DRAWING BOARD

Here's the keyboard section of the all-electronic audio router.

ROBERT GROSSBLATT

If you've been reading my articles for any length of time, you should realize that every one of my designs goes through exactly the same sequence of stages. I put a lot of thought into the diagrams I drew last time, and a lot of bench time goes into translating those diagrams into working circuits. Although nothing is written—or wired—in stone, the first working circuit is usually pretty close to the original drawings. This is how to produce prototypes.

As you go through the process of creating the prototype, a lot of new ideas will occur to you about how the design can be made better, faster, smaller, and so on. But these new ideas should not be acted upon until you have a working circuit on your bench.

A perfect example of what I'm talking about is the project I'm working on now—the all-electronic audio router. Although I haven't made one exactly like it before, its basic elements are things I've done many times before. The first part of the design is the keyboard. Its details are shown in Fig. 1. There are two basic parts to the circuit: the keyboard encoder made from a CD4017B CMOS counter/divider and the clock made from a pair of CD4011B NAND gates.

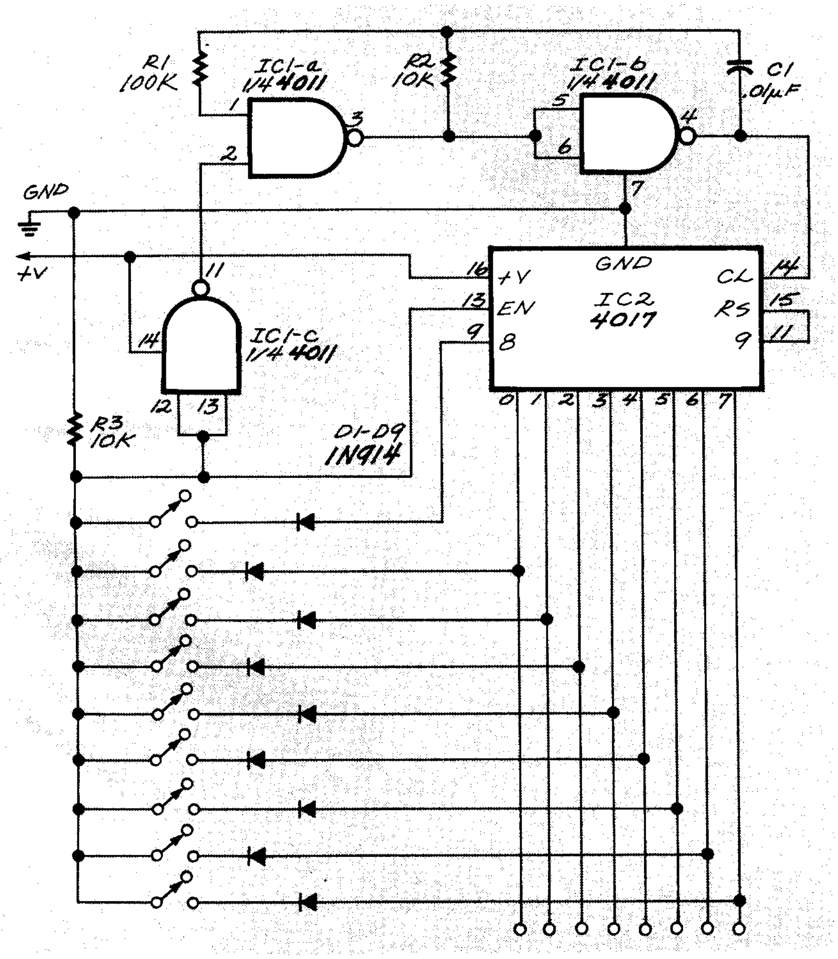

FIG. 1—THE FIRST PART of the all-electronic audio router design is the keyboard. It consists of an encoder made from a CD4017B CMOS counter/divider and a clock made from a pair of CD4011B NAND gates.

The clock circuit is a gated oscillator whose component values were chosen to produce a 10-kHz square wave. The clock runs when pin 2 of IC1-a is logic high, and it stops when pin 2 is logic low. When the clock is running, the outputs of the 4017 keyboard encoder will go high sequentially.

The reset control (IC2 pin 15) is connected to pin 11, the last output of the 4017. When that output is selected, the 4017 resets itself and starts over again with the first output. The reset pin could have been grounded permanently, but I connected it to pin 11 of the 4017 to remind me that there are times when the 4017 should not count through a full sequence.

The final circuit will be able to select from eight inputs, so I need eight 4017 outputs—one for each selected input. As shown in the schematic, however, I'm using nine of the outputs. The reason for this will become clear as I go through the keyboard operation.

The nine outputs of the 4017 are connected to the keyboard switches through diodes D1 to D9. Those diodes prevent a logic high on one line from feeding back to the low lines. Although only one keyboard switch is supposed to be pressed at a time, I must guard against mistakes. Just as when typing on a keyboard, sometimes the wrong key is pressed and sometimes more than one key can be pressed at a time. The diodes are an inexpensive way to ensure that fat fingers only cause mistakes, and not smoked silicon.

The logic high that results from pressing a key is fed to the 4017's players all contribute to the cumulative total noise exposure.

Q. Is there any way of knowing when sound is loud enough to cause injury to people?

Aside from making objective measurements with instruments designed to measure sound intensity, there are some significant subjective physiological effects. If anyone leaves a sound-intensive environment with ringing or a feeling of stuffiness in one or both ears, then his/her ears have suffered some degree of overstress. Repeated exposure could also cause low levels of damage without any obvious immediate symptoms. The stuffiness is a common physiological reaction to sustained high sound levels and is accompanied by a temporary threshold shift (TTS)—in other words, a hearing loss. One study reported in the "Canadian Family Physician" found a TTS of 10 dB among 76% of its test participants attending a rock concert. The hearing loss was still present 40 to 60 minutes later. A 10-dB loss approximates a 50% reduction in the loudness of sounds. Although the loss was temporary, it contributes to a possible permanent threshold shift.

Q. How do decibels relate to sound level?

Decibel is a technical term used in many areas of electronics and in acoustics. Sound intensity, which is the topic that concerns us here, is usually measured in dB. The human ear's response to sound level is roughly logarithmic (based on powers of 10), and the decibel scale reflects that fact. An increase in level of 3 dB doubles the sound intensity, but an increase of about 10 dB is required before the sound is perceived to be twice as loud. Therefore, a small increase in decibels represents a great increase in intensity. For example, while 10 dB is 10 times more intense than one dB, 20 dB is 100 times more intense (10 × 10, rather than 10 + 10), 30 dB is 1000 times more intense (10 × 10 × 10), and so on. The sound intensity multiplies by 10 with every 10-dB increase. The reason for this scale is simply that the human ear is sensitive over such a wide range of acoustic energy that the numbers involved had to be compressed for convenience. The dB scale converts a range of one million into 120 dB.

TABLE 2—DECIBEL LEVELS OF SOME COMMON SOUNDS

Sound Pressure (dB)	Sound Source
130	Jack Hammer (at 5 feet)
120	Discomfort/Pain Threshold Loud Rock or Disco Music
110	Riveter Heavy Truck (at 50 feet)
100	Very Loud Classical Music
90	Heavy Street Traffic (at 5 feet)
80	Loud Singing Heavy Traffic (at 40 feet)
70	Fairly Loud Speech (at 3 feet) Department Store/Noisy Office
60	Background Music Average Office Quiet Residential Street
50	Light Traffic Average Residence
40	Very Low Radio at Home Very Soft Music
30	Country House Quiet Auditorium
20	Faint Whisper (at 5 feet)
10	Leaves Rustling Anechoic Chamber
0	Threshold of Audibility

Note: Sound-pressure level varies with distance from source.

To provide an approximate reference, the approximate or average levels of common sounds are shown in Fig. 2.

Q. Fig. 2 shows that classical music can become quite loud also. Why isn't it as dangerous to human hearing as rock music or disco sound?

If you were to compare rock and classical music with test instruments such as a sound-pressure meter and a real-time spectrum analyzer, you would find that rock music and disco sound typically have a continuously high level and more energy at both high and low frequencies. Classical orchestral music can have its loud moments, but they are seldom sustained for any length of time. Human ears have a chance to rest and recover. There is also less high- and low-frequency energy. Exceptions to this generalization are the works of some modern classical composers who work with electronic synthesizers that are as capable of blasting your ears as the best of the heavy-metal freaks.

Q. Why do DJs tend to play music so loudly?

Mostly in response to their public's demand. The louder the music (particularly the bass line), the greater the visceral impact and the sensory saturation. As a background for those who have not been at school dances, the sound levels heard and the impact felt are exaggerated versions of the kind of sound occasionally heard from high-volume car stereos touring the streets of our cities. Also, DJs (and rock musicians) tend to play even louder than needed to attain the desired effects because they have already been somewhat deafened and have to operate at high levels for the music to sound as loud to

Continued on page 90

enable input pin 13 and through inverter IC1-c to the gate control of the oscillator. As long as a keyboard switch is held down, the selected 4017 output remains high. When the key is released, the oscillator starts, and the 4017 outputs start sequencing high again.

I also need a way to clear the input if I make a mistake. That's why I'm using nine of the 4017 outputs. Note that the ninth output doesn't go anywhere. When this key is pressed, the other outputs are all logic low so I can use this to clear the input selection. Also note that the common side of the switches is held low by resistor R3. Without that, the input of IC1-d would float.

There's nothing critical about the keyboard section of the circuit, but if it doesn't work properly, the rest of the circuit isn't going to work either. Build the keyboard first, and make sure it works correctly before continuing with the project. If you are having trouble getting the keyboard to work, replace C1 with a 10µF capacitor to slow the clock down. Then, with a multimeter, logic probe, or even a simple LED and current-limiting resistor, check to see if the 4017 counter/divider is operating correctly.

When the keyboard is working properly, proceed with the circuitry that stores the column information for the router. Because the keyboard line remains high only as long as a key is pressed, I need some way to store that logic state. I have eight possible inputs (remember that the ninth output of the 4017 is for the clear line), so I need to design a circuit that can store an eight-bit word.

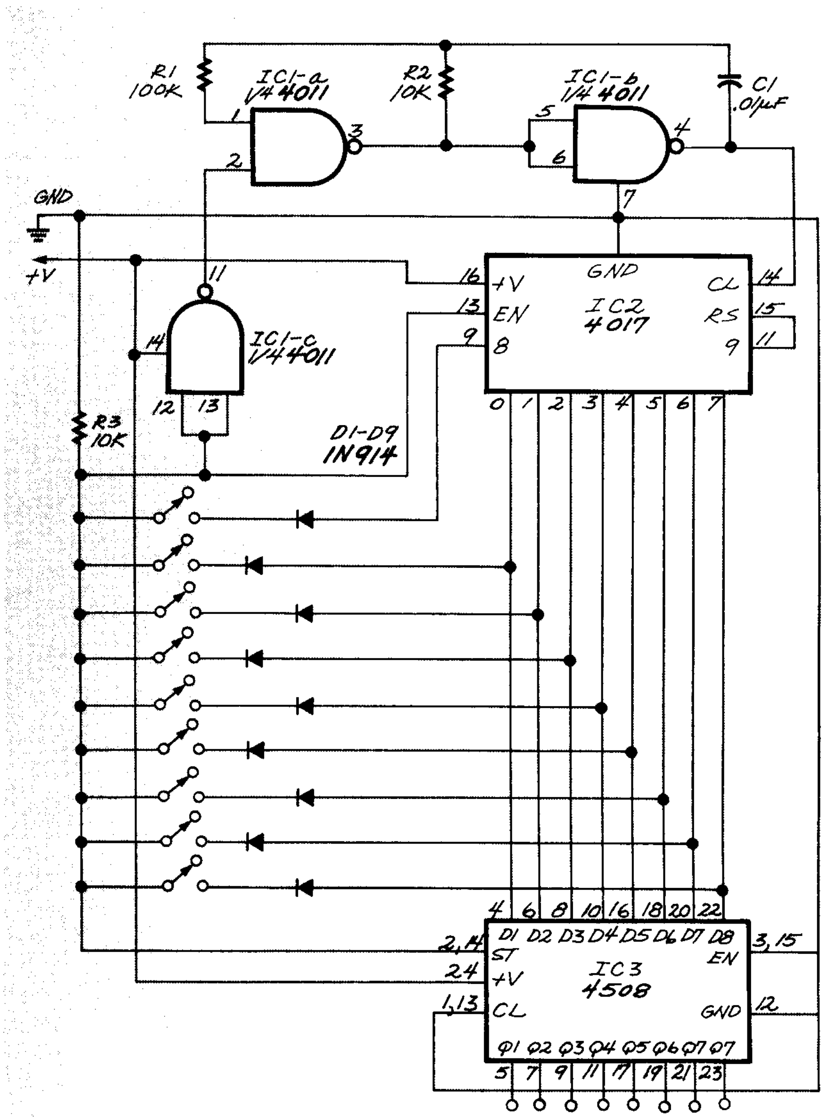

FIG. 3—THE OUTPUT OF THE CD4017B feeds the inputs of the CD4508B. Every time a key is pressed, the state of the eight 4017 output lines will be latched in by the 4508.

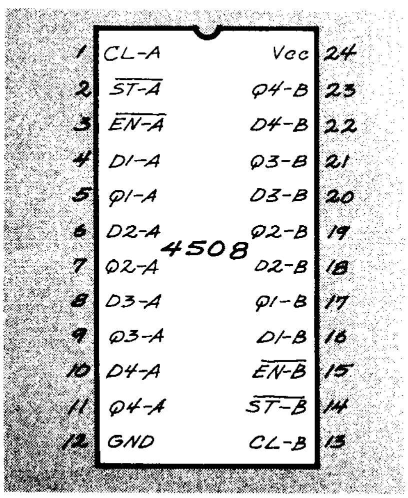

FIG. 2—AN EIGHT-BIT LATCH can be formed by tying together the two control pins of a CD4508B dual four-bit latch.

The simplest solution is with a 4508 dual 4-bit latch whose pinout diagram is shown in Fig. 2. By tying the control pins of each half of the latch together, I can use it as a single eight-bit latch.

The STORE lines (pins 2 and 4) control the IC's inputs; when they are high, the input data is transferred to the internal latches, and when they're low, the data is ignored.

When brought high, the CLEAR lines (pins 1 and 13) will store zeros in the latch. When the ENABLE lines (pins 3 and 15) are high, the latch outputs will be three-stated. The three-stated, or high-impedance output capability is useful at times but I won't need it in this circuit because it doesn't use a common data bus. As a result, there's no possibility of bus contention and all of the outputs can be left permanently enabled.

Figure 3 shows the 4508 added to the circuit. The 4017 outputs feed the inputs of the 4508. Because the 4508 latches data on the rising edges of of the signals at its STORE inputs, I have connected those lines to the "any key pressed" common side of the switches. Every time I press a key, the state of the eight 4017 output lines will be latched in by the 4508 and transferred to its output pins. Ω

DRAWING BOARD

This new circuitry for the audio router will assign a switch selection to one of four outputs.

Every year a new crop of electronic wonders shows up on the local store shelves, where they are bought by a public that is hungry for new features. If new features and technology are not added to last year's products, fewer people will buy them. In truth, however, most of the so-called "new technology" each year isn't really new at all.

The same can be said of my project designs. Complex projects are really just combinations of simple projects. All the elements of the audio router I'm putting together are things that I've done before. Those building blocks are proven circuits that have worked well for me many times over the years, and using them will help ensure that this new project will work as I expect it to.

The router

With that out of the way, I can get back to the audio router. The circuitry I've laid out so far "saves" pressing an input switch on the keyboard in a latch. Now I need a circuit that will let me assign that switch selection to one of four outputs. Because I want to have four outputs, I need four latches. As with the inputs, I've chosen to use CD4508B CMOS dual 4-bit latches.

The layout for the output latches is shown on the right side of Fig. 1. The inputs to the latches form an eight-bit bus (one bit for each input). The latch outputs will be the control signals for the analog switches that will route the audio signals. As with the input latch, I don't need to control the enabling of the outputs because there's no possibility of bus contention at the output. Therefore I've grounded all of the 4508's enable (EN) inputs (pins 3 and 15 of each chip).

The clear (CL) inputs (pins 1 and 13) are also tied together so I have some way to clear all the output latches at once. In a similar manner, the store (ST) inputs (pins 2 and 14) of each chip are tied together and the resulting four lines are brought out to the four output selection switches previously mentioned.

Once you get this output section wired up, you can connect it to the input section you've already built. Five new switches have been added: four output select switches and a clear switch that clears both the output latches and the input latch as well.

Another function that was added to the circuit is a way to guarantee the state of all the latches when power is first applied. The 4508 is cleared when a high is put on the clear inputs, so a positive-going pulse must be generated at power up. Since there was one NAND gate left over in the 4011, I used it to build a half monostable multivibrator consisting of IC1-d, R5, and C2. When power is applied to the circuit, the brief positive-going pulse that appears at the output of IC1-d travels through R4 and clears all five of the 4508 latches in the circuit. Once that's done, the output of IC1-d stays logic low and prevents the clear inputs from floating.

You could use that same pulse to clear IC2, the heart of the keyboard section. When power is first applied to a 4017, there's no guarantee as to the state of its outputs. If you disable the keyboard clock, turn on power, and check the output pins of IC2, you'll see that the 4017 might have more than one output logic high—an illegal and somewhat confusing state. But that's corrected as soon as the chip gets a reset pulse. The reason I don't specifically reset

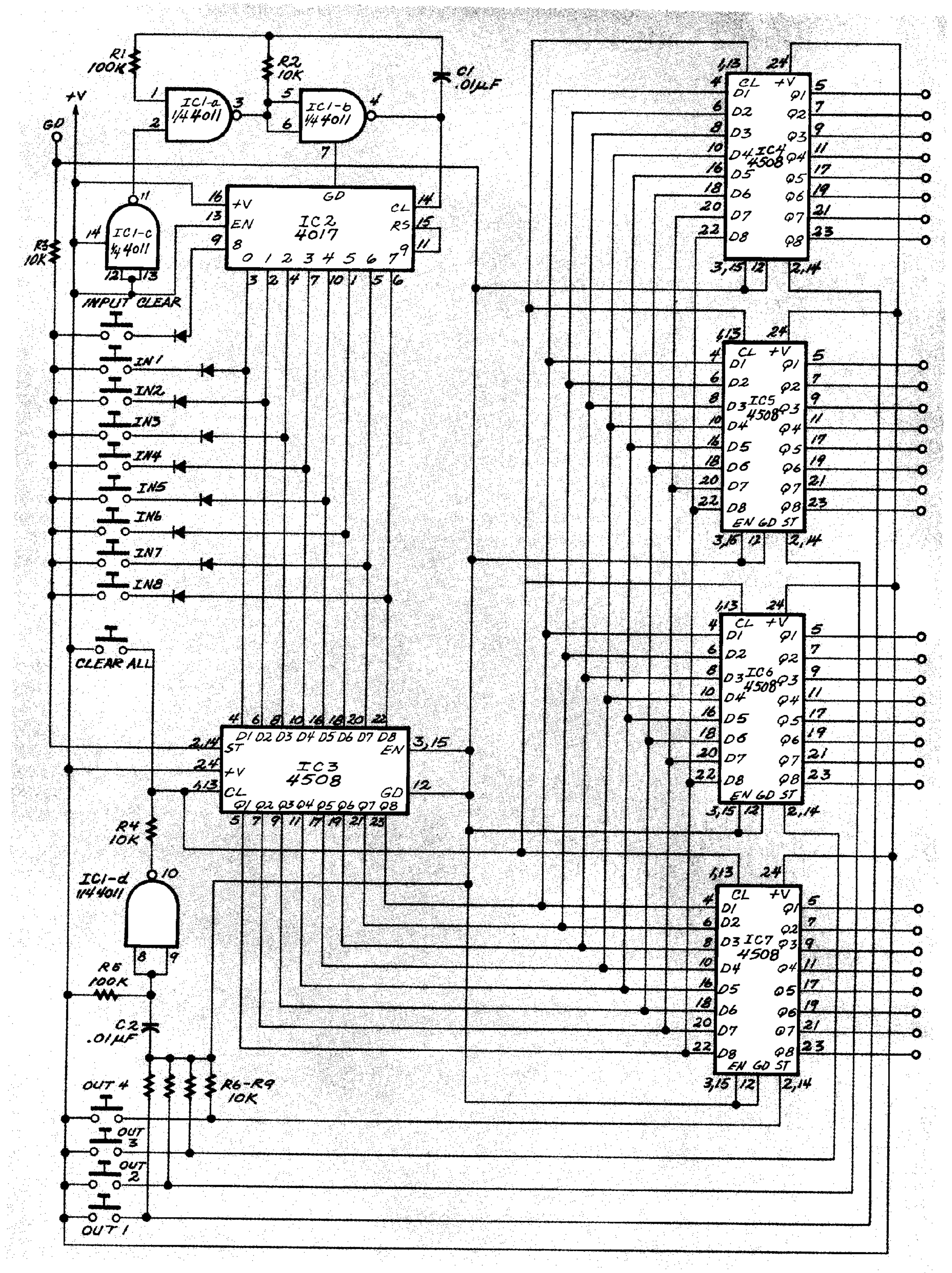

FIG. 2—THE COMPLETE CIRCUIT so far. Four output select switches and a clear switch that clears both the output latches and the input latch have been added.

the 4017 at power up is that it will be done automatically after a maximum of nine clock pulses. Since that takes only a couple of milliseconds, I'd rather just wait for it to reset than have to add an additional OR gate to do the job.

The four output selection switches connect one of the output latch's store inputs to +V, which is a cheap and easy way to generate the positive pulse needed to store the latch contents. I did not debounce the switches because it doesn't matter if the latches receive a short train of pulses. The state of the inputs doesn't change, so the number of pulses sent to the output latch is irrelevant. Resistors R6 to R9 are connected between the store inputs and ground to prevent those inputs from floating.

After it's wired up, test the operation of the circuit by selecting an input and an output and then, using a logic probe, check the output latches. Make sure that the expected output is logic high and that the other outputs remain low. Check both clear switches as well. If you get consistent errors, you've probably miswired the output bus. If you get random errors, you probably have left one of the control inputs floating.

The circuit now generates the signals needed to control the ICs that will be doing the actual audio switching. When we get together next time, I'll add this final section to the circuit and the project will be complete. I'll talk about a few extra features that can be added to the switcher to make it more useful, and I'll supply a PC board layout. Ω

"Well doctor, we first became concerned when he refused to play Nintendo."