Designing with Speech Processing Chips

Designing with Speech Processing Chips

Ricardo Jiménez

WESTNET Telecommunications
Calexico, California

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich Publishers
San Diego New York Boston London
Sydney Tokyo Toronto

This book is printed on acid-free paper. ∞

ACADEMIC PRESS, INC.
San Diego, California 92101

United Kingdom Edition published by
Academic Press Limited
24–28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data

Jiménez, Ricardo.
Designing with speech processing chips / Ricardo Jiménez.
p. cm.
ISBN 0-12-385348-6
1. Integrated circuits--Design and construction--Data processing.
2. Speech processing systems. 3. Computer-aided design. I. Title.
II. Title: Speech processing chips.
TK7874.J56 1991
621.39'9--dc20 90-49691
CIP

PRINTED IN THE UNITED STATES OF AMERICA
91 92 93 9 8 7 6 5 4 3 2 1

Preface

The speech processing chip is a relatively new and complex device that appeared in the late 1970s. This book provides the theory and the basic design tools needed to utilize speech processing chips more effectively in electronic circuit design. It presents design examples for a wide range of real-world applications and information on interconnection of these components into functional equipment for instrumentation, data processing, inventory display, and control systems. Special emphasis is placed on those circuits with the most potential for future development, LSI and VLSI devices. Popular commercially available products are used throughout as illustrative examples, and the important characteristics of the devices are summarized for each functional category.

The book goes far beyond the presentation of block diagrams, microcontroller architecture, and software programming. It shows a step-by-step development of hardware and software, how to combine them most effectively, and how to interface speech processors with input devices, such as sensors or data sources, and output devices, such as relay actuators, thyristors, or display devices.

In this book, practicing design and system engineers, technicians, engineering students, and other interested readers will find a comprehensive overview of the entire topic of speech processing chips. The book also describes popular, commercially available circuits for each functional category presented and discusses specific applications in sufficient depth to interest the experienced designer. Engineering students will be able to follow the book if they have been exposed to courses on circuit design theory, logical circuits, and integrated circuits.

As with my other publications, I wanted this to be a book that was organized and used from a practical viewpoint. Throughout the book emphasis is placed on using the popular CMOS and HCMOS ICs of each functional category as illustrative examples.

Speech processing chips for electronic and other applications are available at low cost. The progress made in the manufacture and supply of these chips expands enormously the opportunities to design and build highly effective equipment and systems. This book shows how to make use of these developments. The reader is shown step-by-step how to build both simple and sophisticated projects with all the necessary details. Many proven examples are included throughout the book for industrial, laboratory, health care, and home use.

The information needed to follow the many design examples in this book is given in a simple, direct manner with supporting flowcharts and tables. Once this know-how is acquired, you will be able to build systems with artificial voice with less effort and less time than ever before.

The book is divided into seven chapters. Chapter 1 introduces the different speech processing techniques, describes how the basic speech processor integrated circuit works, and presents IC pin comparisons of the different packages. It also includes the basic datasheet for the device.

Chapter 2 explains how a speech processor can be used in applications for which it was not originally intended—how this basically digital device can be used as a variety of different logic devices. Chapter 3 provides help understanding analog-to-digital converter families and their respective advantages and limitations. After reading these three chapters, you should come away with a fundamental understanding of what a speech processor is and how to use it.

To write information into the specific controller and then announce it, interface devices or systems are required. Chapter 4 shows how such interface devices can be built with minimal effort and at a cost that is often a small fraction of the price of commercial products.

Chapter 5 presents a wide selection of test and measurement circuits that also can be interfaced with a specific application by the user or designer. These circuits are widely used in data acquisition systems. Chapter 6 presents different kinds of burglar alarms varying from simple designs to fault-tolerant systems where failures are critical and not acceptable.

Chapter 7 covers voice recognition techniques as well as devices now available. Some applications for control systems are also considered.

Acknowledgments

I owe thanks to several people who helped me complete this book.

I would like to thank all the publishers who gave permission to include material that originally appeared elsewhere. Footnotes to selected articles cite these original sources.

I am grateful to those instructors from other institutions who reviewed the manuscript for Academic Press, Inc., and who made many constructive suggestions.

Thanks also to my research students of the courses "Logical Circuits II," "Sequential Systems," and "Instrumentation" who suffered through the building and testing process of many of the circuits presented here and in particular to Ruben Rios, Francisco Meza, Jose J. Lara, Ramiro Jacinto, Ruben Avalos, Francisco J. Mendoza, Alvaro E. Salgado, Julio Garcia, Rigoberto Hernandez, and Fabian Muro.

Thanks in particular goes to the Dean of the Technological Institute of Mexicali, Ramon A. Heredia.

I am also grateful to Marcos Silva, Roberto Dewar, and Dr. Fidel Diaz for proofreading the manuscript to make it more understandable. Of course, the patience and love of my wife Patricia was required throughout.

Ricardo Jiménez, E. E.

Contents

Preface **xi**
Acknowledgments **xiii**

CHAPTER 1

Speech Processing Chips

1.1 Introduction to Voice Synthesis Digital ICs **1**
1.2 Synthesis Techniques **3**
1.3 SPO256B/SPO264 Narrator Speech Synthesis Processors **5**
1.4 SPO256-AL2 Allophone Speech Synthesis Processor **11**
1.5 Serial Speech ROM SPR128A **11**
1.6 SPR000 Parallel-to-Serial Speech Interface Chip **15**
1.7 SDS264 Speech Synthesis Development Board **16**
1.8 Digitalker Kit DT1050 **17**
1.9 Toshiba CMOS Speech Synthesis LSI Devices **22**
1.10 TSP522OC Voice Synthesis Chip from Texas Instruments **22**
1.11 CMOS ADPCM Speech Synthesizers and Recorders from Oki Semiconductor **26**
1.12 Samsung Voice Synthesizers ICs **29**
References **31**

CHAPTER 2
Experimenting with Speech Processors

2.1 Evaluating the Allophone Addresses for the SPO256-AL2 **32**
2.2 Interfacing the SPO256-AL2 with a PC/XT **37**
2.3 Interfacing a Speech Synthesizer to a Commodore 64 Computer **38**
2.4 Generating a Technical Vocabulary for the SPO256-AL2 **42**
2.5 Designing a Speech-Synthesized Event Counter **45**
2.6 Circuit Vocalizes Hex Code for a 4-Bit Input **47**
2.7 Circuit Vocalizes 8-Bit Binary Input **49**
2.8 Improved Technique Vocalizes Binary Code for 8-Bit Input **52**
2.9 Microcontroller Routine Handles the Basic Functions of a Speech Processor **60**
2.10 Field Programmable Controller Am29CPL100 Drives the Speech Processor SPO256-AL2 **65**
2.11 Multiplexing a Speech Processor with Different Data Sources **81**
References **84**

CHAPTER 3
Analog Circuits

3.1 Basics of the A/D Converters **85**
3.2 Interfacing the ADC0804 to Digitalker DT1050 **90**
3.3 Handling Multiple A/D Converters **98**
3.4 Using a Window Comparator to Drive a Speech Processor **107**
3.5 Using a 10-Step Voltage Comparator to Drive a Speech Processor **113**
3.6 Interfacing a Speech Processor to a Logic Probe **119**
3.7 Talking Programmable Gain Amplifier **123**
3.8 Designing an Electronic Thermometer That Announces Readings **127**
3.9 Interfacing Displacement Transducers to Speech Processors **135**
3.10 BCD A/D Converter Interface to SPO256 **139**

CHAPTER 4
Digital Circuits

4.1 4-Bit Magnitude Comparator Calls Out the Results **142**

4.2 A Talking Hexadecimal Keyboard Encoder **148**

4.3 Designing a Talking Semaphore **159**

4.4 How to Design a Talking Clock **167**

4.5 Designing a Speaking Coin Detector **174**

4.6 Designing a Talking Coffee Machine Controller **179**

4.7 Designing a Talking Random Number Generator **189**

4.8 Designing an Alternating Current Motor-Speed Controller with Artificial Voice **194**

CHAPTER 5
Test and Measurement Circuits

5.1 Designing a Talking Autorange Frequency Counter **200**

5.2 Designing a Talking Voltmeter Using SPO256-AL2 **216**

5.3 Designing a Direct Current Voltmeter with the Digitalker Kit DT1050 **217**

5.4 Using a Field Programmable Controller to Design a Compact Autorange Direct Current Voltmeter **220**

5.5 Designing a Circuit to Announce Alternating Current Line Voltage **230**

5.6 Designing a Circuit to Announce Alternating Current Line Frequency Cycles **234**

5.7 Using a Speech Processor to Monitor Respiratory Rate **237**

5.8 Designing a Fault-Tolerant Respiratory Rate Meter **245**

5.9 Designing a Soil Moisture Meter **251**

5.10 Designing a CMOS MSI Capacitance Meter **253**

5.11 Designing a Talking Solid State Barometer **263**

5.12 Designing a Talking Darkroom Timer **265**

5.13 Talking Current Meter **275**

5.14 Designing a Liquid-Level Annunciator **281**

CHAPTER 6

Speech-Synthesized Burglar Alarms

6.1 Designing a Burglar Alarm with Artificial Voice **283**

6.2 Designing a Simple Fault-Tolerant Burglar Alarm **290**

6.3 Designing a Vocal Warning Alarm Using CMOS MSI Chips **292**

6.4 Designing a Microcontroller-Operated Burglar Alarm **297**

6.5 Designing an Alarm to Monitor Entrance Devices Independently **303**

CHAPTER 7

Voice Recognition Chips

7.1 Introduction to Voice-Recognition Techniques **308**

7.2 The Word-Recognizer VCP200 **311**

7.3 Voice-Recognizer VCP200 Controls Direct Current Motors **314**

7.4 How to Control Direct Current Motors with a Word Recognizer and a Speech Synthesizer **315**

7.5 Voice-Recognition Synthesis Device SP1000 **321**

References **323**

Index **325**

CHAPTER 1

Speech Processing Chips

1.1 Introduction to Voice Synthesis Digital ICs

Circuits with artificial voice offer a new dimension of sophistication to almost any electrical or electronic modern system. Traditionally, magnetic tape recording has been used in applications requiring speech announcements, for example, telephone announcement systems; a system of this type is costly because it requires a large number of tapes for different messages. It will not let you create mixed messages for different situations. Consider the case of a public service telephone that tells you the time. This device will need 60 different tapes for each hour, not to mention the number of tapes required for a complete 24-hour day. In contrast, a telephone system that uses artificial voice stored in digital memories can create different messages by pulling up the different words required to create a specific message. This system requires only a few chips that will do the work of a large number of tapes.

Let us now consider a talking voltmeter in the range of 0 to 5 V with a resolution of 0.1 V. Here, we will need 50 different messages corresponding to the 50 possible voltage readings. It would be costly and time-consuming to develop a tape mechanism for this project.

In the past, speech systems were treated as data acquisition circuits, in which a voice waveform was treated like any other fluctuating voltage input: The circuit recorded the waveform by periodically taking a sample of the signal's voltage through an analog-to-digital (A/D) converter and storing it as a binary value. (The number of samples needed per second depends upon the frequency of the input signal.) These digital speech signals were stored as pulse code modulation (PCM) in semiconductor memories. Once the samples were stored in RAM or ROM, the circuit could recreate the original waveform by sequentially sending the stored values to a D/A converter at the same rate as the original sampling. One second of digital voice required from 4 to 32 Kbytes of memory. If the amount of data stored was reduced (compressed) using a known principle, the restoration of the original sound was called "synthesis."

The synthesis technique provides a dramatic reduction in the amount of memory required for one second of speech. Memory requirements vary from 400 to 2,000 bits per second, depending on the desired speech attributes and overall quality. A bad reproduction will sound unnatural or unintelligible. A speech signal is highly redundant and predictable, and by coding only the slowly varying coefficients of speech or by dramatic compression of digitized speech, significant bandwidth reductions in the digitized signal can be obtained. The synthesizer technique becomes practical when it is developed with VLSI semiconductor technology.

Today, applications for voice synthesis are endless. The following are some of them: telecommunications; consumer appliances; automotive; counters; consumer products; instrumentation; teaching aids; clocks; language translation; annunciators; voice interactive computer terminals; nautical and aeronautical instrumentation annunciators; voice back units for banking, weather, and time announcements; elevators; trains; subway systems; toys and games; warning systems for fire and police emergency.

In the area of instrumentation, a speech synthesizer is a very important tool. When a failure is presented in the system being monitored, the speech synthesizer will immediately start reading the procedures contained in the manual in order to indicate to the operator how to correct the specific failure. Great benefits are obtained if a speech synthesizer is installed in power stations, nuclear plants, or places where the user must monitor a myriad of controls. Here the speech synthesizer augments the operator's ability to respond rapidly and correctly when a process has extended its normal limits.

In industry, a speech synthesizer can be used to augment productivity by giving spoken messages on how to assemble specific products, thereby freeing a user for other tasks. Here, failure to follow precise directions could lead to the destruction of equipment or injury to personnel.

The use of vocal warnings on automobiles has been spreading since the early 1980s to remind the driver of the electrical or mechanical situation of the vehicle. The same features are also applied to airplanes where the synthetic speech guides the pilot with directions such as "slow down," "climb," or other appropriate instructions.

The pace of the speech processing field is so rapid that some systems now under development are excluded from this book. The emphasis of this book is on designing with the systems now available.

1.2 Synthesis Techniques

The basic phonological element of speech is the phoneme, which is the name given to a group of similar sounds in language. A phoneme is acoustically different depending upon its position within a word. Each of these positional variants is an allophone of the same phoneme. Phoneme reproduction is a basic element in any speech synthesizer. The method of "allophone speech synthesis" is used to create words or phrases where the user has to think in terms of sounds, not letters. With this technique you can synthesize an unlimited vocabulary by using allophones and silences in the appropriate sequence. Phonemes, together with speaker inflection and volume, are the fundamental building blocks of speech.

The American English language consists of approximately 38 to 40 phonemes: 14 to 16 vowel sounds and 24 consonant sounds. For example, the initial K sound used in words like "comb" sounds slightly different from the Ks in words like "can't." These small variations are due to the vowel which follows them, in this case, "o" and "a." Each phoneme is generated with either a voiced sound, as in "eye" or an unvoiced sound like the "sh" in "shy." There are also allophones classified as resonants, voiced fricatives, voiceless fricatives, voiced stops, voiceless stops, affricates, and nasals.

Voice synthesis methods are divided into three major types: waveform encoding, parametric synthesis, and synthesis by rule. Each method is explained below.

Waveform Coding Methods

This type of voice synthesis includes differential pulse code modulation (DPCM), adaptive delta modulation (ADM), and adaptive differential PCM (ADPCM). The original sound wave amplitude is sampled at fixed intervals, digitized, and the volume of data is then reduced on the basis of the synthesis principles.

Parametric Synthesis Methods

Characteristic information included in voice waveforms is extracted as parameters for synthesizing purposes. The partial autocorrelation (PARCOR) method is a typical example. In this method, models of the human vocalization mechanism are used. Voiced and voiceless consonant sounds are discriminated, and voiced sound pitch and amplitude data are extracted together with filter characteristics of the vocal tract. Voice synthesis is then obtained by passing these data to hardware consisting of digital filter circuits.

Synthesis by Rule Method

In this synthesis method, groups of phonemes expressed by small quantities of data are skillfully linked together to reproduce any desired words or phrases.

which makes it easy to develop your own set of words or phrases for your specific application. However, this method lacks the flexibility to create intonation, accents, and length of certain sounds in order to get a natural-sounding voice. This method will be more efficient when vowels and diphthongs with different accents are contained in the allophone set.

There are two general approaches used to derive synthetic speech: (1) time domain synthesis and (2) frequency domain synthesis. The first method works with a synthetic speech waveform representation of the original speech. About half of the synthetic waveform is silence and is made up of symmetric segments which range over a very restricted set of amplitude values. In this form, the synthetic waveform can be stored using only 1% of the bits that are necessary to reproduce the original speech waveform.

Frequency domain synthesis has two main branches: formant synthesis and linear predictive coding (LPC). Formant synthesis generates speech by reproducing the spectral shape of the waveform using the formant center frequencies, bandwidths, and the pitch periods as inputs. A frequency region where the amplitude of a vowel sound is concentrated (i.e., frequency peaks in the voice spectrum) is known as formant. Figure 1.1 shows an electronic speech model of the human speech production mechanism. This model is used in the Signetics speech synthesizer FPC8200.

LPC is based on a mathematical model of the human vocal tract. Pitch, amplitude, and speech variables are obtained from speech recordings. The speech data are analyzed and encoded to reproduce input data suitable for the digital model.

The basic model used in linear predictive analysis is illustrated in Figure 1.2. The two major components are a flat-spectrum excitation source and a spectral shaping filter H(z).

The excitation source provides a signal u(n) containing a flat spectral envelope that is used to drive the filter H(z), resulting in the synthetic speech output signal S(n). Because the excitation signal has a flat spectrum, the spectral envelope of the output signal S(n) will have the same shape as the spectrum of the filter H(z).

In speech synthesis, the parameters of H(z) must be set on a time-varying

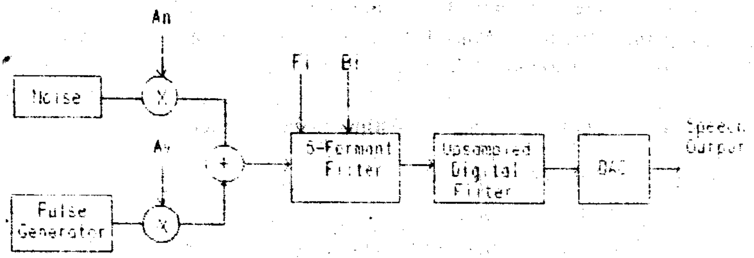

Figure 1.1 Electronic speech model of the human speech reproduction mechanism.

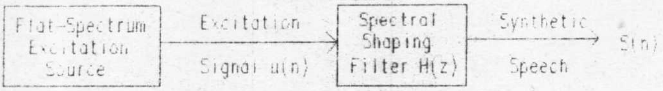

Figure 1.2 Speech synthesis model.

basis such that its short-term spectrum is the same as that of the desired short-term speech spectrum envelope.

1.3 SPO256B/SPO264 Narrator Speech Synthesis Processors

The SPO256B/SPO264 speech processors use the method of synthesis by rule. They also support LPC synthesis, formant synthesis, and allophone synthesis. Phoneme synthesis works by combining basic sound elements (phonemes) to form complete words and sentences. This method is suitable when the type of vocabulary required is not fixed.

Microchip (2355 W. Chandler Blvd., Chandler, AZ 85224-6199), the only manufacturer of SPO256B/SP0264 speech processors, uses the approach just described, combining phoneme synthesis with a digital filter. The speech processors from Microchip employ formant coding, which is a frequency domain synthesis that is similar to LPC. These devices model speech as the output of a series of cascaded resonators.

The speech process is initiated by addressing the ROM address that contains the phoneme desired. A maximum of 256 phonemes can be stored in the 16 Kbits of internal ROM. This device contains a microcontroller and a vocal-tract model. The vocal-tract model is a digital filter. These processors are classified as narrators because they use a preprogrammed custom vocabulary, or phrase set, which is recorded in a serial ROM SPR128 or SPR128B. One of these memories is directly interfaced with the speech processor (SP). If a parallel ROM is desired for recording the phrase set, a parallel-to-serial interface is required. This is made by using the SPR000, as we will see in Section 1.6.

The SPO256B contains an on-chip controller with 16 Kbit internal ROM, while the SPO264 has a 64 Kbit memory. Both devices support a controller for external ROM SPR128A, which is a 128 Kbit ROM. These processors incorporate four basic functions:

- A software programmable digital filter that can be made to model a vocal tract.
- A 16K ROM which stores both data and instructions.
- A microcontroller which controls the data flow from the ROM to the digital filter.
- A pulse width modulator that creates a digital output which is routed to an external low-pass filter in order to get an analog signal.

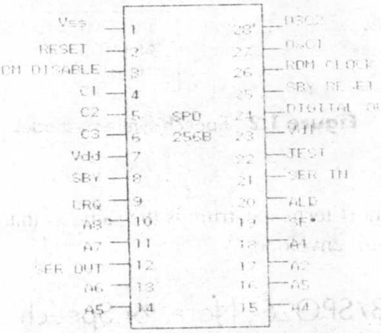

Figure 1.3 Pin configuration of the SPO256B/SPO264 speech processors.

When amplified, this analog output signal will drive a loud speaker. The digital output is equivalent to a flat frequency response varying from 0 to 5 kHz, a dynamic range of 42 dB, and a signal-to-noise ratio of approximately 35 dB.

A nice feature of these synthesizers is the natural speech, and the external ROM directly expandable to a total of 480 K. A version of the SPO256B with internal preprogrammed ROM is the SPO256-AL2.

Figure 1.3 shows the pin configuration of the SPO256B, which is identical to the SPO264.

Operation[1]

The addressing of the SPO256B is controlled by the address pins (A1–A8), address load (ALD), and strobe enable (SE). Strobe enable controls the two modes available for loading an address into the chip.

Mode 0 (SE = 0): The SP latches an address when any one (or more) address pin makes a low-to-high transition. All address lines must be returned to zero prior to entering a new address. In this mode of operation, address zero (0000 0000) is not a valid address input. This mode of operation is used in applications consisting of no more than eight words or messages such that single address line transitions can be made. These words or messages must be stored using the binary address format 1, 2, 4, 8, 16, . . . , 128.

Mode 1 (SE = 1): The SPO256B will latch an address specified on the address bus (A1–A8) when the /ALD pin is pulsed low. Any address varying from 0 to 255 can be loaded using this mode. Specific setup and hold times are required using this mode.

In order to interface the SP with microprocessors (μP) or microcontrollers (μC), two interface pins are available. They are load request (/LRQ) and

[1] This section and the following section, *Test Modes*, are adapted from Publication #DS5018A-2, p. 3. © 1988 Microchip Technology, Inc.

standby (SBY). /LRQ indicates when the input buffer is full. SBY tells the specific processor that the chip has stopped talking and no new address has been loaded. When /LRQ is low, a new address may be loaded onto the address bus. Pulsing /ALD low will cause the new address to be loaded and /LRQ to go high. When the address bus is available to accept a new address, /LRQ goes low again.

The SPO256B can load a new address while it is speaking the last word or message.

Standby (/SBY) goes low when an address is loaded and stays low while the chip is talking. SBY can be used to determine the time between address load requests, which will be variable depending on the length of the word or message currently being spoken.

Several pins are designated for use with an external ROM. They are ROM Disable, C1, C2, C3, Serial Out (SER OUT), Serial In (SER IN), and Test. ROM Disable is tied to a logic zero to enable the external ROM; when left floating, it disables the external ROM. C1 to C3 are the output control lines which are synchronous to the ROM clock. ROM CLOCK (pin 26) is a 1.56 MHz clock output used to drive an external serial speech ROM. The operating frequency of the SPO256B is set by an external 3.12 MHz crystal connected to OSC1 and OSC2 (pins 27 and 28), respectively. You can also use 3.27 to 3.14 MHz crystals; in this case the voice pitch will be slightly altered.

When C1C2C3 = 000, no operation is executed. When C1C2C3 = 001, the address shift register load (ASRL) serially shifts out data from the SER OUT pin as shown in Figure 1.6. The ASRLD loads 16 bits of the ASR with two 8-bit load sequences followed shortly by a program counter load (PCLD).

When C1C2C3 = 010, the contents of the address shift register are loaded into the program counter (PC) when 16 ASR loads have occurred.

When C1C2C3 = 011, the data shift register loads the 8-bit data shift register with the contents of the ROM pointed to by the current address in the program counter. The data shift register will shift out the LSB of the 8 bits and increment the program counter. With C1C2C3 = 100, data is shifted out of the data shift register starting with the second LSB (the first LSB is shifted out with the occurrence of C1C2C3 = 011). Seven shifts occur after every DSRLD.

When C1C2C3 = 101, the stack is loaded with the current value of the PC. With C1C2C3 = 110, the PC is loaded with the contents of the stack to perform the RETURN operation. Finally, C1C2C3 = 111 will occur when /SBY RESET and /RESET are pulsed low.

Test Modes

By using the TEST logically anded with the address inputs A1, A2, A3, or A5, the SPO256B can be interfaced to an SPRO000 (serial-to-parallel ROM) in order to use an EPROM as the speech data device.

The test modes are controlled by the TEST pin (22). This is achieved by