The Kaypro Column

By Alan Barlow

Box 3634
Seattle WA 98124

We've received a number of calls which start out like this: "Help! My new drives are tearing themselves up. Hear them grinding?"

Actually, they aren't grinding. They're just fast-stepping drives running on a slow-stepping machine. If you want to speed up your data accesses and quiet down those new half-height drives, then you're in the right spot.

Drives too noisy? Want faster disk access? Here are two ways to increase the maximum step rate of quad disk drives, from 6 ms to 3 ms, when used with Micro Cornucopia's Pro-8 ROM, version 2 or 3. Both involve minor hardware modifications, including soldering.

The job requires about the same degree of dexterity as the 2 to 4 upgrade or the 5MHz speed-up (both described in issue #21). In other words, it'll be a cinch for some, but out of the question for others. If you're not sure of your abilities, have a technician do it.

Although the fast step mod is independent of the speed-up mod, you might want to tackle the speed-up first, since increasing the CPU clock from 2.5MHz to 4 or 5MHz results in a more dramatic improvement in overall data processing speed than increasing the drive step rate only.

Step Rate

What is "step rate" anyway? And why change it?

The tracks on a floppy disk are in concentric circles, each a measured distance from the center. A stepper motor positions the read/write head(s) accurately at a given track. This motor doesn't rotate continuously, but at a fraction of a revolution for each step pulse received from the disk controller chip on the main board. One step moves the heads radially from one track to the next over the surface of the disk.

For example — the stepper in my Mitsubishi drives rotates exactly 1.8 degrees (1/200 revolution) per step. Since there are 80 tracks per side, the stepper never completes even half a revolution. The quad drives are configured at 96 tracks per inch (tpi); therefore, each step moves the heads 1/96 of an inch. (The direction of movement, inward or outward, depends on whether the "direction" signal is high or low.)

The original Tandon drives were designed to step from one track to the next in 6 milliseconds. The newer quad drives, including my Mitsis, all seem to be capable of stepping twice as fast, requiring only 3 ms per step. Thus every seek operation (stepping the heads from wherever they happen to be to the desired track) should take much less time at the faster rate.

The time won't be reduced by half because a few milliseconds of settling time is required to allow the heads to stabilize over the new track. This time is fixed, regardless of the step rate.

Speed

I ran CRC.COM on a disk loaded with 780K in 95 files (628K if measured in 1K blocks). It took 223 seconds at 6 ms/step and 198 seconds at 3 ms/step (at 5MHz). Time saved — 11 percent.

^KS (save file to disk and continue editing) on a 52K file in WordStar required 74 seconds at 6 ms/step and 68 seconds at 3 ms/step. Time saved — 8 percent.

I performed this test at 2.5MHz as well. The times were 79 seconds at 6 ms/step (the stock Kaypro II), and 74 seconds at 3 ms/step.

Curiously, the 5MHz speed-up and the 3 ms step rate mod deliver exactly the same improvement in time, about 5 seconds. Both mods together result in a reduction of 11 seconds, a 14 percent improvement over the unmodified Kaypro.

Most of this operation (^KS) is writing data to the disk, which isn't improved by either modification (data transfer to the drives is always 250K bits per second, regardless of CPU clock speed or drive step rate). (See Table 1 for a complete comparison.)

Hardware Or Software

If you've installed one or more quad density (96 tpi) drives in your Kaypro, and read the ad for Micro C's Pro-8 monitor ROM, version 2 or 3, you might think you just need to buy the new Pro-8 and plug it in to get the 3 ms step. The ad does say you can select a slow or fast step rate for each drive. But here, fast means as fast as the hardware can go (6 ms), and slow is slower.

Given the 1MHz clock input, the controller produces a step pulse every six milliseconds. Through software, the controller can be programmed to produce step pulses less frequently, but the fastest step rate is still 6 ms.

Why not simply double the disk controller clock speed from 1MHz to 2MHz?

There is a 2MHz clock signal available on the CPU board. However, the controller uses its clock to determine not only the step rate but also the data transfer rate to and from the drives.

The controller must have a 1MHz

(continued next page)

Table 1 - Benchmarks

CRC	6	3	
5 MHZ	2:55	2:38	(75 files/688K (560K in 1K blocks)
	3:43	3:18	(95 files/780K (628K in 1K blocks)
2.5	5:04	4:47	
^KS (Save file to disk and continue editing in WordStar)			
5 MHz	1:14	1:08	(52K file)
2.5	1:19	1:14	(52K file)
$$ea (save file and continue editing in Vedit)			
5 MHz	0.45	0.44	(56K file)
2.5	0.48	0.47	(56K file)
Assembly (submit file — load M80, L80)			
5 MHz	0.23	0.20	
2.5	0.25	0.22	

(continued from page 45)

clock when when reading or writing standard double or quad density 5.25″ disks. It needs a 2MHz clock for reading and writing eight inch double density disks and some high density (1.2 Mbyte) 5.25″ drives (the data transfer rate doubles to 500K bits per second). When it has a 2MHz clock, the controller can generate step pulses at a 3 ms rate. Now if the controller received a 1MHz clock when it was reading and writing data and a 2MHz clock when it was stepping...

I called Micro C and discovered that Dana had already solved the problem by adding a multiplexer to the disk controller circuit.

A multiplexer is an electronic switch that can select one of several inputs and gate that signal to a given terminal. In this case we are selecting either the 1MHz or 2MHz clock signal, depending on the operation in progress, and gating that signal to the clock input of the disk controller chip. Two circuits accomplish this task.

Two Methods, Two Circuits

Through software control, the head load signal can remain off during track seek or restore operations, but a read or write operation always turns it on. The circuit is merely a selector between 1 or 2MHz controlled by the head load output on the 1793 disk controller.

The first method has the advantage of being non-destructive and easily reversible if U87 is socketed.

A multiplexer is piggy backed on U87, and the resulting hybrid chip is plugged back into the U87 socket. Two jumpers run to the 1793 to complete the modification. All modifications can be done from the top of the board, so the main board doesn't even have to be removed.

The second method has the advantage of using spare gates on the board to create the multiplexer, so no extra hardware is needed.

Method #1: Add A Chip

Dana added a 74LS157 multiplexer chip to his CPU board. (This chip actually has four multiplexers on it, but only one is used here.) His circuit is shown in Figure 1.

If you clock yourself down to your favorite five-and-ten-volt store you can pick up a 74LS157 for about forty cents. If U87 is socketed, it's a good idea to pick up a 74LS390 also so you'll have the original chip, just in case you want to reverse the procedure.

On some boards, U87 will have a couple of jumpers on it and a pin removed. If this is the case, prepare the 74LS390 you purchased to match the chip you're removing. Then cut all the pins on the 74LS157 except 8 and 16 at the place where they get wider.

Then solder pins 8 and 16 to pins 8 and 16 on the 74LS390, simultaneously providing power to the 74LS157 and holding it atop the 74LS390. When you reinsert the piggy back chip set, have pin 13 of the 390 bent out slightly so it doesn't go into the socket.

Wiring:

Add a wire from the bent-out pin 13 of the 390 to pin 3 of the 157 (the 1MHz input to the multiplexer).

Add a wire from U82-28 to pin 1 of the 157 (the select control for the multiplexer).

Add a wire from the 390 to pin 2 of the 157 (the 2MHz input to the multiplexer).

Add a wire from U82-24 to pin 4 of the 157 (the output of the multiplexer).

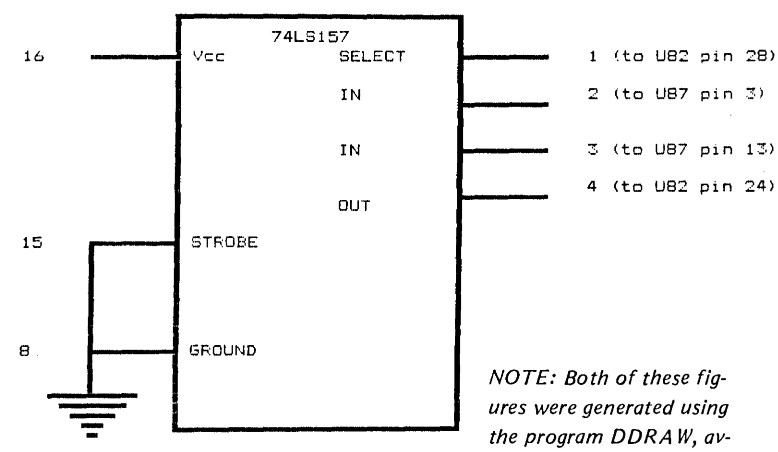
Add a wire from pin 15 of the 157 to pin 8 of the 157 (the multiplexer enable).

This completes the wiring for Method #1. See below for setting the step rates.

Method #2: Use Existing Hardware

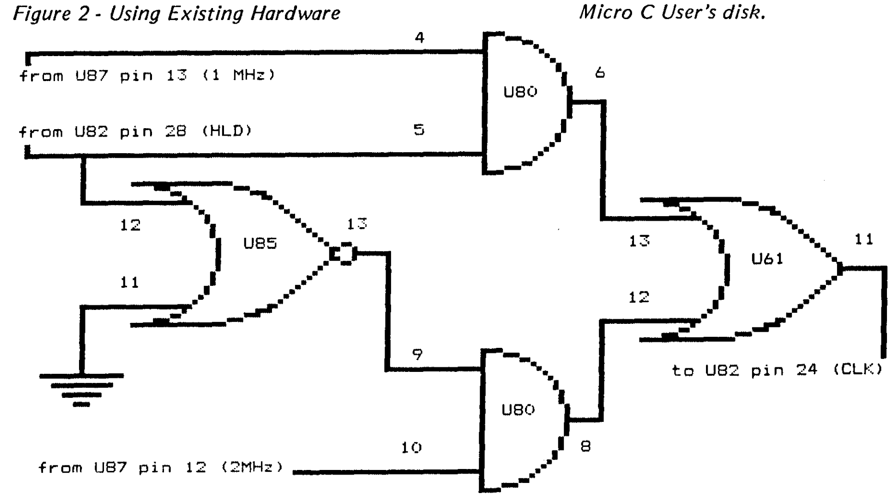
If you don't want to wire in an extra chip, you can easily roll your own

Figure 1 - The Add-a-Chip Method



NOTE: Both of these figures were generated using the program DDRAW, available in the future on a Micro C User's disk.

Figure 2 - Using Existing Hardware



multiplexer, using leftover gates on your Kaypro circuit board. You'll need to cut six traces and then add eight wires. You won't need to bend out any with this method. The trace cutting requires precision and dexterity; read before attempting. See Figure 2.

You might want to correct your Micro C schematic diagram (Kaypro II schematic, 1983, dated 5/23/83).

Look for the U80 gate at coordinates A-7. Pins 1, 2, and 3 are used. Now find the lower of the two U80 gates at coordinates A-3 1/2. Same pins shown, right? (Forget that the first gate is drawn as an OR function and the second as an AND — they are both correct presentations of the same piece of hardware.) The pins of the second gate should be labeled 9, 10, and 8 instead of 1, 2, and 3 respectively. I.e., 9 and 10 are inputs; 8 is the output.

Ready, everybody? It's time for a trace-cutting party! (Hold the beer until after you've finished.) "U87-13" means pin 13 of U87. (Refer to Micro C issue 21, The Kaypro Column, for hints on pin counting.)

After each trace is cut, use an ohmmeter to verify that you've indeed broken the continuity. Traces will be cut on the foil (bottom) side except the first two below:

Cut trace at U80-10. This is the only tricky item. The trace is on the component side, covered by part of the socket. Remove U80 from its socket. With your smallest needle-nosed pliers, break out part of the bridge between the two rows of pins, near the right end of the socket. (The plastic is soft and breaks without difficulty.) Locate two parallel traces near the right end of the socket. Cut the one to pin 10, nearest the center of the socket. Replace U80 in its socket.

Cut trace from U87-13, on the component side. It runs next to the letter "R" of "R34", between U84 and U87.

Cut trace at U61 between pins 12 and 13.

Cut trace at U61 between pins 13 and 14.

Cut trace at U85 between pins 11 and 12. (U85-11 will remain grounded.)

Cut trace at U80-5.

That's all the cutting. Now add these wires (Figure 2):

Add wire from U87-13 to U80-4.
Add wire from U82-28 to U80-5.
Add wire from U82-28 to U85-12.
Add wire from U85-13 to U80-9.
Add wire from U87-12 to U80-10.
Add wire from U80-6 to U61-13.
Add wire from U80-8 to U61-12.
Add wire from U61-11 to U82-24.

NOTE: U85-11 to ground already exists. Leave it.

This completes the wiring for Method #2.

Whichever method you followed, check your work carefully. Use an ohmmeter to check for short circuits between adjacent terminals where you soldered connections. (If you have any doubt at all about your ability, and U87 is socketed, purchase two chips and use the first method. You can always get back to square one by plugging in the original chip.) Note: If you screw up the clock to your floppy controller, it's awfully hard to boot up.

Setting Drive Step Rates

Run PRO82SET.COM (it came on the disk with the Pro-8 ROM, version 2) or CONFIG83.COM (for version 3). All step rate values shown in the menu will now be cut in half. Select the "Use Slower Step Rate on Selected Drives" option. When the next menu appears, if you select item #5, "No Slow Drives; Exit this Function," you are setting all drives to the default value, which is now 3 ms/step.

Now return to the first menu and write the changes to the disk in drive A:. Finally, you must perform a cold boot (RESET) in order to re-read the system tracks and enter the new step rates into RAM. You can use SYSGEN to copy the modified step rate to other boot disks.

I discovered that my one remaining Tandon drive seems to step reliably at 3 ms. (Nevermind that it sounds like a rusty cement mixer gargling pea gravel.) Not all Tandon drives can handle this. If you cannot log onto your Tandons or you start getting errors, use a slower step rate. The second fastest step rate (it says 12 ms, but it's really 6 ms after the modification) will do. There may also be a danger with drives that have a loose band drum. They can knock themselves out of alignment with the faster step rate.

Most quad density disk drives are designed to step at 3 ms. The TEAC and Mitsubishi quad drives which buzz at the 6 ms step rate will become nearly silent at 3 ms. Some Shugart 465s were designed for 3 ms and others for 6 ms. If your Shugart quad is quiet at the 6 ms rate, chances are you have the 6 ms version of the drive.

■ ■ ■