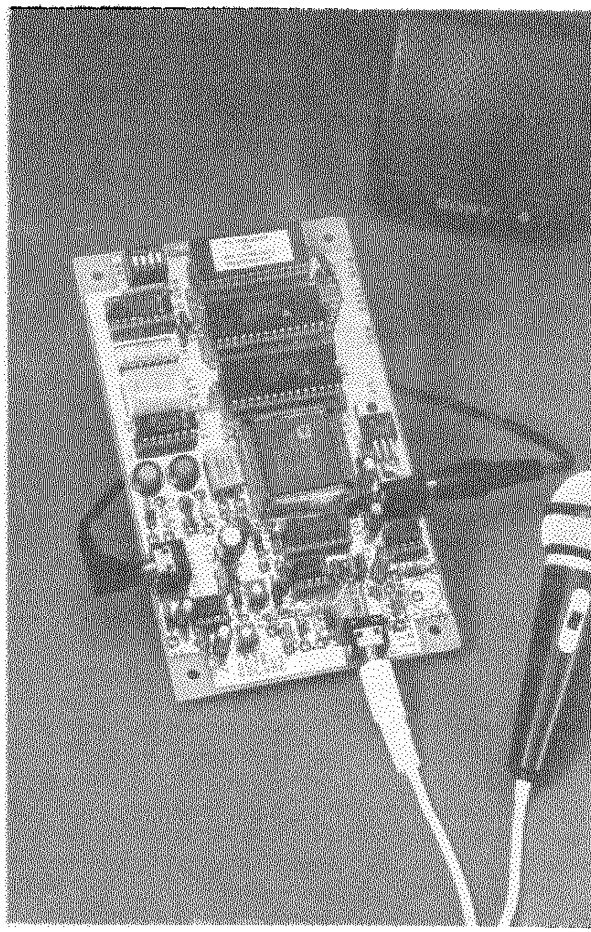
VFX DIGITAL SIGNAL PROCESSOR

WOULD YOU LIKE TO CHANGE THE pitch of your voice or create such special audio effects as echo or reverb? We will show you how to build a voice effects processor (or VFX processor, for short) that can generate such unusual effects. We'll explain the basic algorithms used to perform these DSP (digital signal processing) techniques, and examine the heart of the VFX hardware, the Analog Devices-2105 digital signal processing microcomputer. If you're on a tight budget, you'll appreciate that this project costs much less than any commercial single-effect generator.

Before we describe the details of the hardware and software, let's look at what the VFX processor does. The VFX processor accepts audio signals, digitally processes the information in one of three user-selected modes, and amplifies the signal for listening with a speaker or a pair of headphones. All you need besides the VFX processor is a microphone, a pair of headphones, and a 9-volt DC power source—all of which are available from the source given in the Parts List.



A little DSP goes a long way in generating unusual sound effects.

CRAIG BORAX and DAVID BECK

Basic operation

The VFX block diagram is shown in Fig. 1. A four-position DIP switch (of which only three are used) puts the VFX into one of four operating modes: harmonizer, echo, reverb, and test. Table 1 shows the DIP switch positions for each mode. The harmonizer voice effect raises or lowers the pitch of your voice. A high pitch makes you sound as if you're breathing helium, and a low pitch makes you sound like a baritone singer. In this mode, a single-digit LED readout indicates the pitch change level; 0 is the maximum down shift (Ꝩ%?œ Hz) and 9 is the maximum upshift (+305 Hz). The VFX board powers up in level 4, which is no shift at all. A SHIFT button lets you step through the range of pitch shifts; after 9 the processor returns to 0. Each pitch shift increment is approximately 51 Hz; we'll explain why later.

The echo effect has an adjustable delay; you can decrease the echo delay time by pressing the SHIFT button. In this mode the LED displays a number from 9 to 0 indicating a delay time of 0.63 to 0 seconds. Each press of the SHIFT button decreases the time delay by 70 milliseconds.

The reverb effect is similar to the echo effect, except that the delay time is fixed at 78 milliseconds and the amplitude of the feedback signal is adjustable from 0.5 to 0 with the SHIFT button. The effect is more subtle than the echo effect and simulates the acoustics of a large room.

The test mode helps troubleshoot the VFX board. The test mode will be discussed in greater detail later on.

The basic circuit

As shown in the block diagram (Fig. 1), the VFX processor consists of a microphone input circuit that uses a National Semiconductor TP3054 CODEC (coder-decoder), an Analog Devices ADSP-2105 DSP (digital signal processor), an 8K × 8 EPROM (eraseable program-

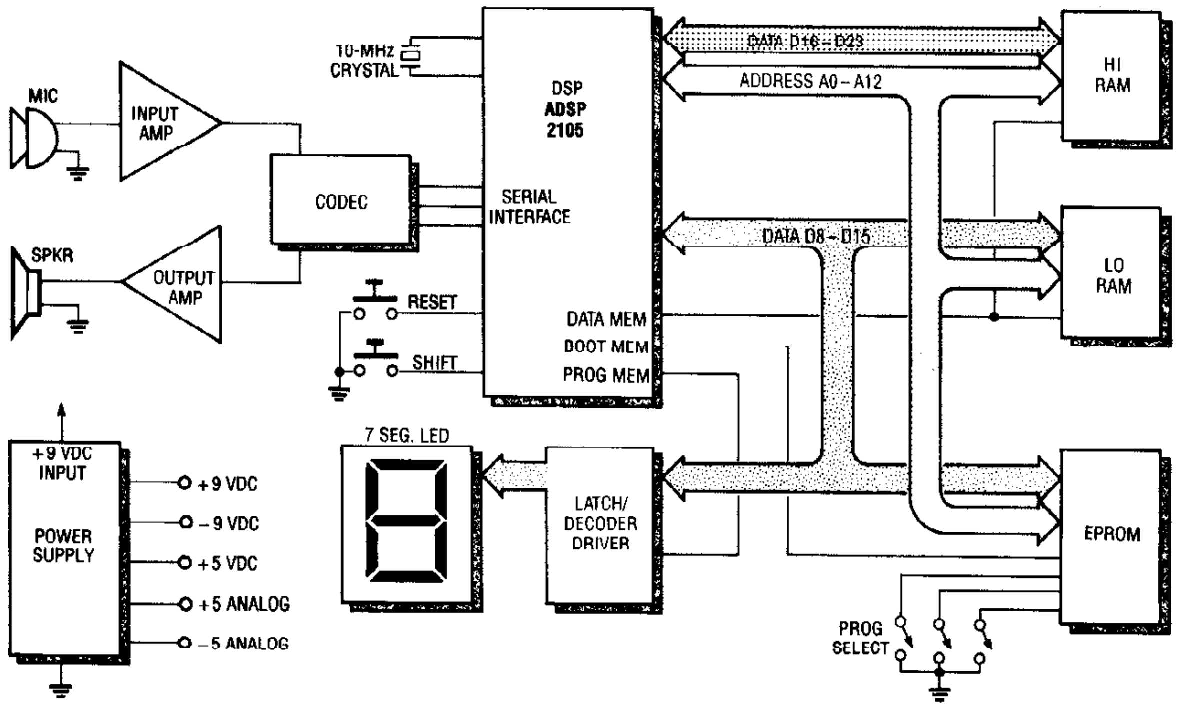


FIG. 1—VFX BLOCK DIAGRAM. The VFX has a microphone input, a CODEC, a DSP, EPROM, SRAM, power-supply circuitry, and audio conditioning.

mable read-only memory), two 8K × 8 SRAM's (static-random access memory), power-supply circuitry, and audio conditioning. The CODEC incorporates an input anti-aliasing filter, an A/D converter, a D/A converter, an output filter, and control circuitry. The SRAM's provide 8K×16-bit words of data storage to supplement the 2105's internal 512 words. The DSP can access external memory in 100 nanoseconds but has an internal wait-state generator to allow the use of slower devices. The VFX processor has three wait states programmed for external data memory access.

The VFX processor performs four functions; the theory of operation for each implemented in hardware is virtually identical. The software makes the hardware perform these multiple effects. The EPROM hex code will be posted on the RE-BBS (516-293-2283, 1200/2400, 8N1), as a file called VFX.HEX. Let's look at the algorithms used for each effect.

Algorithms

The harmonizer shifts the pitch of an audio signal, such as music or speech, up or down. One of the most widely known uses of this technique is seen in the novelty musical group, the Chipmunks. Recorded in the early 60's, the Chipmunks' up-pitch effect was made simply by playing back audio tapes at a higher speed. Today, the high-tech approach is to use digital signal processing.

The principal algorithms performed by the DSP hardware for the harmonizer are the fast Fourier transform (FFT) and the inverse FFT (IFFT). Those algorithms convert the audio signal in the time domain to the frequency domain, and then back again. Figure 2 shows an original audio signal and the data at each stage in the process as it is spectrum-shifted. Figure 2-*a* plots the audio input versus time. Figure 2-*b* shows the frequency spectrum of the audio signal in 2-*a*. Figure 2-*c* shows the original spectrum at the top and the up-shifted spectrum at the bottom. Figure 2-*d* shows the original audio signal on top with the processed audio signal, which contains higher-frequency components, at the bottom.

The timing of the algorithm of the harmonizer is shown in Fig.

TABLE 1—DIP SWITCH SETTINGS

	S1-a	S1-b	S1-c	S1-d
Harmonizer	X	ON	ON	ON
Echo	X	OFF	ON	ON
Reverb	X	ON	OFF	ON
Test Mode	X	OFF	OFF	ON

THE FFT

The Fourier series and its related transforms and algorithms are widely used in electronics. The Fourier transform (FT) is a mathematical method for converting a signal from the time domain to the frequency domain, or simply a way of expressing a continuous waveform as a series of sine waves. The fast Fourier transform (FFT) is an algorithm enhanced for computer computation of a discrete Fourier transform (DFT), which is the digital equivalent of the Fourier transform. **R-E**

3, and its block diagram is shown in Fig. 4. In Fig. 3, the input signal from the microphone is sampled at a 6.5-kHz rate. At that rate buffer #1 is filled in 19.7 milliseconds with 128 samples. (That determines the pitch resolution because the resolution in the frequency domain is the inverse of the sampling period, or 50.7 Hz.) Then the next 128 samples are stored in buffer #2. (The double-throw switch in Fig. 4 is there to suggest the toggling from one buffer to the other.) While buffer #2 is being filled, the VFX processor begins the harmonizing effect by processing buffer #1 through a 128-point FFT, then comes the shift, and then the IFFT. The entire FFT/Shift/IFFT algorithm takes approximately 6 milliseconds so that all processing is finished before the next buffer is filled. That allows real-time processing with a minimal two-buffer delay of 39.3 milliseconds between the time the microphone input arrives at the VFX processor and when it is output to the speaker.

The echo-effect algorithm uses a digital implementation of an adjustable-length analog delay line as shown in Fig. 5. The input signal from the microphone is sampled at a 6.5-kHz rate. It is then summed with the delayed signal received n × 455 periods ago (where n is a number from zero to nine as shown on the VFX's LED display). The delay line is implemented in 4K of external SRAM. The software allows the adjustment of the delay from 0.63 to 0 milliseconds.

The reverb effect is very similar to the echo effect (see Fig. 6) except that the length of the delay line is fixed at 78 milliseconds, and the reflection factor is adjustable from 0% to 50% with the SHIFT button. The reflection factor determines the attenuation of the signal before it is stored in the delay line and simulates the reflection factor of a room.

The test mode can be used during hardware checkout to isolate problems with your VFX board. We'll discuss how to use the test mode later.

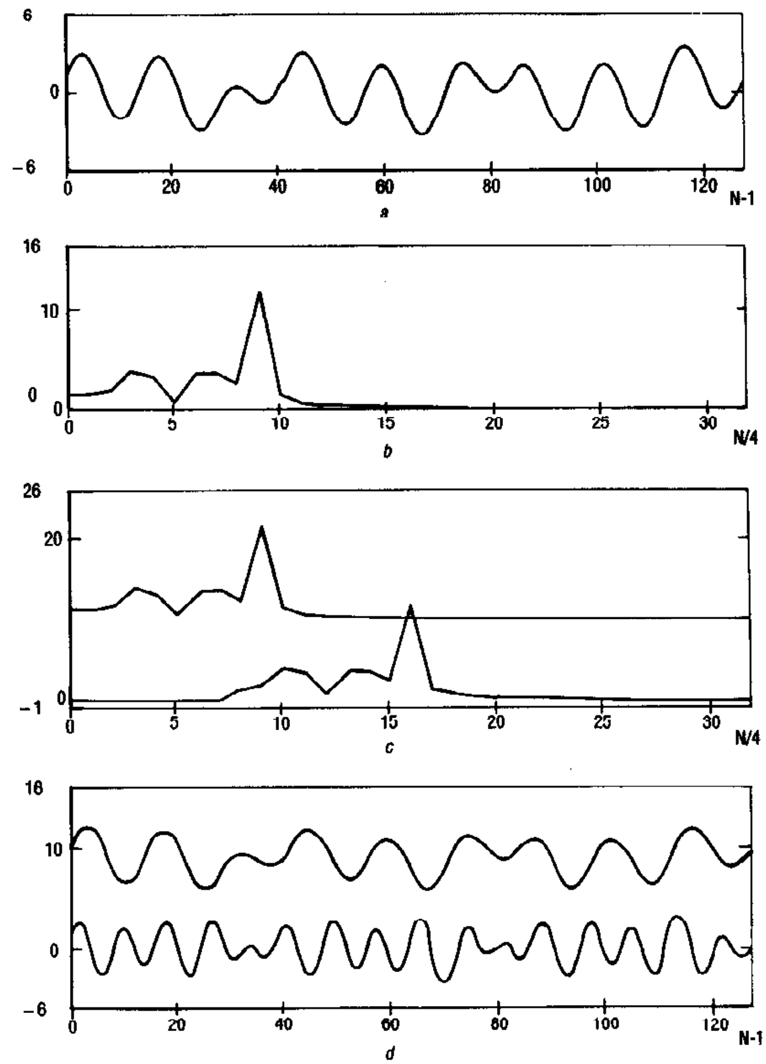


FIG. 2—HARMONIZER ALGORITHM. In *a* you see the audio input vs. time, *b* shows the frequency spectrum of the audio signal, *c* shows the original spectrum at the top and the up-shifted spectrum at the bottom, and *d* shows the original audio signal on top with the processed audio signal at the bottom.

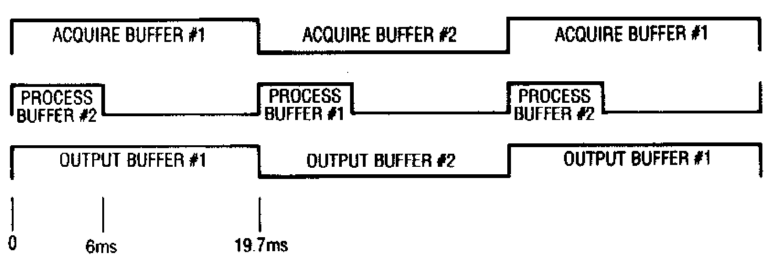


FIG. 3—HARMONIZER TIMING DIAGRAM. The input signal is sampled at a 6.5-kHz rate and fills buffer #1 in 19.7 milliseconds with 128 samples. The next 128 samples are stored in buffer #2.

Circuitry

The schematic diagram for the VFX processor is shown in Fig. 7. The ADSP-2105 DSP microprocessor, IC1, has 1K × 24-bit words of fast program memory (PM) on chip. An on-chip oscillator requires a 10-MHz

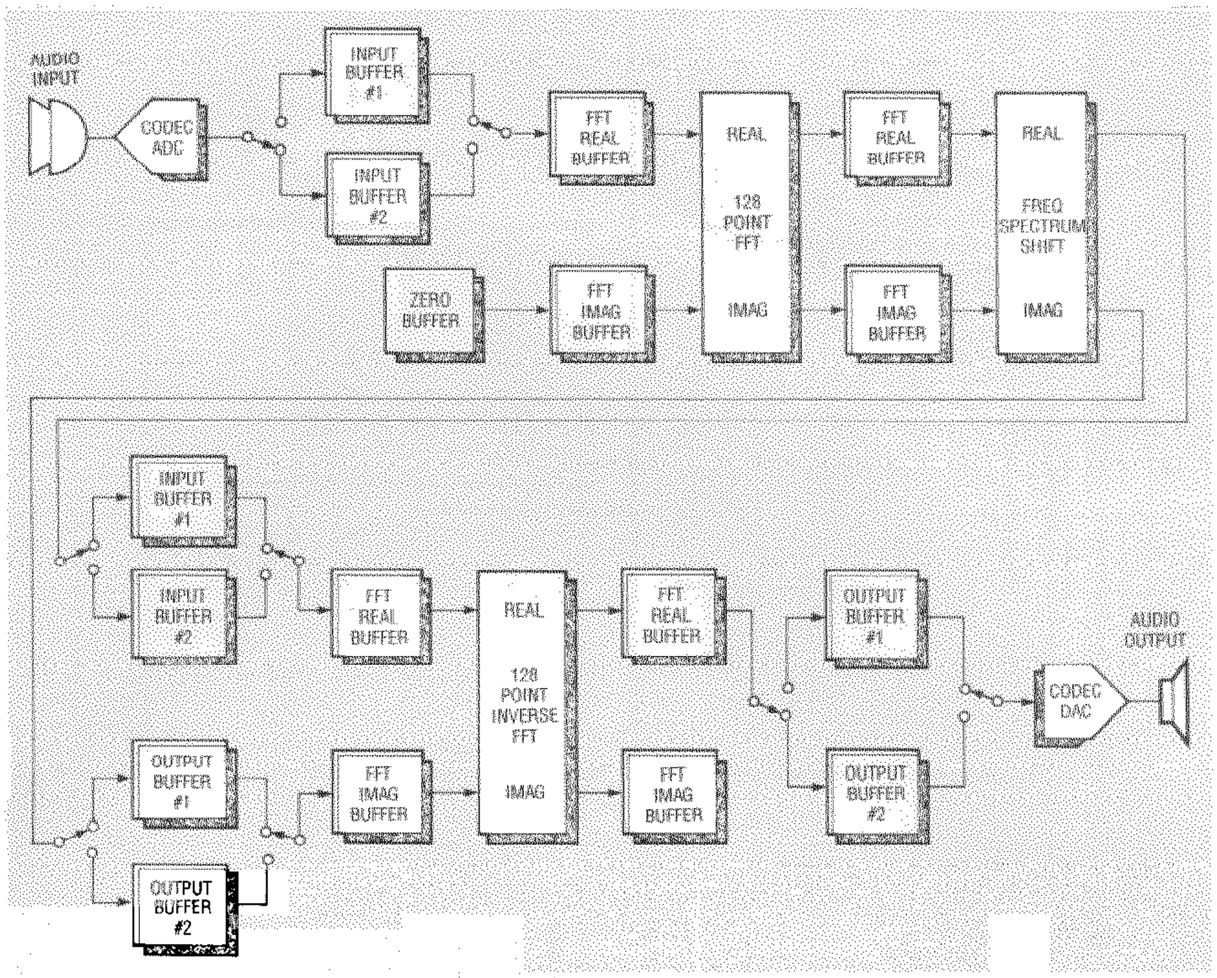


FIG. 4—HARMONIZER BLOCK DIAGRAM. The double-throw switches indicate the toggling from one buffer to the other. While buffer #2 is being filled, the VFX processor begins processing buffer #1.

crystal (XTAL1) and two small capacitors (C1 and C2). On power-up and after a reset, the 2105 boots the program from the EPROM (IC2) into the on-board memory. The boot function is built into the 2105 and it allows a slower and inexpensive EPROM (250 ns) to supply the 1K words (3K bytes) of PM. The BOOT MEMORY SELECT (BMS) output of the 2105 selects the EPROM, and the addressing is automatically generated on the external address bus. The selection of the program booted can be programmed by the 2105, but to simplify the VFX hardware and software, the program is selected by setting the three most-significant bits (MSB's) of the EPROM's address with DIP switch S1.

In addition to the on-board PM, there is 0.5K × 16-bit words

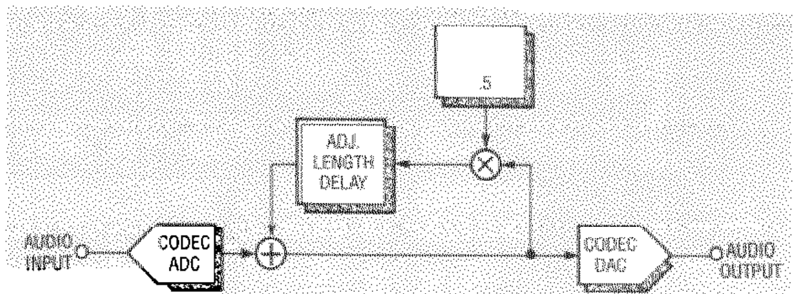


FIG. 5—ECHO-EFFECT BLOCK DIAGRAM. A digital implementation of an adjustable-length analog delay line is used.

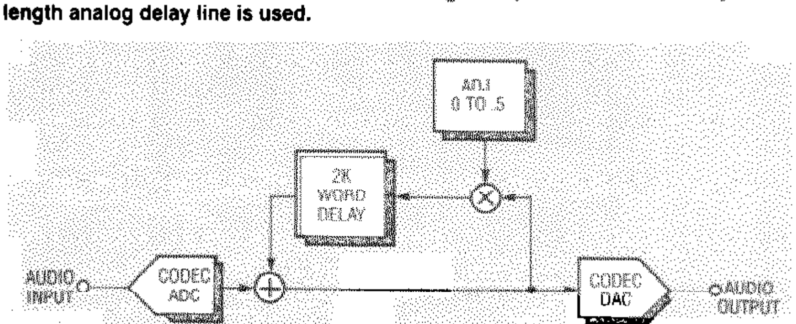


FIG. 6—REVERB EFFECT BLOCK DIAGRAM. The length of the delay line is fixed at 78 milliseconds, and the reflection factor is adjustable from 0% to 50%.

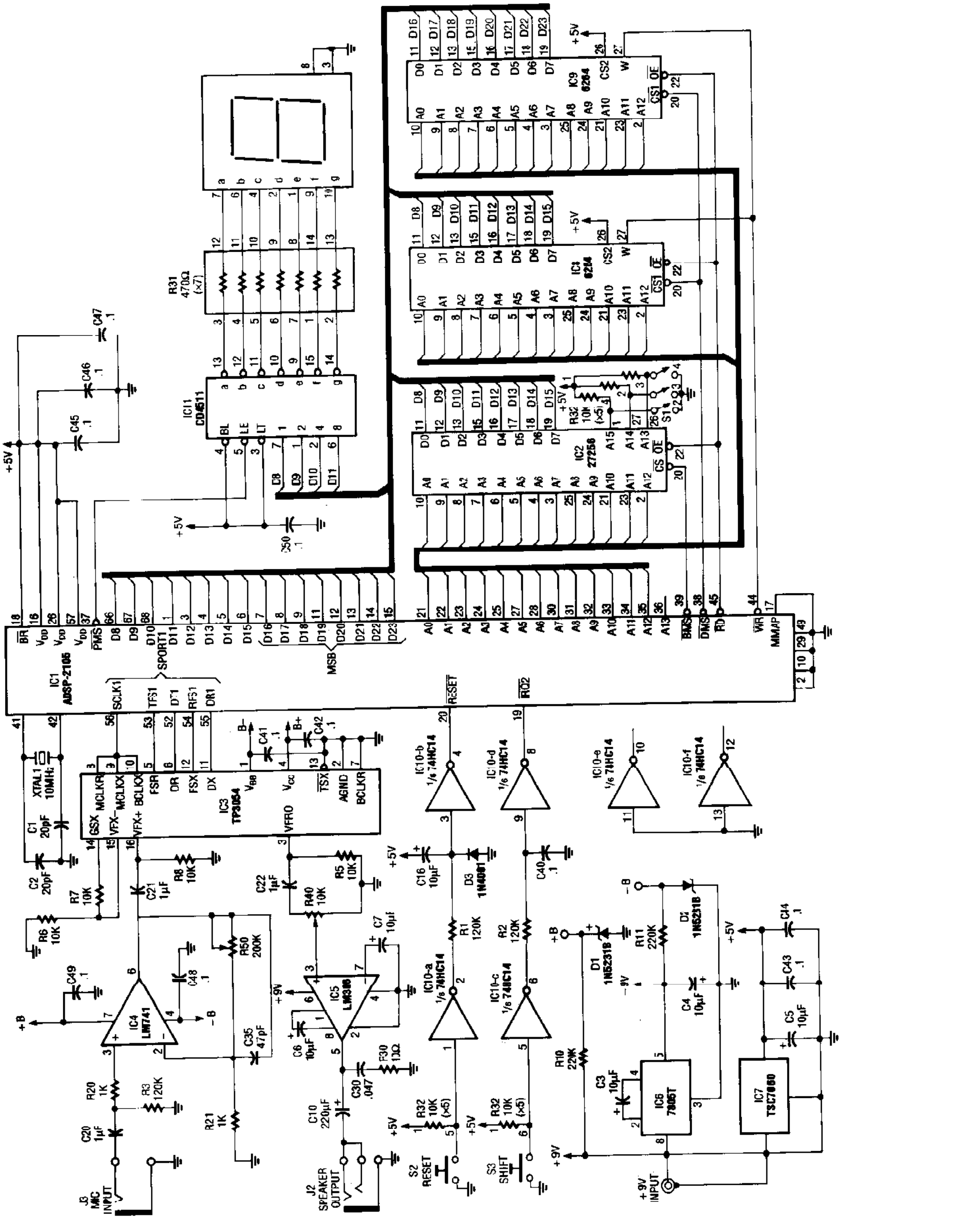


FIG. 7—VFX SCHEMATIC. The DSP microprocessor (IC1) has 1K × 24-bit words of fast program memory (PM) on chip. On power-up, the 2105 boots the program from the EPROM IC2 into the on-board memory.

THE ADSP-2105

The Analog Devices ADSP-2105 is the engine of the VFX processor. The ADSP-2105 is a second-generation digital signal processing (DSP) microcomputer based on the earlier ADSP-2100. It has significant architectural improvements over earlier generations (see block diagram in Fig. 10). The 2105 has built-in data memory RAM (0.5K × 16 bits) and program memory RAM (1K × 24 bits) so it really is a DSP microcomputer and not a microprocessor. Both of those memory banks are expandable with off-chip fast static RAM. That allows the program memory to be loaded, using the resident boot memory loader, from a slow PROM or EPROM (250 nanosecond access) and keeps high-speed data transfer inside the chip to reduce EMI and board-layout requirements.

The program and data memory can be easily expanded off-chip—as has been done with the VFX processor—when the internal data memory is not sufficient for the algorithms. The chip has resources built-in to simplify external memory hardware interfacing. They include separate selects for program memory, data memory, and boot memory, and a programmable wait-state generator to allow for slow external memories.

The 2105 incorporates several peripheral devices and their associated interrupts. There is a built-in 16-bit interval timer with programmable prescaler and interrupts. A high-speed synchronous serial interface (SPORT1) can interface to μ-law and A-law CODEC's using hardware companding as well as digital audio-oriented D/A and A/D converters. Additionally, the serial port can connect multiple 2105's together in parallel processing applications.

The ADSP-2105 also offers high performance by virtue of its instruction set. With a 100-nanosecond cycle time, multiple operations per cycle, and zero-overhead looping, the numerical performance of the chip is respectable. In addition, the 1-micron low-power CMOS processing holds power dissipation to less than 1 watt; a powerdown mode reduces the power consumption to a mere 80 milliwatts.

The ADSP-2105 incorporates three execution units:

- Barrel shifter
- Arithmetic Logic Unit (ALU)
- Multiplier Accumulator (MAC)

The three units are optimized for their specific function, and are, therefore, very fast, completing any instruction in one cycle. Access to the three execution units is made via registers associated with each execution unit. For example, the ALU has the following 16-bit registers:

AX0, AX1, AY0, AY1, AR, and AF

The MAC has:

MX0, MX1, MY0, MY1, MR0, MR1, MR2,

LISTING 1

```
START:
            I0=buffer#1;                   {Address of buffer#1 in DMD}
            M0=1;                          {Post modify value}
            I4=buffer#2;                   {Address of buffer#2 in PMD}
            M0=1;                          {Post modify value}
         CNTR=2048;                        {Number of words in buffer}
DO MOVE_BUFFER UNTIL CE;                   {Do loop}
AR=PM(I4,M4);
DM(I0,M0)=AR;
MOV_BUFFER:                  {End of the loop}
```

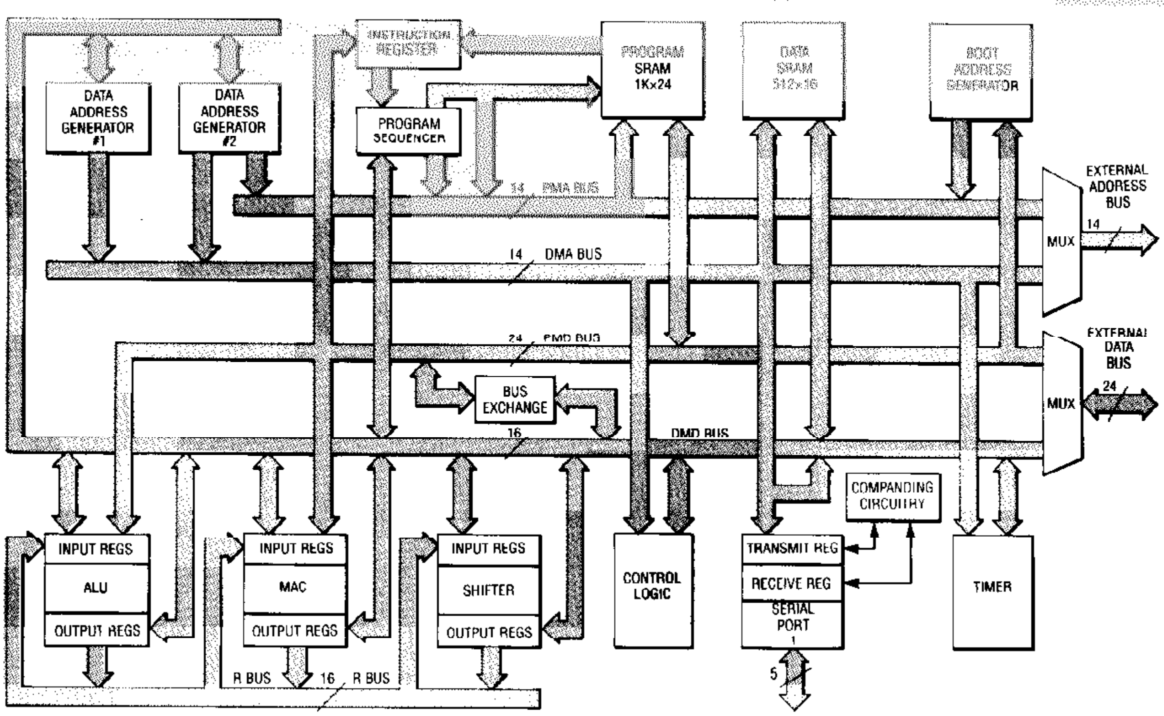


FIG. 10 ADSP 2105 BLOCK DIAGRAM. The 2105 has built-in data memory and program memory implemented in fast SRAM. That keeps high-speed data transfer inside the chip.

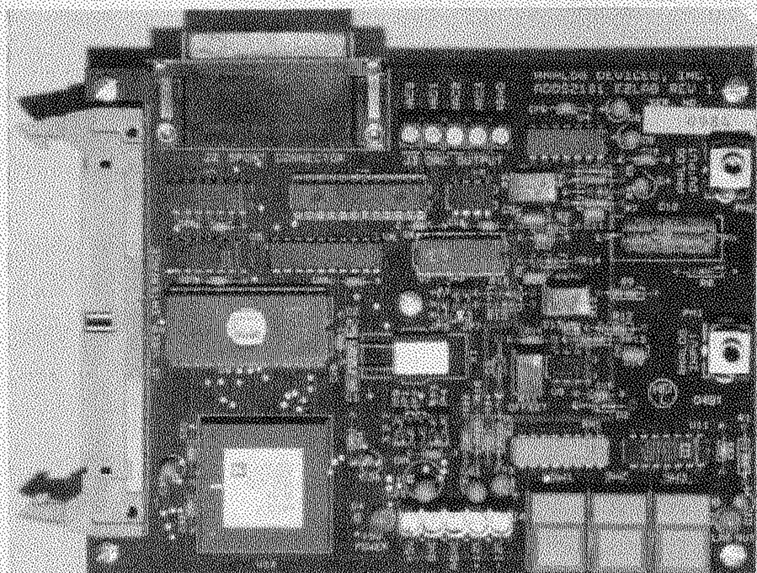


FIG. 11—THE EZ-LAB KIT includes a demo board, an ADSP-2101/5 family assembler, linker, and other miscellaneous development software. You can design and debug software for certain applications at minimal cost.

and MF
The barrel shifter has:
SB, SI, SE, SR1, and SR0

In addition, the ADSP-2105 incorporates two data address generators (DAG's), one of which can perform the bit reversing that is required for certain FFT algorithms. One DAG accesses program-memory data (PMD) and the other accesses data-memory data (DMD). The DAG's use a set of three registers to control indirect addressing and circular buffers. Those are the index registers (I0 to I7), the modify registers (M0 to M7), and the length registers (L0 to L7). For example, by setting up the registers so that I0 has the starting address, M0 has a value of 1, and L0 is zero, blocks of data can be moved from one buffer to another with very little programming (see Listing 1).

To be successful, any microcomputer, including a digital signal processor, must have readily available low-cost software tools. Analog Devices has supplied the ADSP-2105 with quality software tools at a reasonable price. The assembly language is algebraic and straightforward. Included is a powerful personal computer-based software simulator that allows software debugging without an expensive in-circuit emulator (ICE).

The ADSP-2105 has the same kind of interrupt handling capabilities as other microcomputers. The interrupts can be individually masked or enabled, edge-triggered or level sensitive. Interrupts are vectored to the program memory locations shown in Listing 2. A second mirror set of Data registers can be enabled to facilitate fast context switching during its interrupt servicing. The device has on-chip clock generation circuitry and is packaged in a 68-lead plastic leaded chip carrier (PLCC).

LISTING 2

```
IRQ2                 0004h
SPORT0 (Transmit)    0008h
SPORT0 (Receive)     000Ch
SPORT1 (Transmit)    0010h
SPORT2 (Receive)     0014h
TIMER                0018h
```

The VFX processor described in this article was developed with Analog Devices' ADSP-2101 EZ-Lab Kit (see Fig. 11). The Analog Devices EZ-Lab kit includes an EZ-LAB demonstration board, an ADSP-2101/5 family assembler, linker and other miscellaneous development software, including the essential simulator. With this package one can design and debug software for certain applications with excellent results at minimal cost. Of course, an in-circuit emulator (ICE) will speed up the development process, although, of course, at a much higher price: the EZ-Lab kit sells for less than $500 dollars, and an emulator costs more than $2000 dollars. For people with limited capital resources and small to medium complexity algorithms, the kit is great. **R-E**

of on-board data memory (DM). Since that is not enough to perform the 128-point FFT and IFFT, two external static RAM's are also attached to the data bus, one for the high byte (IC9), and one for the low byte (IC8) of the memory. That 2×8K bytes of SRAM addressed by the 2105 is accessed when the $\overline{\text{DATA MEMORY SELECT}}$ ($\overline{\text{DMS}}$) strobe is active.

The seven-segment LED display adds to the interactivity of the VFX processor, and is written to as if it was external program memory. The $\overline{\text{PROGRAM MEMORY SELECT}}$ ($\overline{\text{PMS}}$) signal from the 2105 is activated to latch data from the bus into IC11 (the seven-segment BCD latch/decoder/driver), which then drives the seven segment display. No decoding is required for the selection of IC11 because there is no external program memory in the system.

The VFX processor uses a CODEC to digitize the audio input and convert it into a serial data stream. The CODEC interfaces directly with the 2105's synchronous serial port SPORT1, which includes pins 52–56. SPORT1 is configured for 8-bit synchronous data transfer with word-framing sync pulses and μ-law companding. The 2105 generates a 1.66-MHz serial clock (SCLK1) and 6.5 kHz framing pulses on TRANSMIT FRAME SYNC (TFS1) and RECEIVE FRAME SYNC (RFS1) to synchronize the data transfer.

The CODEC implements μ-law companding, which improves the dynamic range of the conversion by taking advantage of human perception of sound; that is, that the ear is much more sensitive to noise in low-level (volume) signals than in high-level signals. The CODEC receives and transmits 8 bits of data, and the digital signal processor has built-in companding hardware to convert it into a 14-bit number.

The other components of the VFX processor are the power supply and analog components. The VFX board accepts +9 volts DC and generates −9, +5, +5.1, and −5.1 volts DC. Voltage converter IC7 (a TSC7660) generates the negative supply from the positive supply. A 5-volt DC regulator (IC7, a 7805) supplies +5 volts DC to the 2105 and all logic IC's, and two Zener regulators (D1 and D2) generate the analog voltages of plus and minus 5.1 volts DC. Op-amp IC4 and audio amplifier IC5 condition and amplify the audio input and output, respectively.

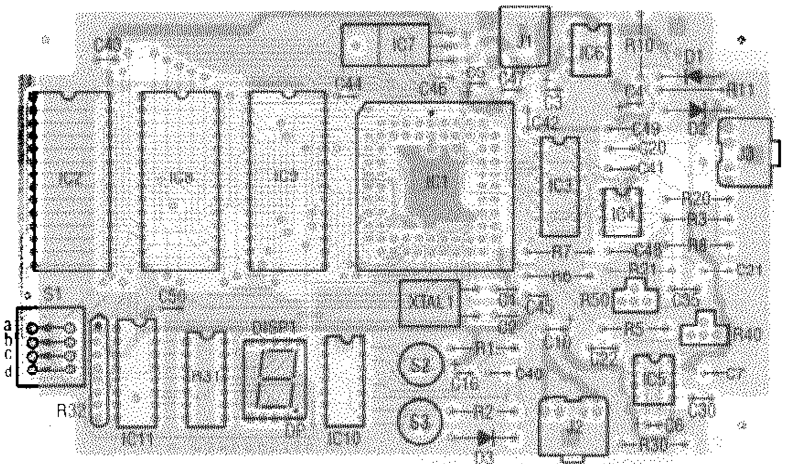


FIG. 8—PARTS-PLACEMENT DIAGRAM. Make sure that you install the mono jack at J1 and the stereo jack at J2.

PARTS LIST

All resistors are ¼-watt, 5%, unless otherwise noted.

R1–R3—120,000 ohms
R4, R9, R12–R19, R22–R29, R33–R39, R41–R49—not used
R5–R8—10,000 ohms
R10, R11—220,000 ohms
R20, R21—1000 ohms
R30—10 ohms
R31—470 ohms (×7), 14-pin DIP
R32—10,000 ohms (×5), 6-pin SIP
R40—10,000 ohms, multiturn potentiometer
R50—200,000 ohms, multiturn potentiometer

Capacitors

C1, C2—20 pF, 100 volts, ceramic
C3–C7—10 µF, 35 volts, electrolytic
C8, C9, C11–C15, C17–C19, C23–C29, C31–C34, C36–C39—not used
C10—220 µF, 25 volts, electrolytic
C16—10 µF, 6.3 volts, Tantalum electrolytic
C20–C22—1 µF, 50 volts, ceramic
C30—0.047 µF, 100 volts, ceramic
C35—47 pF, 100 volts, ceramic
C40–C50—0.1 µF, 100 volts, ceramic

Semiconductors

IC1—ADSP-2105KP40 DSP processor
IC2—27256-25 32K×8 EPROM (256K)
IC3—TP3054J CODEC
IC4—LM741N op-amp
IC5—LM386N-3 audio amplifier
IC6—7660SCPA voltage converter
IC7—7805T 5-volt regulator
IC8, IC9—6264-15 SRAM, 150 ns
IC10—74HC14N hex Schmitt trigger inverter
IC11—CD4511 7-segment decoder/driver
D1, D2—1N5231B 5.1-volt Zener diode
D3—1N4001 diode
DISP1—LTS6780R 7-segment common cathode LED

Other components

XTAL1—10-MHz crystal
S1—4-position DIP switch
S2, S3—momentary pushbutton, N.O.
J1—2mm DC power jack
J2—mini stereo jack
J3—mini mono jack

Miscellaneous: IC sockets, 9-volt DC wall transformer, microphone, headphones, PC board, solder, etc.

Note: The following items are available from American Distributors, Inc., 9 Whippany Road, Whippany, NJ 07981 (800) 877-0510:

- **VFX kit (includes PC board and all PC-mounted components)—$105 (plated through holes, solder masked and silkscreened)**
- **9-volt wall transformer—$12**
- **Headphones—$15**
- **Microphone—$16**

Add $5 shipping and handling. Check, MasterCard, or Visa.

Construction

The VFX processor is easy to build. All the necessary components including a double-sided PC board, all IC's, semiconductors, and passive components are available from the source given in the Parts List. A microphone, DC wall outlet transformer, and headphones are also available if you don't already have them. We've provided foil patterns in case you want to make your own PC board.

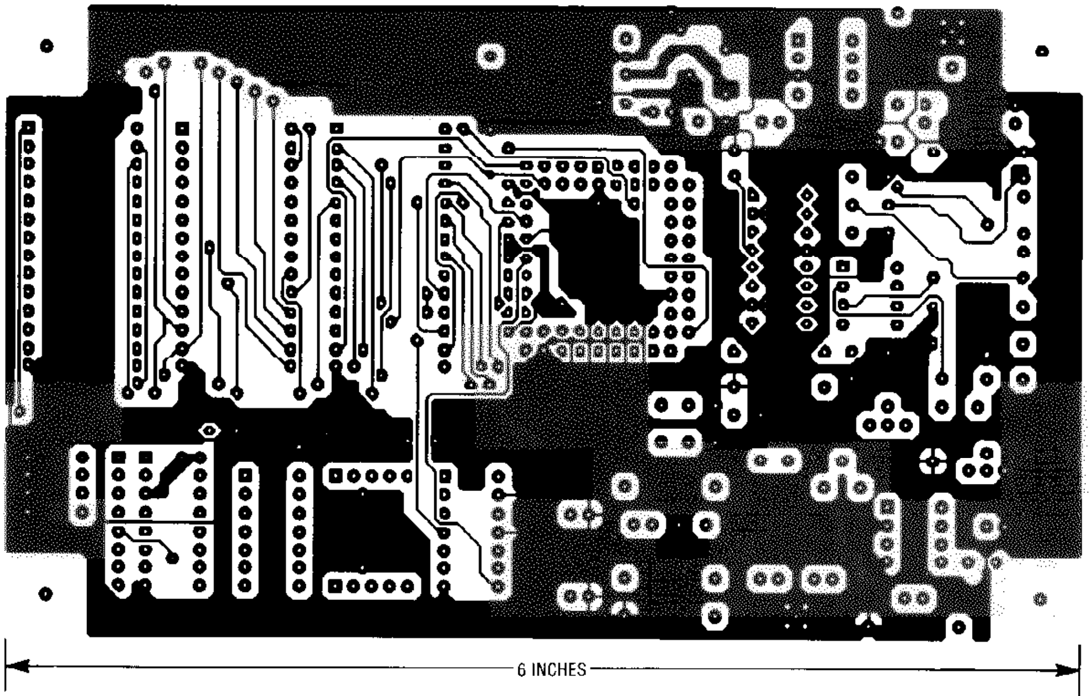
Following Fig. 8 as a guide, mount the components beginning with the resistors. Next install the capacitors, the crystal, switches, jacks, voltage regulator (IC7), and then all of the IC sockets. Make sure you don't install the input and output jacks in the wrong locations. The output jack has three terminals so that if you use headphones you'll hear sound from both sides. Be sure to orient the polarized capacitors correctly. Do not install the IC's yet and don't remove them from their packaging just yet either. When you've completed the soldering, carefully double check parts placement and look for solder splashes and bridging. The completed VFX card is shown in Fig. 9.

Hardware checkout

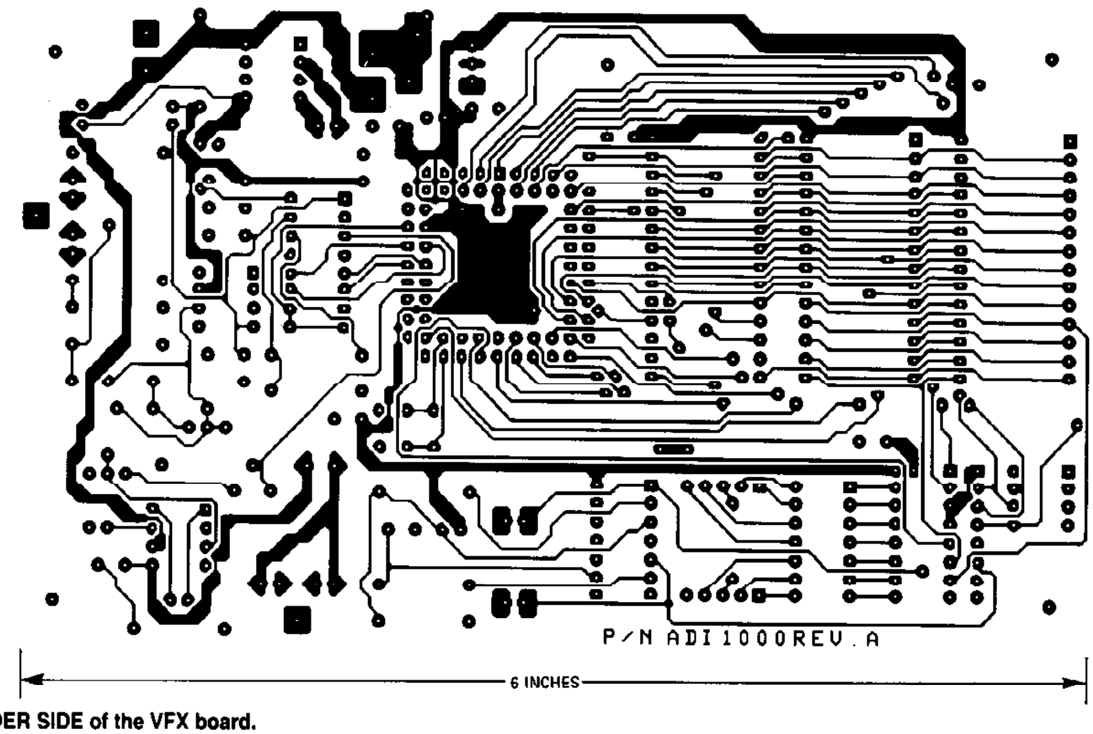
Now we will perform a hardware checkout, one IC at a time. Precautions against static discharge should be followed when handling the IC's. Electrostatic discharge can cause very subtle damage in the IC that can be hard to find—the kind that is

FILTER DESIGN SOFTWARE

The digital filter used in the echo and reverb effect was designed with the filter and digital analysis software (FDAS) from Momentum Data Systems software. The software was used to implement a finite impulse response (FIR) filter. The optional code generator wrote the source code for the filter, given its characteristics. The package also runs on the ADSP-2101 processor and other family members. The FDAS software can also implement other kinds of digital filters than FIR, including infinite impulse response (IIR) and some analog equivalents. **R-E**



COMPONENT SIDE of the VFX board.



SOLDER SIDE of the VFX board.

worth avoiding.

First apply 9-volt DC power to J1 and verify that there is 5 volts on the power pins of each IC socket. When verified, remove power from the board and plug in IC6, the negative-supply generator. Now reapply 9-volts DC to J1 and measure pin 5 of IC6; the voltage should be the negative of the voltage on IC6 pin 8 or −12 volts (whichever is less).

Next, check pins 1 and 4 of IC3 for −5 volts ± 0.2 volt and +5.0 ± 0.2 volt, respectively. Install IC1, IC2, IC10, and IC11 in their respective sockets. For the RAM test, set all the S1 DIP switches to the "on" position and apply power to the board. The LED display should show the number "6." Press the SHIFT

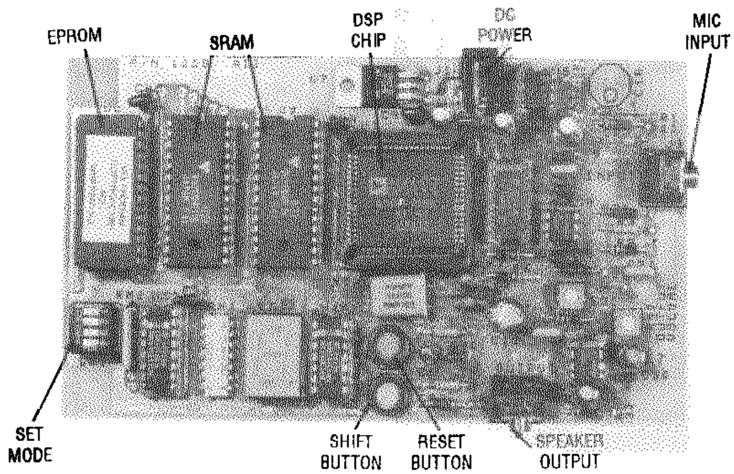


FIG. 9—THE VFX PROTOTYPE. Carefully check the board for solder splashes and bridging before applying power.

button and the LED display should count fast enough so that all the segments (an "8") appear dimly lit.

Remove power from the board and install the two RAM's (IC8 and IC9). Apply power to the board. The LED should again display "6." Press the SHIFT button and the LED should display "0." If any RAM errors occur, they will cause the LED display to increment.

Next install IC3 and IC5. Connect your speaker or headphones to J2 and reapply power. Press the SHIFT button twice and a tone should be heard in the headphones or speaker. Install IC4 and connect a microphone to J3. Apply power and press the SHIFT button three times. Then speak into the microphone and your voice should be heard through the headphones. Adjust potentiometers R50 and R40 for minimum distortion. Now that your VFX board is working, you can change the DIP switches according to Table 1 for the other three effects.

As mentioned before, there is a test mode that can help troubleshoot the VFX processor. It is activated by setting the DIP switches as shown in Table 1 and pressing the reset button. The test mode individually tests the system RAM, the CODEC, and the LED display.

In the test mode the external SRAM is constantly written to and read, and the number of errors are displayed on the LED. If the LED display is blank and all the power supplies are normal, there is something wrong with the LED or the driver. If the LED has a number other than zero, there might be a problem with the SRAM.

The CODEC data is received and immediately retransmitted, so the microphone input is echoed back the headphones. If there is no output or if the output doesn't sound like the input, there is a problem. If there are no other fault indications and the microphone and speaker are working, there might be a problem with the CODEC. If nothing happens and the power supplies are normal, there might be a problem with the digital signal processing chip or the EPROM.

Where to go from here

The VFX processor is intended to demonstrate in, an enjoyable way, the capabilities of digital signal processing. The four applications programmed into the VFX board are just four out of many possible applications. The VFX processor hardware is capable of being reprogrammed to perform other functions as well. Some of the possibilities are speech recognition, active noise cancellation, voice compression/recording, and a spectrum-shifting hearing aid.

For example, the VFX processor could easily recognize the numbers from 0 to 9 and display them on the LED indicator. That requires that the speech be converted into the frequency domain and the spectral peaks of the sound be compared with pre-stored templates. The closest matching sound is selected and displayed on the LED. The processor could then generate the DTMF signals for that number to make a voice-activated telephone dialer.

A voice compressor/recorder converts an audio input into the frequency domain, picks out the most prominent spectral energies, and stores them in data memory as frequency and amplitude. The technique can reduce the amount of data that must be stored compared to that from conventional digitizing processes from 6.5K words per second to 650 to 300 words per second. The VFX board with 8K words can record approximately 12 to 25 seconds of compressed speech.

Active noise cancellation is being developed for applications ranging from muffling the sound of automobile engines and industrial machines to eliminating the background hissing noise in fighter-aircraft intercom-system headphones. Similar applications for the VFX board are being developed. Let us know if you have any other applications you would like programmed into the VFX processor. If you are interested in programming your own applications, look into the EZ-LAB system sold by Analog Devices that has been referenced in this article. It is an affordable way to implement small- to medium-sized algorithms.. **R-E**