LAB NOTES

COMPUTER MUSIC, WITHOUT THE COMPUTER. or: What to do 'til your processor arrives.

By: John S. Simonton, Jr.

I realize that a lot of you will respond to the introduction of the 8780 Equally Tempered D/A with a frustrated, "But, I don't HAVE a computer."

Here's a little surprise. You don't really need a computer to do some very interesting and useful things with the 8780. You are going to need some additional hardware, as we'll see in a moment, but it's not only inexpensive, it's also equipment that you'll need for processor interfacing later on anyway. You're not building something that will be scrapped when your computer arrives, just getting a head start. Getting READY:_, so to speak.

Let's shift our mental gears for a minute, and instead of thinking of the 8780 as a computer peripheral, we'll consider it in terms of being a digital sample and hold.

Our analog S/H circuits are acceptable, but they will always drift because they store information by charging a capacitor. Even if we were able to miraculously devise a capacitor with no leakage, we still have to measure the charge on the capacitor; and whatever circuit we use to do that will itself eventually drain away all the charge (I think that a Mr. Heisenberg had something to say about this, but I'm not certain). With a digital S/H, we don't have that problem, because we're storing the information as a pattern of 1's and 0's.

To use our new digital S/H we need some way to provide it with the 1's and 0's it needs to decide what voltage to produce. We need some way to "encode" our AGO keyboards.

There are lots of ways to do this, including the simple expedient shown in figure 1.

This is frequently referred to as a "brute force" encoder. When a switch closes, any diode connected to the switch line forward biases, causing a 1 to appear on the data line connected to it. The diodes are there in the first place to prevent "sneak" current paths back through the matrix. This is an acceptable encoder as long as you assume that only one key is going to be down at a time. But, when two keys are pressed

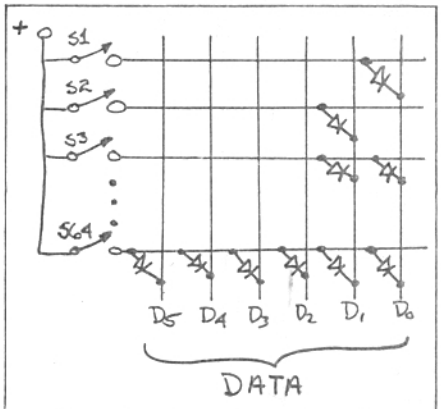


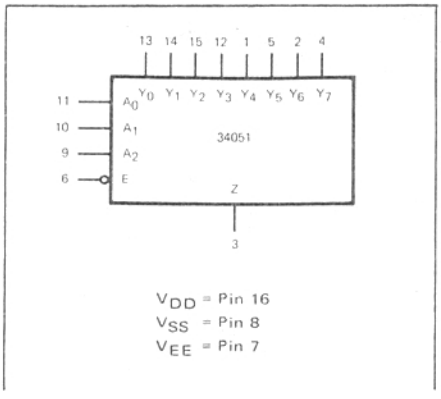
Figure 1 - Brute Force Encoder

simultaneously, the diodes act like OR gates and the data that comes out may or may not (most probably not) represent those keys. If, for example, we were to press the first two keys down at the same time, data lines D_0 and D_1 would both go high. Exactly the same situation that we had defined in figure 1 as being an indication that key 3 was down:

BUMMER

A more popular approach (because it works better) is to "scan" the keyboard a switch at a time to see if any are closed. There are LSI chips that do this with a single integrated circuit package; but, while saving design time is a great temptation, we're not going to use them. They're too expensive, and worse yet, not versatile enough to do all the things that I have in mind.

So that you can follow the design that I prefer, let me turn you on to a new part:



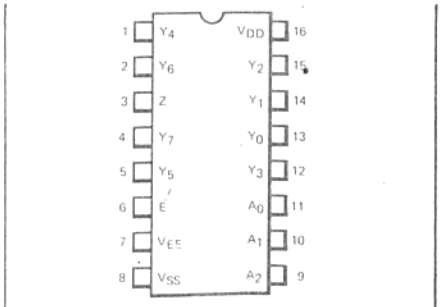


Figure 2

This is called a "4051 8 channel analog multiplexer/demultiplexer". Or, just 4051. Inside the package are 8 bilateral CMOS switches. While one side of each of these switches is tied to one of the pins Y_0 - Y_7, the other side of all the switches are commoned and connect to pin Z. In mechanical terms, it looks like this:

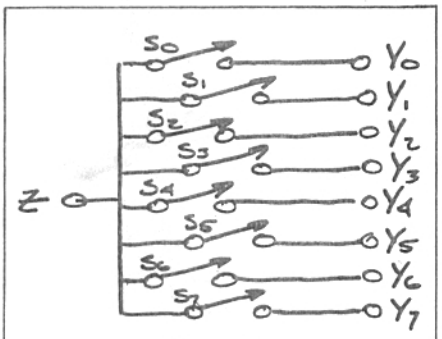


Figure 3 -
Mechanical Equivalent of 4051

One of the neater things about the 4051 is that each of those switches is individually "addressable" from the pins marked A0 - A2. If I put the binary number 000 into the address pins, switch S_0 will "close". 001 causes switch S_1 to be activated, and so on to 111 which addresses S_7.

You will also notice a pin labeled E. This is an enable pin that sort of says "GO" to the rest of the circuitry in the package. As long as this pin is held at a high voltage, all of the switches will be "off", but when the E pin is grounded, the switch specified by the address currently on the A pins will close.