

(12)

EUROPEAN PATENT APPLICATION

(21) Application number: **81305149.7**

(51) Int. Cl.³: **G 10 L 1/08**

(22) Date of filing: **29.10.81**

(30) Priority: **03.11.80 US 203042**

(43) Date of publication of application:
12.05.82 Bulletin 82/19

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **GENERAL INSTRUMENT CORPORATION**
600 West John Street
Hicksville New York 11802(US)

(72) Inventor: **McLaughlin, Philip T.**
46 Clinton Street
Farmingdale New York 11735(US)

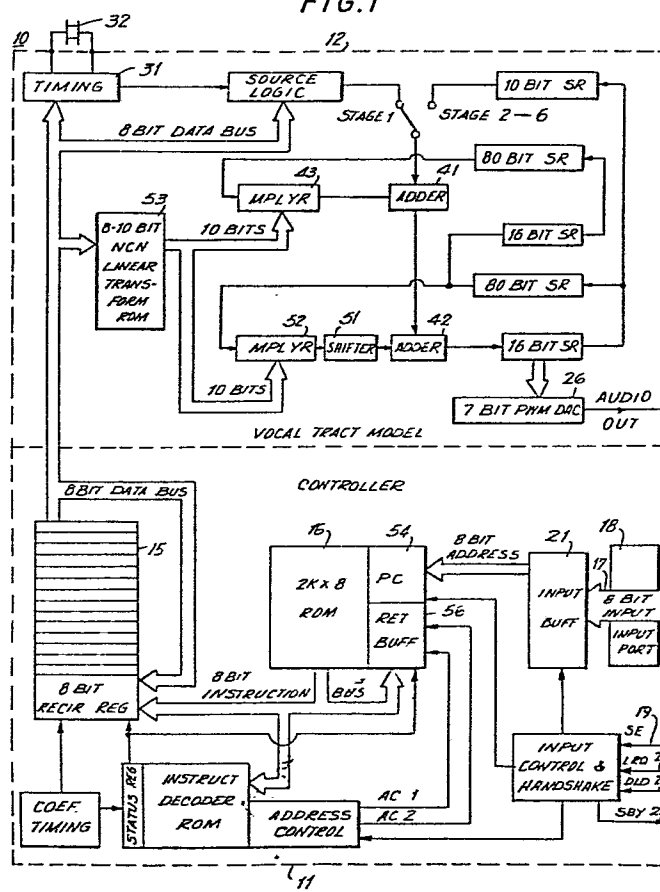
(74) Representative: **Warren, Francis Charles et al,**
Baron & Warren 18 South End Kensington
London W8 5BU(GB)

(54) **Speech processor.**

(57) A system and apparatus for generating human speech or other complex sounds from a digital command signal and using a hybrid synthesis technique comprises a digital, fixed repertoire speech processor (10) which includes a first circuit (12) for modeling the behaviour of the human vocal tract and a second circuit (11) for controlling the behaviour of the modeling circuit (12).

The processor may advantageously be implemented on a single VLSI integrated circuit chip.

FIG. 1



- 1 -

SPEECH PROCESSOR

Broadly speaking, this invention relates to the synthesis of human speech. More particularly in a preferred embodiment, this invention relates to methods and apparatus for synthesizing human speech using a hybrid synthesis technique.

In recent years considerable interest has been expressed in the synthesis of human speech, and other sounds. Heretofore, this has required a large computer and expensive speech processors connected to, and driven by, the computer.

More recently, advances in large scale integration have permitted the speech synthesis circuitry to be reduced in scale so that it can be accommodated on a few VLSI integrated circuit chips, or in the case of the instant invention, on a single VLSI chip.

As is well known, there are four basic techniques for synthesizing human speech. These are (1) phoneme synthesis, (2) formant synthesis, (3) linear predictive coding, (LPC), and (4) wave-form digitization with compression (WD).

Three of the goals of any speech synthesis are (1) understandability, (2) quality, and (3) price or cost which may be defined as the bit requirement for each second of speech produced.

The four basic techniques for synthesizing human speech can be compared with the three goals. In phoneme synthesis the output voice is clearly under-

standable. The quality of the voice is robotic. There is one voice and it is not identifiable as other than that of an artificial source. The bit requirement is fairly low with 120 bits for each second of speech.

5 In formant synthesis the understandability is good. The quality is better than in phoneme synthesis and it is capable of producing voices which are distinguishable between male and female. The bit requirement is 400-800 bits per second. In linear predictive

10 coding the understandability is the same as in formant. The quality can be very high and an individual person's voice can be easily recognized but this requires more bits for more quality. Typically between 1,200 and 3,000 bits are required for each second of speech. Wave-form

15 digitization with compression makes a very broad range of all three goals. The understandability can be very good to very poor. The quality also extends over the same broad range. This reflects the cost or the number of bits required which varies from approximately 1,000

20 to 5,000 bits for each second of speech, the best quality and understandability being with more bits.

The present invention provides an inexpensive and very flexible speech synthesizer and is capable of providing high understandability, a range of qualities from

25 acceptable to highest quality, and a flexible bit rate which is adjustable in the chip from 500 bits to 3,000 bits per each second of speech.

In one embodiment the entire synthesizer can be constructed on a single chip. Prior circuits required

30 multiple chips. This has an important result in cost,

as the single chip synthesizer significantly reduces the cost over multiple chip synthesizers.

5 An advantage of the present invention is that it uses formant synthesis for voiced (vowel) sounds and LPC for unvoiced sound. Formant and LPC coding can be used in the same word. It is thus a reduction in the size of the memory and bits needed to produce the same sound.

10 A further advantage of the present invention is that more memories can be dedicated to a particular sound or group of sounds, thereby permitting the ability to increase the quality or to adjust the quality of the sound of the synthesizer.

15 The present invention is a technique for synthesizing human speech that is relatively inexpensive, overcomes the deficiencies of the prior art, and is suitable for fabrication on a single VSLI chip.

SUMMARY OF THE INVENTION

20 As a solution to the above and other problems, the

instant invention comprises a digital, fixed repertoire,
processor for generating human speech in response to a
sequence of n-bit, digital, command words input thereto.
The processor comprises a means for electronically modelling
5 the behavior of the human vocal tract and means, connected
to the source of the incoming digital command words, for
controlling the operation of the modelling means thereby
to control the speech generated by an analog signal
generating means associated with the vocal tract modelling
10 means.

The invention and its mode of operation will now
be described in detail, with particular reference to
the appended drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

15 Fig. 1 is a block schematic diagram of an illustrat-
ive speech processor according to the invention;

Fig. 2 is a block schematic diagram of an illustrat-
ive digital filter stage for use in the speech processor
shown in Fig. 1; and

20 Fig. 3 is a block schematic diagram illustrating how
a plurality of filter stages according to Fig. 2 may be
connected in tandem.

DETAILED DESCRIPTION OF THE INVENTION

The speech processor disclosed and claimed herein is
25 intended for product applications where the generation of
synthetic speech or complex sounds is required. In a pre-
ferred embodiment, the speech processor is realized as
an N-channel, metal-gate LSI device. One skilled in the
art will realize, however, that other implementations are
30 possible.

- 5 -

The speech processor according to the invention is a fixed repertoire speech and sound synthesizer, which, in the preferred embodiment, is capable of reproducing up to 256 discrete sound sequences. Each sequence may be called
5 by loading the 8-bit address of the sequence into a command register in the speech processor. The sound sequence data is stored in a mask-programmable read-only memory (ROM), which arrangement enables the user to readily specify the speech or sound pattern desired.
10 By use of suitable interfaces additional ROMs may be added and that would essentially provide an unlimited number of words.

As will be explained, the internal organization of the processor enables a large quantity of speech or
15 sound to be specified in 16K bits of read-only memory. In addition, the flexible architecture of an on-board controller associated with the processor allows the user to partition the available storage space into as many sequences as desired, up to a maximum of 256 sequences.

20 From a functional standpoint, and referring to Fig. 1, processor 10 can be divided into two major sections, a controller 11, and a vocal tract model (VTM) 12. VTM 12 is a parametric sound and voice synthesizer which produces complex waveforms under the control of 15 slowly time
25 varying parameters. As will also be explained, the controller 11 executes internal instructions, which are stored in ROM, and modifies the appropriate parameters of VTM 12 to create the desired sound sequence.

In the preferred embodiment, the interface between
30 controller 11 and VTM 12 is accomplished through a plurality of parameter registers and related timing signals.

Referring again to Fig. 1, we shall now present an overview of the processor's architecture. Let us first

consider the VTM source timing. Briefly, the VTM operates under control of 15 parameter registers, the size and function of which are listed in Table A. The duration and pitch of the sounds produced by the processors are controlled by the R and P registers, respectively. The contents of the P register, in particular, specifies the number of sample periods in each pitch period.

Expressed in terms of the pitch (F_0) and the sampling frequency (F_s):

10
$$P = \frac{F_s}{F_0} \quad 1 \leq P \leq 255 \text{ ----- (1)}$$

The pitch source injects unit impulses, spaced P sample periods apart. The contents of the R register (repeat count) represents the number of pitch cycles which will be executed before a register update occurs.

15 When P is assigned a zero value, the pitch source is replaced by a zero-mean, pseudo-random noise source. This mode of operation is referred to as the unvoiced mode. In this mode, the processor requests a register update after 64 x R samples. The amplitude of the source is controlled by the A register. It is coded as 5 bits of
20 mantissa and 3 bits of exponent (i.e. binary shift).

- 7 -

TABLE AVTM PARAMETER REGISTERS

<u>Register</u>	<u>Size</u>	<u>Function</u>
R	6 bits	Repeat count (Number of Pitch Periods)
P	8 bits	Pitch period (Number of Sampling Periods)
A	8 bits	Source Amplitude
5 F1	8 bits	F Coefficient of Filter Stage 1
F2	8 bits	F Coefficient of Filter Stage 2
F3	8 bits	F Coefficient of Filter Stage 3
F4	8 bits	F Coefficient of Filter Stage 4
F5	8 bits	F Coefficient of Filter Stage 5
10 F6	8 bits	F Coefficient of Filter Stage 6
B1	8 bits	B Coefficient of Filter Stage 1
B2	8 bits	B Coefficient of Filter Stage 2
B3	8 bits	B Coefficient of Filter Stage 3
B4	8 bits	B Coefficient of Filter Stage 4
15 B5	8 bits	B Coefficient of Filter Stage 5
B6	8 bits	B Coefficient of Filter Stage 6
Ai	8 bits	Amplitude Interpolation
Ti	8 bits	Pitch Interpolation

Let us now consider controller 11 in detail. As shown, controller 11 is a sequential processor which fetches instructions and data from an internal 16K ROM 16, and which is capable of altering the contents of the 15 parameter registers 15 controlling the processor's vocal tract model. The controller has 16 executable instructions, and supports one level of subroutine nesting. In the present embodiment, the instruction set is specifically designed to allow selective updates of the parameter registers to be performed. In addition, the instructions designated JMP and JSR allow chaining of segments, and sharing of code sequences to eliminate redundancy.

The processors instruction set comprises two groups of instructions, i.e. register modification instructions, and branch control instructions.

We will now discuss how the processor is interfaced with external devices. First, let us consider program entry control. In speech processor 10, the individual sound sequences stored in internal ROM 16 are accessed by use of an 8-bit data bus 17 and input port 18. The significance of this input byte is a function of the state of an SE (Strobe Enable) conductor 19. When the SE line 19 is high and input port 18 has been loaded from the external system via bus 17, the contents of the 8-bit input port is loaded into an 8-bit input buffer 21. This allows any one of 256 entry points to be specified. The entry points are spaced at ^{2, 4 or} 8 byte increments (i.e. ^{for 8 byte:--} 0, 8, 16, 24 ... 2040) throughout ROM 16. This addressing scheme is most useful in applications where the processor is used as a peripheral for an external microprocessor, or

or where more than 8 sequences are programmed into the ROM.

In applications where 8 or fewer sequences are required and it is desirable to select these sequences without
5 handshaking, the processor can be placed into a mode where each speech or sound sequence can be initiated by pulling down a single conductor, for example, by grounding the SE conductor 19. When the processor is operated in this way, no handshaking is required to select the de-
10 sired sequence.

Let us now discuss handshake control. As previously mentioned, input to the processor is accomplished via the 8-bit bus 17, 2 handshake lines 22 and 23, and the input mode select conductor 19. As mentioned in the previous
15 section, the handshake lines are not necessary for some applications. When SE (Strobe Enable) conductor 19 is kept high, the handshake lines 22 and 23 (LRQ and DLD) are used to coordinate the data input. LRQ (Load Request) is on output line 22 is low whenever the Input Buffer Flag
20 is set. When LRQ is high, the input port is loaded by placing the 8 bits of data on input buffer 17 and pulsing the DLD (Data Load) input 23. The rising edge of DLD will cause LRQ to go low, where it will remain until the internal

Input Buffer Flag is reset by an RET instruction.

The output of VTM section 12 drives an internal 7-bit pulse-width-modulation, digital-to-analog converter 26. The design of DAC 26 is such that all noise components
5 are at or above 10 KHZ. The output is low pass filtered to 5KHZ, and amplified, both of these functions being performed externally.

In the preferred, LSI embodiment, the processor has two power supply leads and a common ground. One supply
10 lead powers the interface logic and provides standby current to the controller and parameter registers. The other lead powers VTM 12, controller 11 and internal ROM 16. When standby lead 27 is high (indicating that the processor is inactive, the second power lead can
15 be powered down, externally, to conserve power. This will provide a standby current which is a fraction of the normal operating current.

When the processor is loaded with an entry byte, the standby lead 27 is brought low, signalling to the exter-
20 nal circuitry to power-up the second power lead. The processor will delay execution of the selected sequence to allow the power supply to settle. This is done by an RC time circuit external to the chip but driven by the chip. If it is not desired to implement the stand-
25 by mode of operation, the power leads can be tied to a common supply.

The processor requires one 3.12MHZ clock, which is generated by an onboard oscillator 31 with external crystal control. The crystal 32 is connected to oscillator 31
30 external of the processor.

Let us now consider operation of the processor and the filter structure of the VTM in particular. The processor according to the invention models speech (and other sounds) using a series of six 2nd order resonators, excited by either a pseudo-random noise source, or a periodic impulse source.

VTM 12 is implemented using totally digital techniques. This approach allows one 2nd order section to serve as six sections through the use of multiplexing and information line pipelining. The section that is implemented is the 2nd order infinite impulse response (IIR) digital filter shown in Figure 2. This filter comprises a pair of adder stages 41 and 42 and three multipliers 43, 44 and 46. The filter stage has the transfer function:-

$$H(Z) = \frac{1}{1 - 2F_t Z^{-1} - B_t Z^{-2}} \quad (2)$$

Therefore, it can be shown that the poles of the transfer function occur at:

$$\frac{-2F_t \pm \sqrt{4F_t^2 + 4B_t}}{-2} \quad (3)$$

and when,

$$-1 < B_t \leq 0 \quad (4)$$

and,

$$|F_t| \leq \sqrt{-B_t} \quad (5)$$

the poles will be placed in a complex pair, forming a resonator with the band-width given by:

$$B.W. = \frac{-F_s \ln(B_t^2)}{T} \quad (6)$$

where F_s is the sampling frequency in HZ.

and the center frequency (F_k) given by:

$$F_k = \frac{F_s \cos^{-1} \left[\frac{2F_t}{\sqrt{-B_t}} \right]}{2T} \quad (7)$$

As can be seen from equations 6 and 7 above, the modification of the B coefficient changes both the frequency and the bandwidth of the resonator. The modification of the F coefficient, however, changes only the center frequency and has no effect on the corresponding bandwidth.

Since speech signals, and in particular vowel sounds, convey information through the shifting of resonant peaks in the spectrum, it is desirable to be able to change center frequencies of the 2nd order stages independently of their respective bandwidth settings. In addition, it is important that the parameters of the individual stages, corresponding to particular resonances, can be modified independently. The use of cascade 2nd order stages supports these features, giving the filter configuration according to the invention a distinct advantage over other filter sections currently in use for speech synthesis, such as the Lattice section, and the direct form implementation. The instruction set of the processors controller section is designed to exploit the ability of the VTM parameters to be updated, selectively, to achieve a greater packing density in ROM 16. In addition, this permits the user to trade-off between the quality and quantity of the speech samples possible within the ROM space available.

If it is desired to place resonances at a frequency of zero, these real axis poles can be accommodated directly. Each 2nd order stage may be used to place two real axis poles of variable bandwidth. If X_1 is the real axis location of the first pole, and X_2 the second:

$$F_t = \frac{X_1 + X_2}{2} \quad (8)$$

and:

$$B_t = (F - X_1)^2 - F^2 \quad (9)$$

with the bandwidths of each given by:

$$5 \quad BW_1 = -2Fs \ln X_1 \quad (10)$$

$$BW_2 = -2Fs \ln X_2 \quad (11)$$

where F_t and B_t represent the coefficients in Figure 2.

Coefficient updates to the filter occur at the beginning of a pitch period. This timing results in the smallest possible disturbance to the output at update. The information line precision is maintained at 16 bits throughout the VIM filter. The multiply-by-2 unit 44 shown in Fig. 2 is advantageously implemented as a 1-bit binary shift circuit 51 following the F_t multiplier. The shift operation is performed separately from the multiplication to scale F_t to the same range of values as B_t . The coefficients F_t and B_t are quantized, non-linearly, to minimize coefficient sensitivity.

The two coefficients are processed by the same non-linear transformation hardware in the range.

$$-1 < C < 1$$

where C may be either F_t or B_t .

The non-linear transformation $T(X)$ is implemented with a table-look-up ROM 53.

The input coefficients of each stage, denoted as F and B , are expressed in sign magnitude form and used to generate the multiplier coefficients as follows:

$$F_t = T(F) \quad (12)$$

$$B_t = T(B) \quad (13)$$

Fig. 3 represents vocal tract model 12 and depicts six cascaded filter stages 61-66. For unvoiced sounds (U), the input to the filter comes from a pseudo-random noise source 68 while for voiced sounds a scaled,
5 periodic impulse source 67 is used.

We will now consider the programmability of the processor including the register modification instructions.

The purpose of the processor register modification instructions, of course, is to update the VTM parameters.
10 The R and P registers determine how many sample periods of a particular sound are output by the VTM, before control of the parameter registers is returned to the controller. The controller waits until the completion of the last of R pitch periods, or $64 \times R$ samples in unvoiced
15 mode, before executing the next register modification instructions.

Each of the 13 register modification instructions, with the exception of the RCU instruction, comprise a
4-bit op code followed by 4 bits of data which are
20 loaded into the lower 4 bits of register R. The instruction RCU is a 1-byte instruction which loads the upper 2 bits of the Repeat register, i.e. Register R.

The RCU instruction passes control to the next instruction following execution. The RCU instruction
25 will not cause an immediate transfer of control to the VTM.

	MNEMONIC	INSTRUCTION CODE NNNN = LOWER 4 BITS OF R	No of BYTES	REGISTERS MODIFIED
	AMP	0 0 0 0 N ₃ N ₂ N ₁ N ₀	2	R,A
	FFL	0 0 0 1 N ₃ N ₂ N ₁ N ₀	4	R,F1,F2,F3
5	FFU	0 0 1 0 N ₃ N ₂ N ₁ N ₀	4	R,F4,F5,F6
	FS1	0 0 1 1 N ₃ N ₂ N ₁ N ₀	3	R,B1,F1
	FS2	0 1 0 0 N ₃ N ₂ N ₁ N ₀	3	R,B2,F2
	FS3	0 1 0 1 N ₃ N ₂ N ₁ N ₀	3	R,B3,F3
	FS4	0 1 1 0 N ₃ N ₂ N ₁ N ₀	3	R,B4,F4
10	FS5	0 1 1 1 N ₃ N ₂ N ₁ N ₀	3	R,B5,F5
	FS6	1 0 0 0 N ₃ N ₂ N ₁ N ₀	3	R,B6,F6
	PCT	1 0 0 1 N ₃ N ₂ N ₁ N ₀	2	R,P
	FRL	1 0 1 0 N ₃ N ₂ N ₁ N ₀	15	R,B1,F1,A,B2, F2,P,B3,F3,B4, F4,B5,F5,B6,F6
15	RCU	1 0 1 1 0 0 N ₅ N ₄	1	R (N ₅ ,N ₄ are upper 2 bits of A)
	PRL	1 1 0 0 N ₃ N ₂ N ₁ N ₀	9	R,B1,F1,A,B2, F2,P,B3,F3
20	JSR	1 1 0 1 0 A ₁₀ A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀	2	PC→RB,A ₀ thru A ₁₀ →PC Return Flag Set
	RET	1 1 1 0 0 0 0 0		If Return Flag Set RB→PC, Return Flag reset
25				If Return Flag Not set IB→PC 0→IB IBF Reset
30	JMP	1 1 1 1 0 A ₁₀ A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀	2	A ₀ thru A ₁₀ →PC

In the event that more than one register modification instruction is required to update the parameter registers, but it is not desired to perform a full update (FRL), two or more instructions may be chained
5 together. This feature is also useful for initializing the registers before a JMP, JSR, or return instruction, without causing the VTM to initiate a sound sequence.

If an instruction is to be chained with one or more other instructions, the lower 4 bits of the instruction
10 byte are set to zero. The last instruction in a series of chained instruction bytes is the only one with non-zero lower 4 bits. The lower 4 bits of the last instruction byte in a chained series is loaded into the lower 4 bits of the R register. When chaining instructions,
15 the data bytes appear after the last instruction. The order of the data bytes is given by Table C, below.

For example, it is desired to chain an FFU instruction with an FSl instruction, the following registers would be updated:

20 FFU modifies F4, F5, F6

FSl modifies B1, F1

Referring to Table C, it will be seen that the proper sequence for the above data is:

B1, F1, F4, F5, F6

25 therefore, referring to the processor Instruction Table B, the sequence would be

DATA BYTE LOAD SEQUENCE FOR CHAINED INSTRUCTIONS

	Loaded First	1	B1
		2	F1
		3	A
		4	B2
5		5	F2
		6	P
		7	B3
		8	F3
		9	B4
10		10	F4
		11	B5
		12	F5
		13	B6
	Loaded Last	14	F6

15 Coded as:

0 0 1 0 0 0 0 0
 0 0 1 1 N₃ N₂ N₁ N₀

[B1]

[F1]

20 [F4]

[F5]

[F6]

25 where N₃ N₂ N₁ N₀ represents the lower 4 bits
 stored in register R, and [] represents the
 binary value to be loaded into the indicated
 register.

The processor's Branch Control instructions differ from the register modification instructions in that they do not modify any of the VTM parameter registers. The sole purpose of the branch control instructions is to
5 determine the location in internal ROM 16 from which the next instruction will be fetched. The JSR (Jump to Subroutine) instruction stores the present address (i.e. the contents of the program counter register 54) in the return buffer (RB) register 56. The program
10 counter 54 is loaded with the 11-bit address specified by the last 3 bits of the instruction byte and the following data byte. In addition, an internal return flag is set to indicate that the RB register 56 has been loaded. The controller then fetches and executes
15 the instruction located at PC + 1 in the ROM 16. Only one level of subroutine is allowed.

The Jump instruction JMP loads the program counter 54 with the 11-bit address specified by the lower 3 bits of the instruction byte, and the following data
20 byte. Neither the return flag nor the return buffer are modified. Upon completion of execution, the next instruction is fetched from location PC + 1 in ROM 16.

The RET (Return from Subroutine) instruction is an instruction whose function depends on the state of the
25 return flag. When the return flag is set, indicating that a sub-routine is being executed, execution of an RET instruction will cause the contents of the RB register 56 to be moved into the Program Counter 54. The return flag is reset, and the controller fetches
30 the instruction located at PC + 1 and continues exe-

cution from that location. Again, only one level of
subroutine is allowed. When an RET instruction is
encountered and the return flag is not set, the status
of the input buffer flag (IBF) is checked. If the IBF
5 flag is set, indicating that the starting address of
the next sound sequence has been loaded into the speech
processor, the contents of input buffer register 21
(8 bits) is loaded into program counter 54, left
justified. If input buffer flag is not set, the con-
10 troller will disable any further output from the VTM and
wait for the input buffer flag to become set. The
standby conductor 27 will go high and remain high until
the input buffer flag is set. The standby conductor
was previously discussed in the section on standby
15 operation. When the input buffer flag is set, execut-
ion continues as described above.

The previously discussed preferred embodiment, i.e.
the N-channel, metal-gate LSI device, has the following
characteristics:

20 Maximum Ratings

Power Supply	- .3 to + 12 V.
Storage Temperature	-25° to + 125° C
Lead Temperature (Soldering)10 SEC	+ 330° C

Typical Operating Conditions

25 Power supply	+4.6V to +7.0V.
Operating temperature	0° C to + 55° C

Clock

Crystal Frequency	3.12MHZ (Nominal)
-------------------	-------------------

D.C. CHARACTERISTICS

<u>INPUTS:</u>		<u>MIN</u>	<u>TYP</u>	<u>MAX</u>	<u>UNITS</u>	<u>CONDITIONS</u>
DATA						
DLD, SE, RESET						
5	LOGIC 0	0.0		.6	V	
	LOGIC 1	2.4		VD1	V	
	CAPACITANCE			10	uF	
	LEAKAGE			10	uA	
<u>OUTPUTS:</u>						
10	SBY, DAC					
	(OPEN DRAIN OUTPUTS)					
	LOGIC 0			0.6	V	@ 1ma
	LEAKAGE			10	uA	
	LRQ					
15	(2 LS TTL LOADS)					
	LOGIC 0	0.0		0.6	V	.72ma
	LOGIC 1	3.5		VD1	V	-50uA
<u>CURRENT:</u>						
	VD1			15	ma	VD1, VD2 =
	VD2			45	ma	7.0V
						VSS = 0.0V
20	Standby current (VD1)			15	ma	VD1 = 7.0V
	Oscillator: XTAL IN			10	uA	No Loads
	XTAL OUT					
	LOGIC 0	0.0		0.8	V	.36ma
	LOGIC 1	3.5		VD1	V	-50uA

To expedite testing of the speech processor, the contents of the 2K x 8 ROM 16 can be read-out in serial format. In addition, the processor can be placed in a test mode where instructions and data are input from the 8-bit input port 18 in place of the ROM 16. The information line, the data path in the VTM, is output on the serial output (SER).

Since digital-to-analog converter 26 is a PWM design, the DAC output can be tested as an ordinary digital output. No special level detection is required.

The test program for a processor according to the invention advantageously comprises two sections, a fixed part which tests the functionality and parameters of the processor, excluding ROM 16, and a ROM test section which is unique to the pattern being tested.

The speech processor according to the invention has a totally digital architecture which is designed to operate in the TTL voltage range. The drive requirements, operating voltages, speed requirements, are all compatible with implementation in standard N-Channel Metal Gate LS1 technology.

One skilled in the art can make various changes and substitutions without departing from the spirit and scope of the invention.

CLAIMS

1. A digital, fixed-repertoire, processor for generating human speech and other complex sounds in response to a sequence of n-bit digital command words input thereto from an external source, characterised in that it comprises:
- 5 (a) means (12) for electronically modeling the behaviour of the human vocal tract and other complex sound sources, said modeling means including:
- (1) an m-stage, second-order, infinite impulse response, digital filter (41,42,43,44,46);
- 10 (2) means for exciting the input of said m-stage digital filter; and
- (3) means (26), connected to the output of said digital filter, for generating an analog signal, corresponding to said human speech and other
- 15 complex sound sources; and
- (b) means (11) connected to the source of said incoming digital command words, for controlling the operation of said modeling means thereby to continue the speech generated by said analog signal generating means,
- 20 said controlling means including;
- (4) a plurality of n-bit, recirculating registers (15), connected to said modeling means (12), temporarily storing the parameters which define the sound segment phoneme currently being
- 25 generated by said modeling means;
- (5) a first, digital, read-only memory (16) for storing the parameters which define all possible sounds and sound segments in said repertoire, and
- 30 (6) means (19), responsive to said digital command word, for addressing a particular one of the parameters stored in said read-only memory (16).

2. A processor according to claim 1, characterised in that said modeling means further includes:
a second read-only memory (53) connected to said plurality of n-bit recirculating registers (15) and
5 to said m-stage digital filter, for storing parameters to control the response characteristic of said digital filter

3. A processor according to claim 1, characterised in that said m-stage digital filter comprises:
10 a second-order, infinite impulse response, digital filter section; and
means for multiplexing the digital information applied to said filter section, thereby to cause said section to appear functionally as m cascaded filter sections.

15 4. A processor according to claim 3, characterised in that said filter section includes:
means for altering the center frequency of the filter bandpass independently of the bandwidth.

5. A processor according to claim 3, characterised in
20 that said filter section comprises:
first and second tandem-connected adders (41,42), each having first and second inputs and an output;
first, second and third multipliers (43,44,46), each having first and second inputs and an output; and
25 first and second tandem-connected impedance networks, the input to said filter section being connected to the first input of said first adder (41), the output of filter section being connected to the output of said second adder (42), a first response-controlling
30 parameter signal being connected to the first input of said first multiplier and a second response-controlling parameter signal being connected to a first input of said third multiplier.

6. A processor according to claim 5, characterised in
35 that said second multiplier (44) is a multiply-by-two

multiplier and is interposed between the output of said third multiplier (46) and the second input of said second adder (42).

5 7. A processor according to claim 5, characterised in that said first read-only memory (16) stores parameters defining the amplitude, pitch-period and the number of pitch periods to be requested for said modeling means for each phoneme.

10 8. A processor according to claim 7, characterised in that said first read-only memory (16) further stores parameters defining the first and second response-controlling parameters for each of the m-stages in said digital filter.

15 9. A processor according to claim 1, characterised in that said controller (11) further includes:
means for decoding operation codes stored in said first read only memory; and
means responsive to said decoded operation codes, for altering the parameters stored in said plurality
20 of n-bit recirculating registers (15).

10. A processor according to claim 1, characterised in that said analog signal generating means comprises:
an (n-1) bit pulse width modulated digital-to-analog
connector (26), the output of said filter being
25 connected to an external low-pass filter and an audio amplifier.

11. A processor according to claim 1, characterised in that it further comprises:
a source of pseudo-random noise (68);
30 a source of scaled periodic impulse signals (67); and
means, responsive to the instantaneous command word for selectively connecting the input of said digital filter to the output of either said noise source or said impulse source.

35 12. A system for generating human speech or other complex

- 25 -

sounds from a digital command signal characterised in that voice phonem signals are generated by a formant synthesis circuit, and unvoiced phonem signals are generated by a linear predictive coding circuit: control means responsive to said digital command signal select the phonems signal from said circuits as ordered by said command signal and combine the generated voiced and unvoiced phonems to form words of speech or the other complex sounds, and in that said entire system is on a single semiconductor integrated circuit chip.

FIG. 1

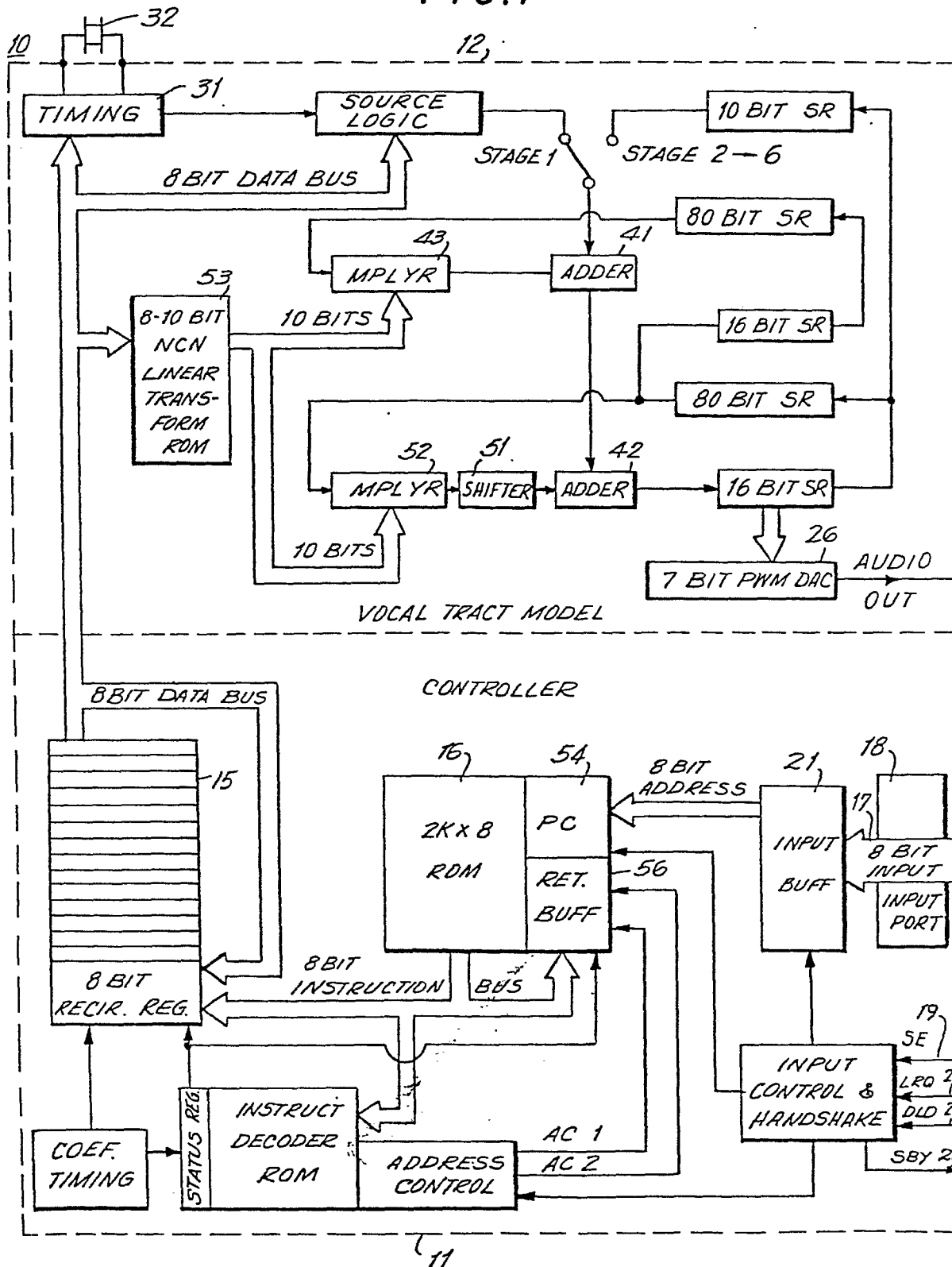


FIG. 2

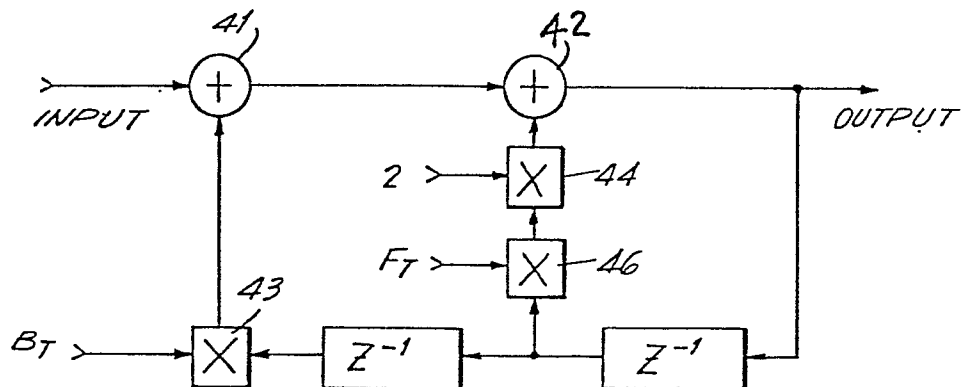


FIG. 3

