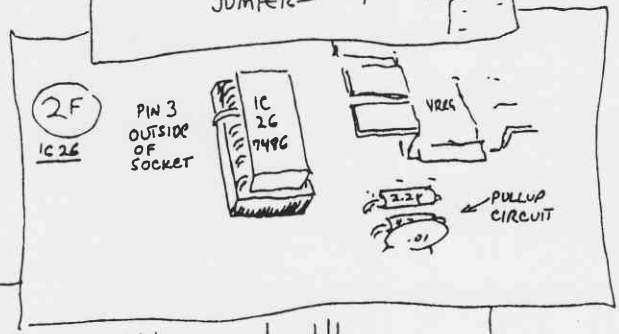
REMOVED:

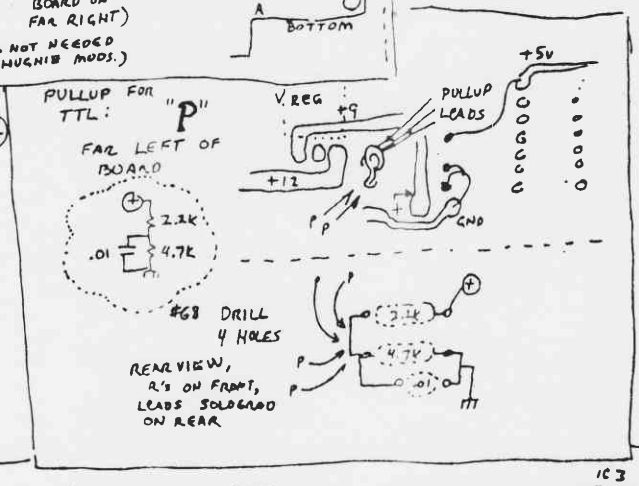
10	7400
11	LM358
13	LM324
37	7493
42	7474
55	7474

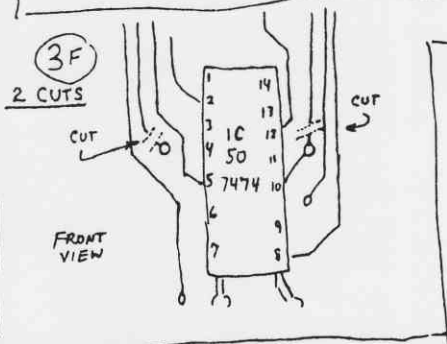


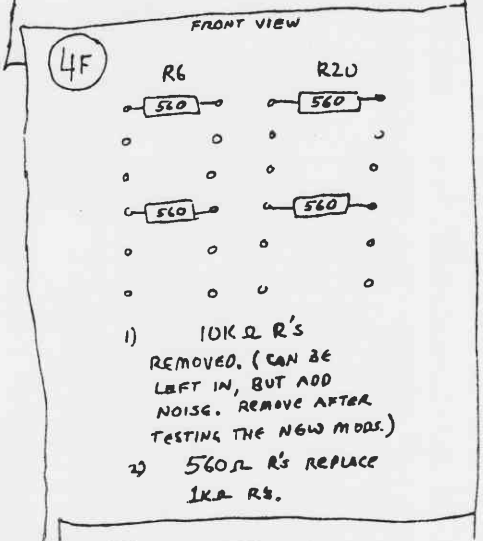






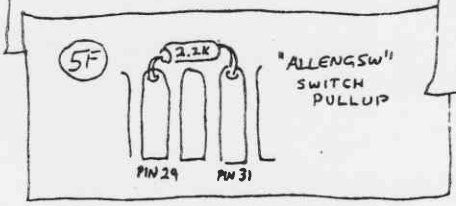


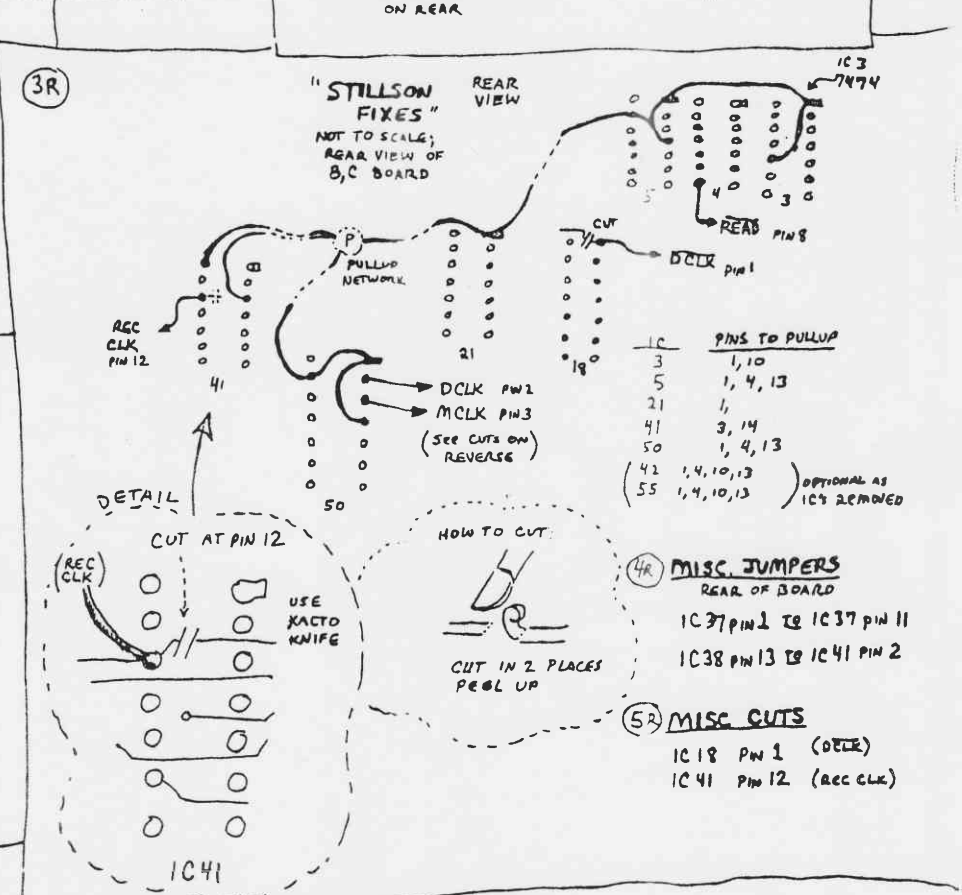




1) 10K Ω R's REMOVED. (CAN BE LEFT IN, BUT ADD NOISE. REMOVE AFTER TESTING THE NEW MODS.)

2) 560Ω R's REPLACE 1KΩ R's.





IC	PINS TO PULLUP
3	1, 10
5	1, 4, 13
21	1,
41	3, 14
50	1, 4, 13
(42	1, 4, 10, 13
55	1, 4, 10, 13)

OPTIONAL AS IC# REMOVED

4R MISC. JUMPERS
REAR OF BOARD

IC 37 PIN 1 TO IC 37 PIN 11

IC 38 PIN 13 TO IC 41 PIN 2

5R MISC CUTS

IC 18 PIN 1 (DCLK)

IC 41 PIN 12 (REC CLK)

phi B, C Mods

Feb. 80

DEBUGGING:

The Hughie-mods undoubtedly will not work for most of you the first time. You will have forgotten a GND connection, or a +, or somesignal, or will have used a surplus IC or put it in backwards. Check it over again. Use a lens to look at the cuts you've made. Don't use surplus ICs on your primary storage device! Do not overlook problems such as bad IC's in the phideck controller, etc., but don't spend too much time looking there when you should be looking at connections (like the poor ones under the mother board, or at the paddle-card -- I had an open in my paddle-card ribbon connection.)

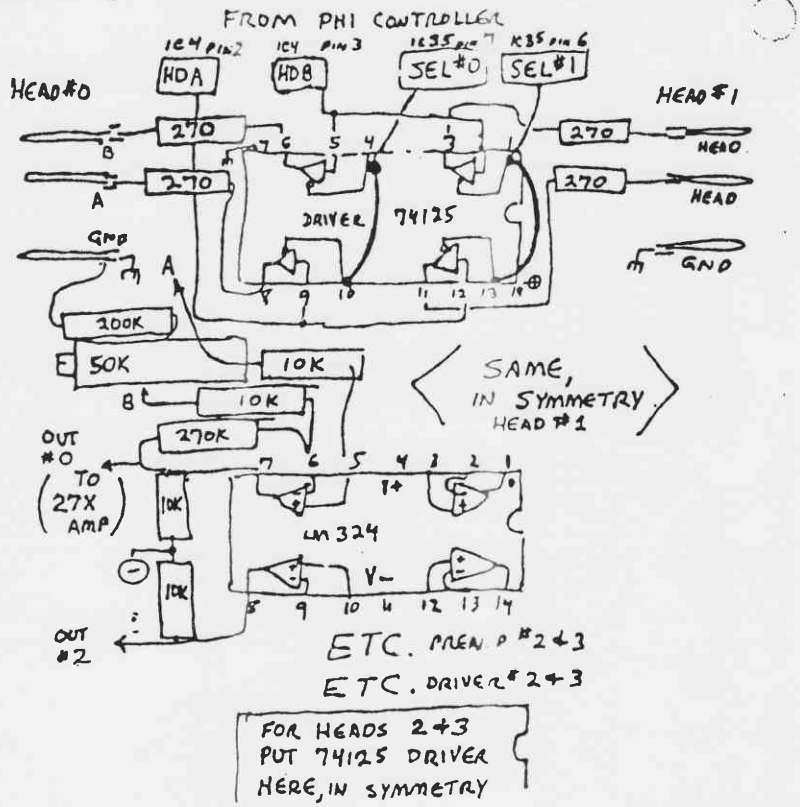
For ease of testing, I have relocated the Phideck test routines supplied with my decks (on suding cassette) and have put in my MAXIMON. However, the unmodified phitest routines can amply test your decks. If you have a very old version of the phi test, check to see that the READ test actually reads more than one byte in each block before giving you a "good". So that I could see a whole C90 test on one screen, I eliminated the "g" letter display. Only the errors "b" will be shown. On the average read of an entire C90 cassette you should get no errors except at the very end of the tape where the leader is attached. Record the test pattern, rewind, and then hit Read first, wait a second, then hit READ TEST DATA. [If you hit READ TEST DATA first, the turn-on-engage transient will usually fool the controller card and give you a few errors.] Let the tape play for about 15 minutes. There should be no b's on the screen. A "b" may appear at the leader.

MISCELLANEOUS:

Connection to Phideck 36pin connector: use pins 8 and 9 for ground, 10 +12V, 11 for Read signal, 12 for REC CLK, 13 for DATA output, 14 nc, 15 -12v.

IN CABINET PREAMP-DRIVER:
suggested layout for driver and preamp together; and LM324. (not tested, just suggested.)

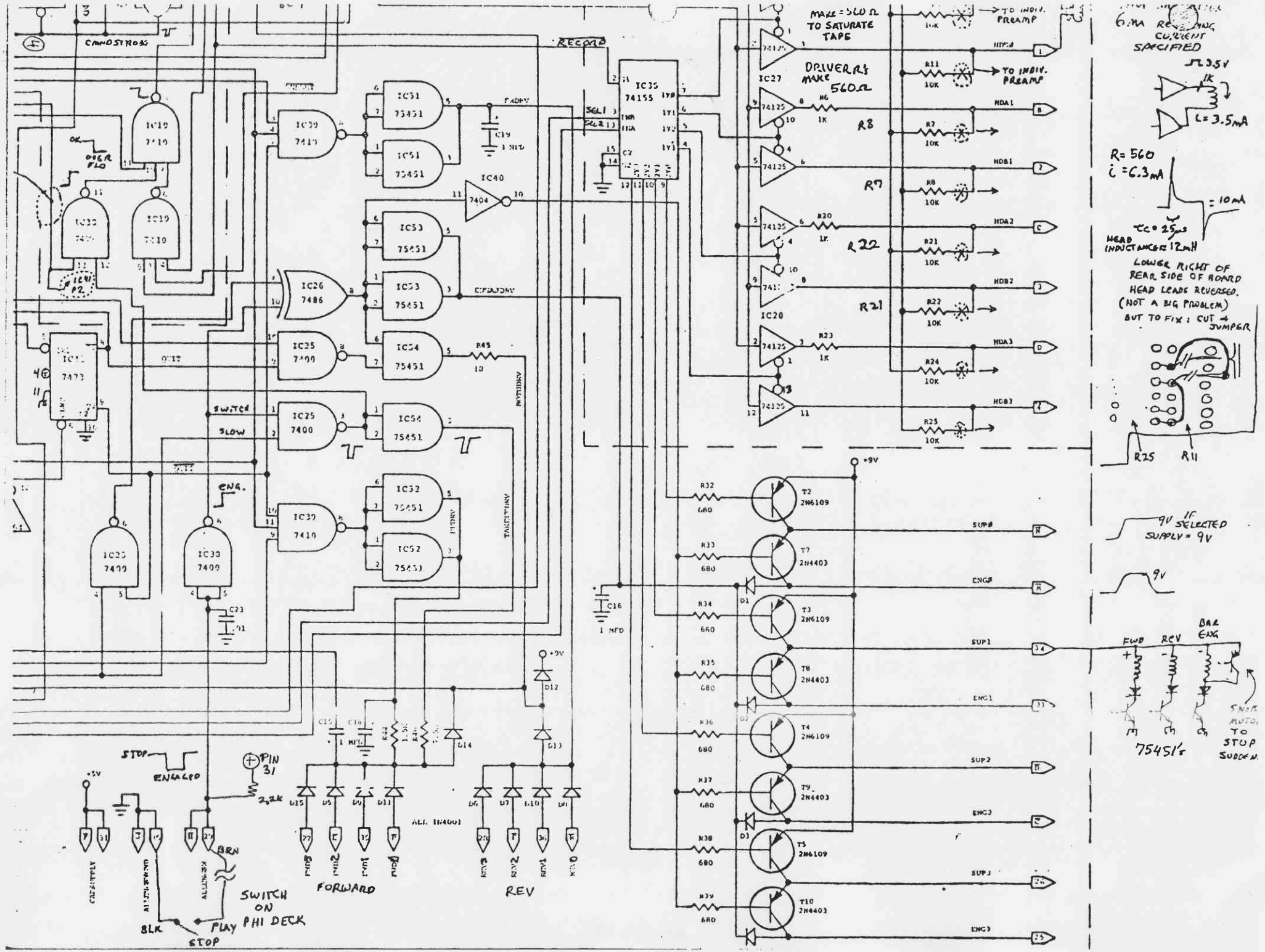
LAYOUT: IN-CABINET PREAMP-DRIVER



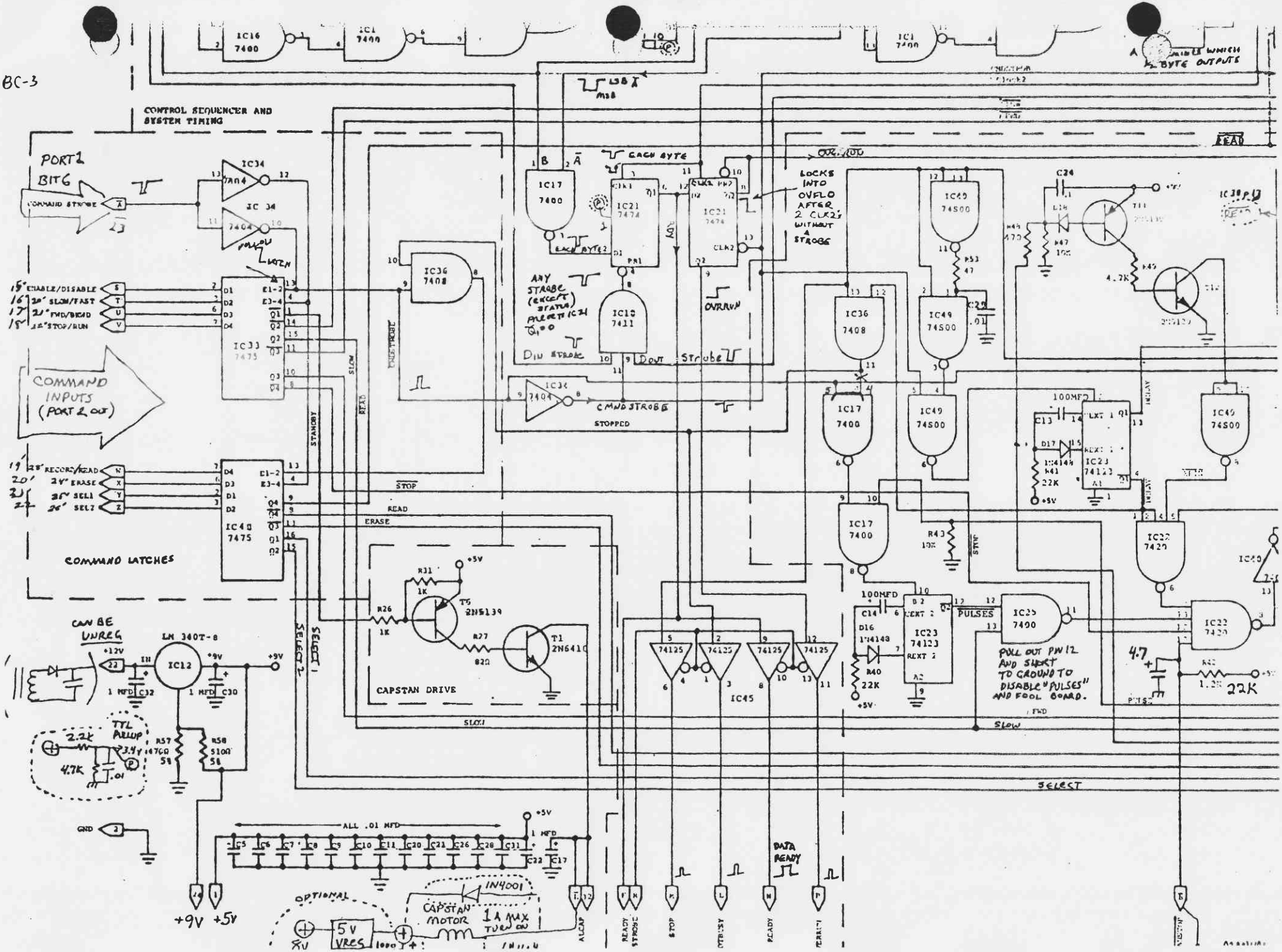
SOFTWARE AIDS:

I have a phideck test program that simulates the head output and measures controller output while checking for accuracy. Each bit of a 256 byte block can be varied in speed, i.e. one bit at -15% speed can be followed by a +15% speed bit. The program sends out bits and inputs bytes all at the same time without requiring a real-time clock: all software timing. Timing resolution 5.2 usec. I may publish this as a TIP, but for now you will have to ask for it.

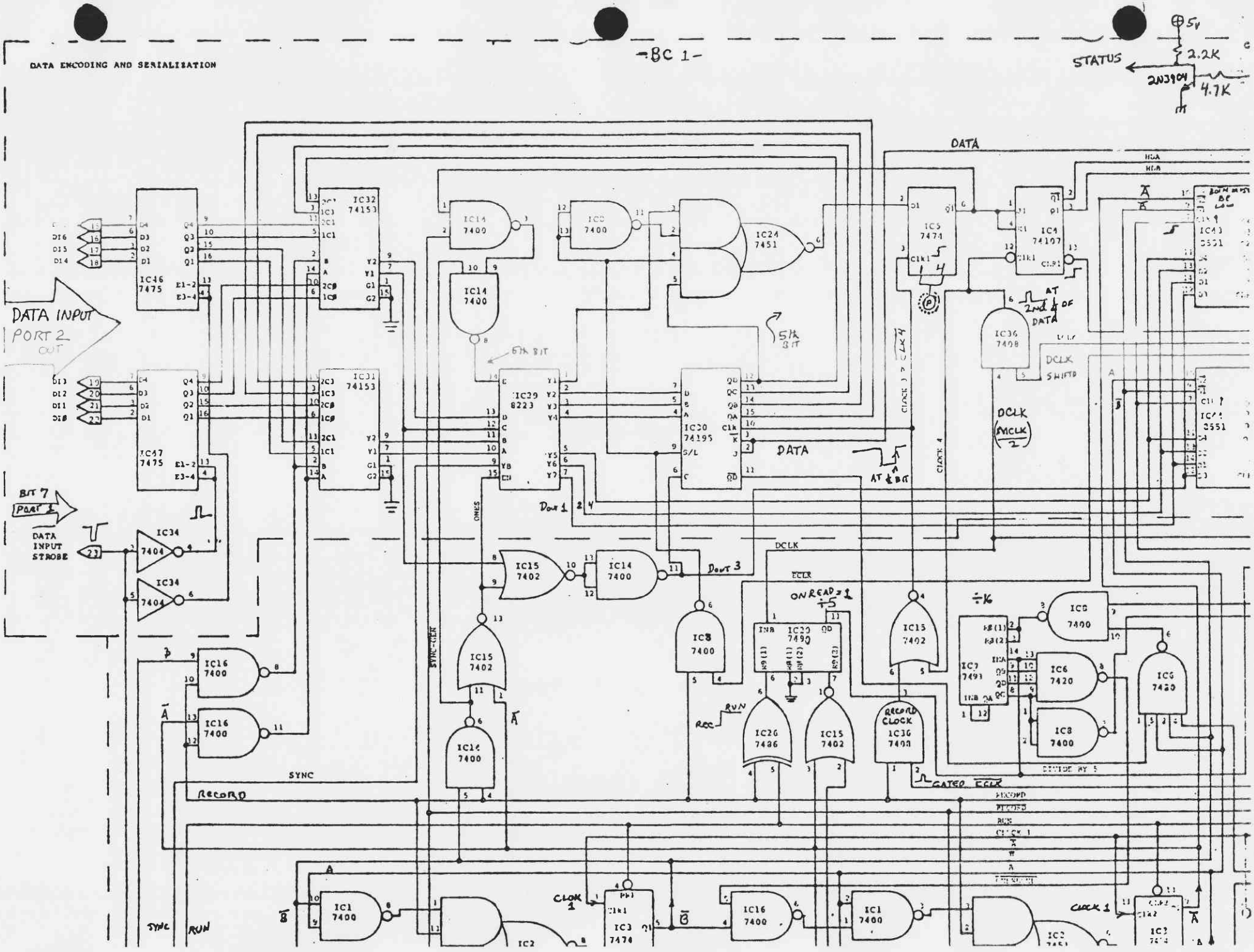
Hughie
March 12, 80

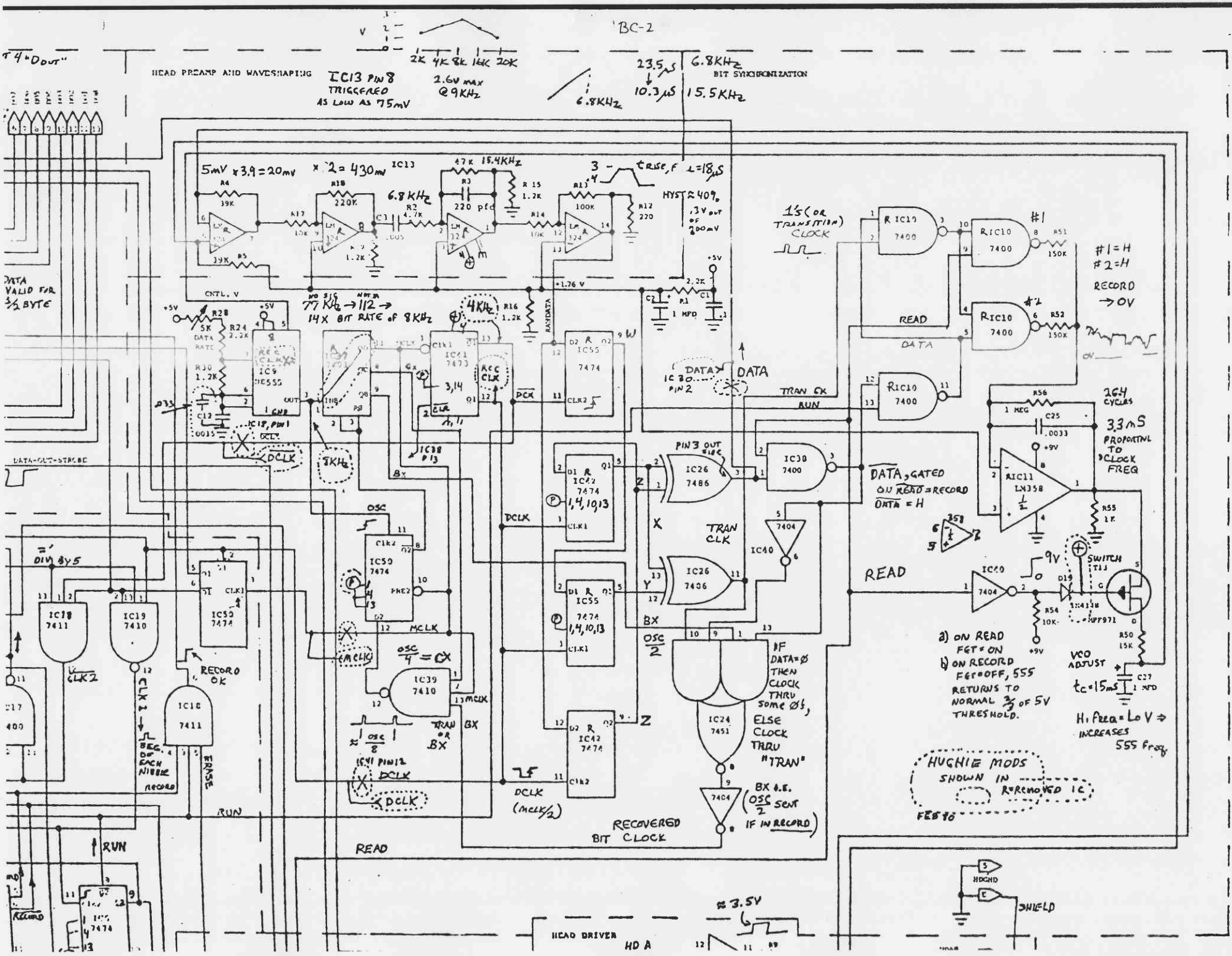


CMNDSTROBE
RECORD
IC35
74155
IC27
IC20
74125
MAKE = 560 Ω TO SATURATE TAPE
DRIVER R's MAKE 560Ω
TO INDIV. PREAMP
R8
R7
R22
R21
HDA1
HDB1
HDA2
HDB2
HDA3
HDB3
IC51
75451
IC53
75451
IC54
75451
IC52
75451
IC39
7410
IC26
7486
IC25
7400
IC40
7404
IC19
7410
IC33
7400
IC41
7473
OK
OVER FLO
QUIT
SWITCH
SLOW
ENG.
FWDRV
ENGAGDRV
BOTHDRV
TAKEUPDRV
C19
1 MFD
C16
1 MFD
C23
+9V
T2
2N6109
T7
2N4403
T3
2N6109
T8
2N4403
T4
2N6109
T9
2N4403
T5
2N6109
T10
2N4403
SUP0
ENG0
SUP1
ENG1
SUP2
ENG2
SUP3
ENG3
D1
D2
D3
D12
D13
D14
D15
D5
D9
D11
D6
D7
D10
D8
ALL 1N4001
FWD3
FWD2
FWD1
FWD0
REV3
REV2
REV1
REV0
FORWARD
REV
STOP
ENGAGED
PIN 31
2.2K
BRN
SWITCH ON PHI DECK
BLK
PLAY
STOP
6 MA RECORDING CURRENT SPECIFIED
3.5V
1K
L = 3.5mA
R = 560
i = 6.3mA
= 10mA
τc = 25μs
HEAD INDUCTANCE = 12mH
LOWER RIGHT OF REAR SIDE OF BOARD HEAD LEADS REVERSED. (NOT A BIG PROBLEM) BUT TO FIX: CUT + JUMPER
R25
R11
9V IF SELECTED SUPPLY = 9V
9V
FWD
REV
BAR ENG
75451's
SNOTE MOTOR TO STOP SUDDEN.


BC-3
CONTROL SEQUENCER AND SYSTEM TIMING
PORT 2 BIT 6
COMMAND STROBE
COMMAND INPUTS (PORT 2 OUT)
COMMAND LATCHES
ENABLE/DISABLE
SLOW/FAST
FWD/BKWD
STOP/RUN
RECORD/READ
ERASE
SEL1
SEL2
CAN BE UNREG
LM 340T-8
TTL PULLUP
GND
ALL .01 MFD
+9V
+5V
OPTIONAL
5V VREG
CAPSTAN MOTOR
1N4001
1A MAX TURN ON
ALCAP
READY STROBE
STOP
READY
DATA READY
CAPSTAN DRIVE
T5 2N5139
T1 2N6410
IC45
74125
IC17 7400
IC21 7474
IC10 7411
IC36 7408
IC49 74S00
IC23 74123
IC25 7490
IC22 7420
IC40 7475
IC33 7475
IC34 7404
IC16 7400
IC1 7400
LOCKS INTO OVFLO AFTER 2 CLK2's WITHOUT A STROBE
ANY STROBE (EXCEPT STATUS) PRESETS IC21 Q1=0
DIN STROBE
DOUT STROBE
CMND STROBE
STOPPED
OVRRUN
EACH BYTE
PULL OUT PIN 12 AND SHORT TO GROUND TO DISABLE "PULSES" AND FOOL BOARD.
PULSES
SELECT
SELECT 1
SELECT 2
READ
ERASE
STOP
STANDBY
SLOW
ENDSTROBE
100MFD
22K
4.7
1.2K
+5V

DATA ENCODING AND SERIALIZATION




BC-2
HEAD PREAMP AND WAVESHAPING
BIT SYNCHRONIZATION
IC13 PIN 8 TRIGGERED AS LOW AS 75mV
2.6V MAX @ 9KHz
23.5µS 6.8KHz
10.3µS 15.5KHz
6.8KHz
5mV x 3.9 = 20mV
x 2 = 430mV
IC13
6.8KHz
15.4KHz
HYST ≈ 40%
1's (OR TRANSITION) CLOCK
#1 = H
#2 = H
RECORD → 0V
READ
DATA
DATA VALID FOR 3/2 BYTE
DATA-OUT-STROBE
77 KHz → 112 → 14X BIT RATE of 8KHz
4KHz
8KHz
DCLK
RAMDATA
TRAN CK
RUN
DATA, GATED ON READ = RECORD DATA = H
26.4 CYCLES
3.3mS PROPORTNL TO CLOCK FREQ
PIN 3 OUT
TRAN CLK
IC26 7486
IC26 7406
IC38 7400
IC40 7404
IC42 7474
IC55 7474
IC50 7474
IC39 7410
IC18 7411
IC19 7410
IC24 7451
IF DATA = Ø THEN CLOCK THRU some Ø's, ELSE CLOCK THRU "TRAN"
BX O.E. (OSC/2 SENT IF IN RECORD)
RECOVERED BIT CLOCK
DCLK (MCLK/2)
OSC/4 = CX
OSC/8
OSC/2
MCLK
RECORD OK
BEG. OF EACH NIBBLE
ERASE
RUN
READ
a) ON READ FET = ON
b) ON RECORD FET = OFF, 555 RETURNS TO NORMAL 2/3 OF 5V THRESHOLD.
VCO ADJUST
tc = 15mS
Hi Freq = Lo V ⇒ INCREASES 555 Freq
HUGHIE MODS SHOWN IN R = REMOVED IC
FEB 80
SHIELD
HEAD DRIVER
HD A
≈ 3.5V