48783

Radio-Electronics

PROTECT YOUR CAR—BUILD AN AUTOMOTIVE BURGLAR ALARM

$1.50 MAY 1983
U.K. 85p
DGS

COMPUTERS - VIDEO - STEREO - TECHNOLOGY - SERVICE

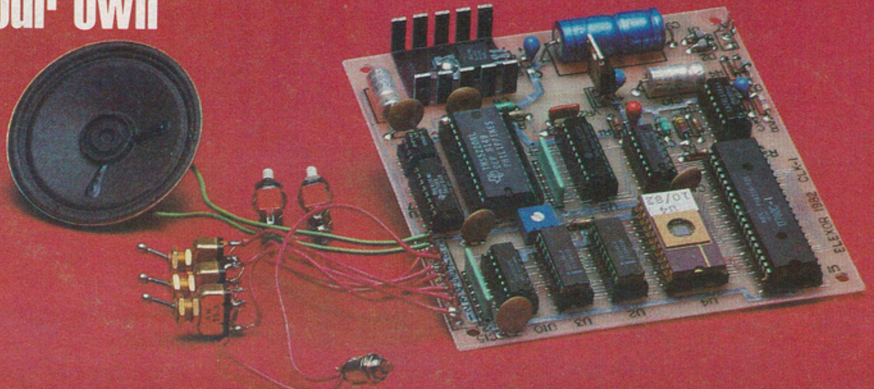
Build this **TALKING ALARM CLOCK** and listen to the time

New IC's for **MUSIC SYNTHESIZERS** simplify design

Build a **VLF ANTENNA TUNER** for your receiver

How to **REWIND TRANSFORMERS** for custom applications

Back-to-school series: **AUDIO POWER AMPLIFIERS** How to design your own



0 71896 48783 05

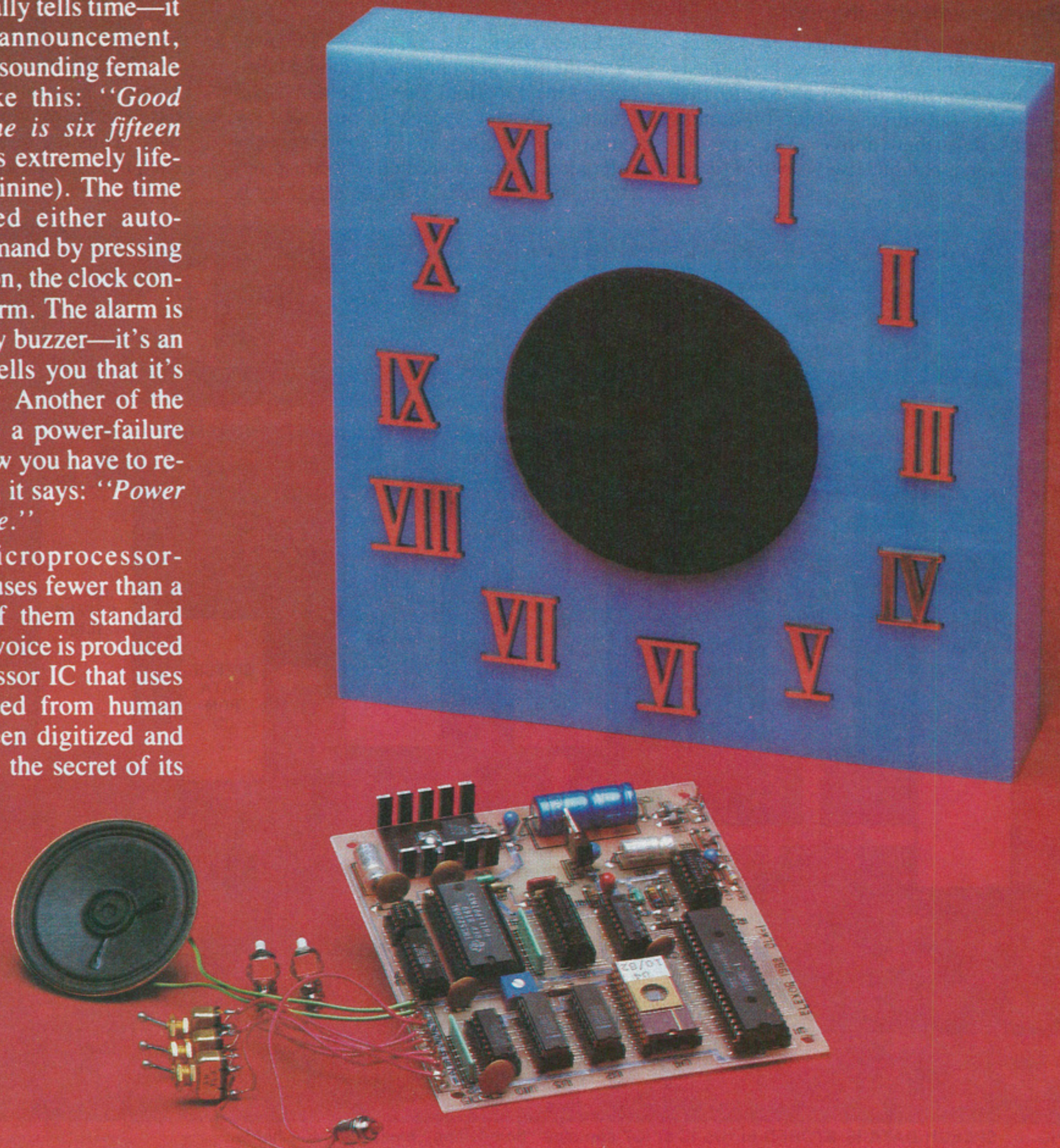
TALKING ALARM CLOCK

Build a talking alarm clock and you'll never have to tell time again—the clock will do it for you.

LEE GLINSKI

HERE'S A LOW-COST ELECTRONIC alarm clock that really tells time—it talks. The time announcement, made in a pleasant-sounding female voice, sounds like this: *"Good morning. The time is six fifteen AM."* The voice is extremely life-like (and very feminine). The time can be announced either automatically or on demand by pressing a switch. In addition, the clock contains a 24-hour alarm. The alarm is not just an ordinary buzzer—it's an actual voice that tells you that it's *"time to get up."* Another of the clock's features is a power-failure alarm. You'll know you have to reset the clock when it says: *"Power failed. Set the time."*

The entire microprocessor-controlled device uses fewer than a dozen IC's, all of them standard parts. The clock's voice is produced by a speech-processor IC that uses speech data derived from human speech that has been digitized and compressed; that's the secret of its excellent sound.



Human speech

Before describing electronic speech-synthesis, it is first necessary to have an understanding of how human speech is generated. The voice-producing mechanism in human beings consists essentially of two parts—the sound source and the vocal tract. The speech process starts with air being pushed out from the lungs. The resulting air stream stimulates the vocal cords, and causes sounds to be produced. Those are called *voiced* sounds, examples of which are vowels like "U" and "A." If the vocal cords are held open so they don't vibrate, the sound produced will be *unvoiced*, like the consonants "S" and "F."

The basic sounds enter the vocal tract—made up of the mouth, nasal passages, and other resonant cavities inside the head, throat, and chest—where they are shaped into speech. Changing the shape characteristics of the vocal tract produces different sounds.

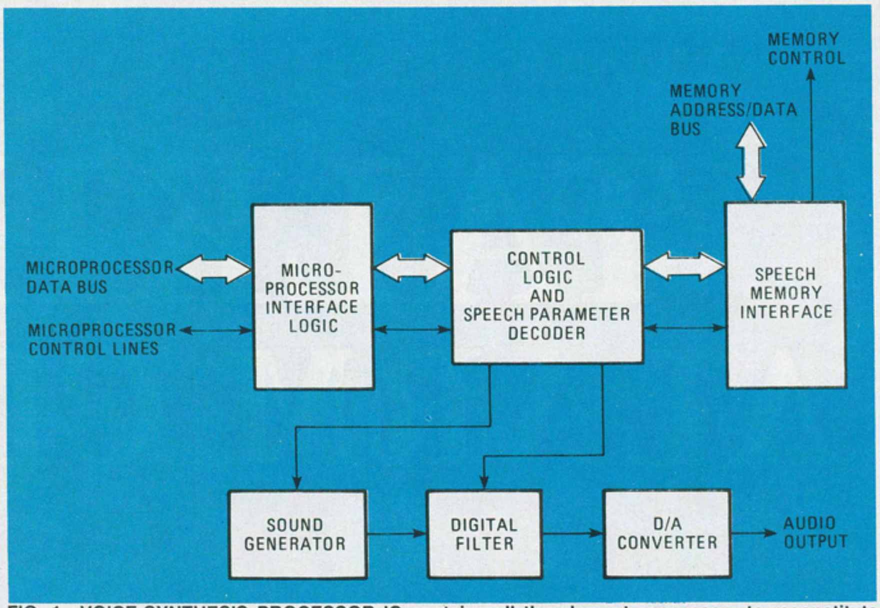


FIG. 1—VOICE-SYNTHESIS PROCESSOR IC contains all the elements necessary to reconstitute speech from compressed data stored in ROM.

Speech-synthesis theory

The voice of the talking clock is generated electronically by a speech-synthesis IC, the Texas Instruments TMS5220, that simulates the human voice-producing organs described above. The speech-generation technique used is called *linear predictive coding*.

Linear predictive coding, or LPC, uses a mathematical technique to model (simulate) the functions of the human vocal tract. Coherent speech is produced by stringing together many short speech-elements. Linear predictive coding determines how each of those elements is generated. Each speech element is generated by mathematical calculations, and a formula generates each new element, based on the previous ones plus some new data. Thus the term *predictive* coding—each new speech element is partially predicted from the previous ones.

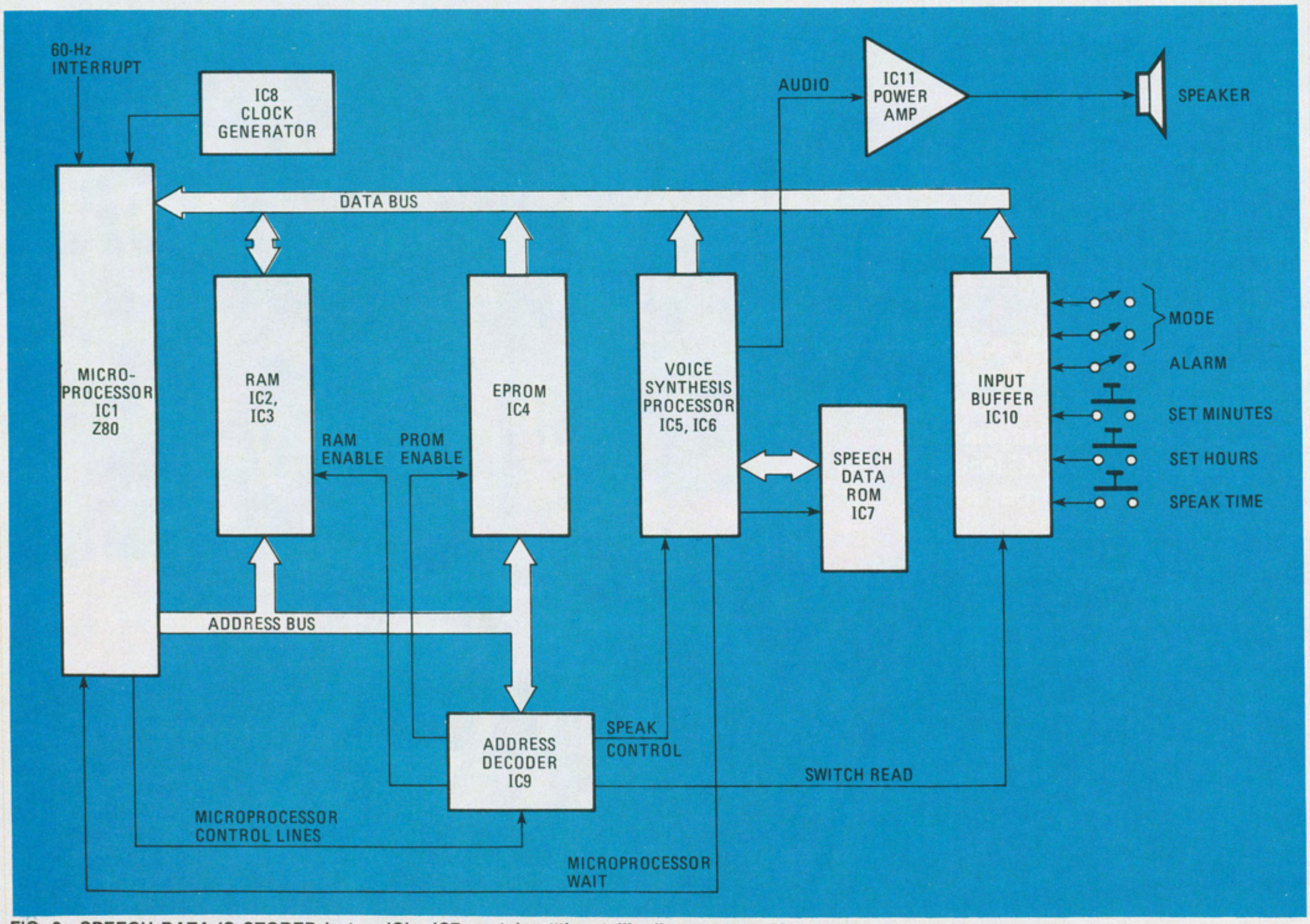


FIG. 2—SPEECH DATA IS STORED in two IC's: IC7 contains "time-telling" messages; IC4 holds messages for alarm and other functions.

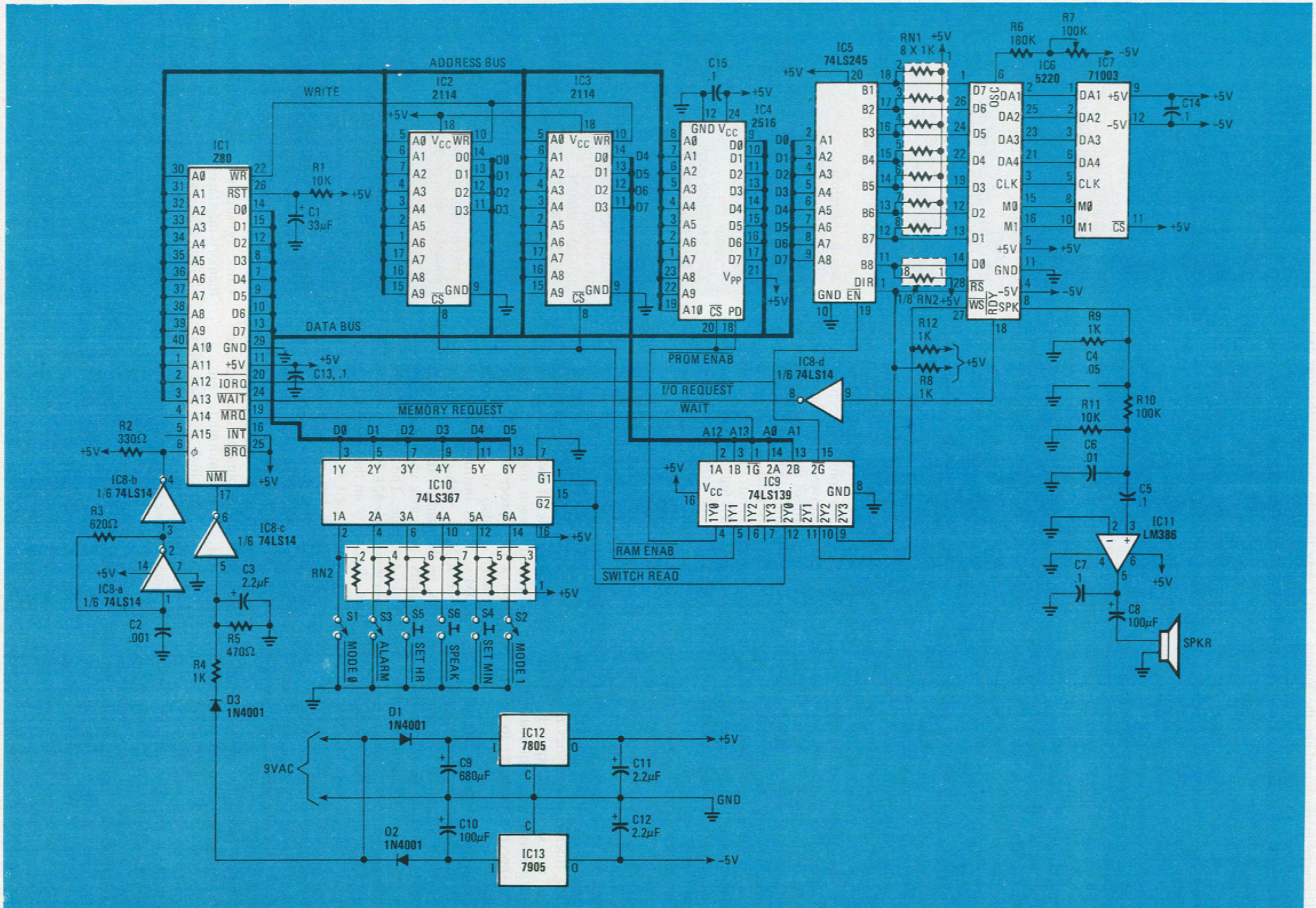


FIG. 3—BOLD LINES in schematic of talking clock represent multiple data and address lines.

The synthesizer simulates the human voice source and the vocal tract. As shown in Fig. 1, the voice is simulated by a sound generator, and the vocal tract is simulated by a digital filter. That digital filter is the mathematical model that performs the calculations to generate speech. Both the sound generator and the digital filter change their characteristics continuously as speech is produced.

There are two sound generators: a variable-frequency generator to simulate voiced sounds from the vocal cords, and a noise generator to simulate unvoiced noise-like speech sounds.

The digital filter shapes the signals from the sound generator to produce small time-samples of speech. Its characteristics can be altered to produce different sounds.

Each word produced by the synthesizer consists of many time-samples in sequence. During voice generation, one of the sound sources is selected, and the values of its pitch and loudness set. The sound source is then fed to the digital filter. The parameters of the filter are then programmed to shape the sound source into the desired speech pattern. The filter generates each speech sample from a calculated sum of the previous 10 samples. That is done to minimize the amount of data required to generate each new sample, and is the main characteristic of linear predictive coding.

The information that determines the characteristics of each sample is the *digital speech data*. That data is a description of certain parameters of the original spoken words. It contains parameters to describe the voice frequency, strength, and the filter characteristics required to create the synthetic speech. During speech generation the required data is fed to the speech synthesizer to control its operation.

A collection of speech data for a number of words makes up a speech synthesizer's vocabulary. To generate a vocabulary for the speech synthesizer, the words are first spoken and recorded on a high-quality master tape. Each word from the tape is sampled and digitized at an 8-kHz rate, and the resulting data is then fed to a computer for analysis. That's done to compress the data so that a minimum of memory is needed to store it. Typically, the data will be compressed by a factor of 100 or more.

Computer programs analyze the data using a mathematical model of the human speech-producing "mechanism." The computer extracts parameters from the data that describe the speech in terms of vocal-tract qualities, pitch, and energy level as a function of time. Once those values have been extracted, other computer programs further analyze and compress the data. That will produce speech data that can be used by the synthesizer for voice generation.

The compressed speech-data is coded in a way that the voice synthesizer can read and use effectively, and is stored in a ROM (*R*ead *O*nly *M*emory). The voice synthesizer reads the data contained by the ROM, performs the mathematical calculations to simulate the vocal tract, and produces synthetic speech.

Voice synthesizers

The voice synthesizer IC used by the talking clock is manufactured by Texas Instruments. It's their TMS5220 voice-synthesis processor (VSP), and contains all the circuitry necessary to interface with a microprocessor and to generate speech. The VSP (refer to Fig. 1) consists of three major sections: the speech synthesizer itself, the microprocessor interface, and the speech-memory interface.

The speech-synthesizer section of the VSP uses the LPC method described earlier.

The TMS5220 uses a digital filter to simulate the action of the human vocal tract. The filter takes highly compressed LPC speech data from the speech memory ROM and processes it. Its output consists of another form of digital data, which is no longer compressed. The data—now in an expanded format—is a direct digital representation of the original speech waveform and is fed to an 8-bit digital-to-analog (D/A) converter, which outputs an analog voltage reproducing the original audio waveform. The voltage is then filtered to eliminate digitizing noise, and fed to an amplifier and speaker.

As explained previously, the speech synthesizer needs compressed digital speech-data to generate speech. The TMS5220 was designed to accept speech data from one of two sources: from a dedicated speech memory, or directly from a microprocessor. The dedicated memory consists of specially designed ROM's. Texas Instruments has several voice ROM's, with different vocabularies, on the market. Industrial, avionics, military, and clock vocabularies are currently available. The voice ROM's are memories either 32K bits or 128K bits in size, depending on the vocabulary size.

The ROM used (a VM71003) has a capacity of 32K-bits (in a 16-pin package) and contains data for 34 words (a 128K ROM stores over 200 words). It contains words for all the numbers needed to announce the time, as well as words for other clock-related phrases like "the time is," "AM," "good morning," etc.

In addition, the clock also uses other phrases, such as "power fail" and "set the time." Those phrases are not stored in the clock-vocabulary ROM; they are stored in an EPROM (*E*rasable *P*rogrammable *ROM*) that also stores the program that runs the clock. That speech data is read from the PROM by the microprocessor and fed to the VSP through its microprocessor interface.

The voice ROM is connected to the voice synthesis processor through a memory-interface bus. That bus consists of four address lines and two control lines. The voice ROM is specially designed to work with the TMS5220 through it. When the VSP reads data from the ROM, it first sends an address to the memory IC, and then begins reading the data one-bit-at-a-time in a serial fashion. It generates speech as the data is read. During speech generation the data rate is approximately 1200 bits per second.

continued on page 106

PARTS LIST

All resistors ¼ watt, 5% unless otherwise noted

R1, R11—10,000 ohms
R2—330 ohms
R3—620 ohms
R4, R8, R9, R12—1000 ohms
R5—470 ohms
R5—180,000 ohms
R7—100,000 ohms, PC-mount trimmer potentiometer
R10—100,000 ohms
R13, R14—8 × 1K SIP (*S*ingle *In*-line *P*ackage) resistor pack

Capacitors

C1—330 µF, 10 volts, electrolytic or tantalum
C2—0.001 µF, ceramic disc
C3, C11, C12—2.2 µF, 10 volts, electrolytic or tantalum
C4—0.05 µF, ceramic disc
C5, C7, C13-C15—0.1 µF, ceramic disc
C6—0.01 µF, ceramic disc
C8, C10—100 µF, 16 volts, electrolytic
C9—680 µF, 16 volts, electrolytic

Semiconductors

IC1—Z80 microprocessor
IC2, IC3—2114 1K × 4 RAM
IC4—2516 or 2716 2K × 8 EPROM, preprogrammed
IC5—74LS245 octal bus transceiver
IC6—TMS5220 voice-synthesis processor
IC7—VM71003 clock-vocabulary ROM
IC8—74LS14 hex inverting Schmitt trigger
IC9—74LS139 dual 2/4 decoder
IC10—74LS367 hex Tri-State bus driver
IC11—LM386 audio amplifier
IC12—7805 5-volt positive regulator
IC13—7905 5-volt negative regulator
D1-D3—1N4001
T1—9 VAC, 600 mA, wall-plug transformer
S1-S3—SPST slide or toggle switch
S4-S6—SPST N.O. pushbutton switch

Miscellaneous: PC board, speaker, IC sockets, heat sink for +5-volt regulator, enclosure, wire, solder, etc.

The following are available from ELEXOR, PO Box 246, Morris Plains, NJ 07950: double-sided plated-through PC board, $12.50; IC4, $7.50; IC6 and IC7, $25.00; kit of all parts (less enclosure) $69.50. Please add $2.50 for postage and handling as well as applicable state and local sales tax(es).

TALKING ALARM CLOCK

continued from page 60

The second way of feeding data to the VSP is through its microprocessor interface. The interface consists of a bidirectional data bus with some control lines. In addition to being used to carry speech data, that bus is also used to send commands to the VSP. Those commands control all the VSP functions.

During speech generation, the microprocessor first determines which words to speak. If the speech data for a particular word is in the dedicated speech-memory ROM, the microprocessor sends a command to the VSP to address that word, and then sends another to start speaking the word. The entire data fetching and speech-generation process is handled automatically by the VSP. The microprocessor simply commands the VSP to select a certain word and commands it to start speaking. If the speech data is not in the speech ROM, but in the PROM, then the microprocessor sends a command to the VSP instructing it to start accepting speech data via the microprocessor interface. The microprocessor then sends the coded speech-data to the VSP, and the VSP speaks the word. As the VSP generates speech, the microprocessor constantly reads its status to determine when it has finished a word. It then commands the speech processor to speak another one, thus producing phrases made up of several words.

When we continue next time, we'll describe the clock hardware as well as the software that is needed to drive it. And of course we'll cover completely construction, checkout, and use. **R-E**