324-77. SR
12-19-72 XR 3,706,929

United States Patent

Robinson et al.

[15] **3,706,929**

[45] **Dec. 19, 1972**

[54] **COMBINED MODEM AND VOCODER PIPELINE PROCESSOR**

[72] Inventors: **John L. Robinson,** Wenonah, N.J.; **Robert F. Munnich,** Willow Grove, Pa.

[73] Assignee: **Philco-Ford Corporation,** Philadelphia, Pa.

[22] Filed: **Jan. 4, 1971**

[21] Appl. No.: **103,503**

[52] **U.S. Cl.**......................**325/15,** 178/67, 179/1 SA, 324/77 R, 325/30, 325/38 R, 340/148

[51] **Int. Cl.**...**H04b 1/00**

[58] **Field of Search**........179/1 SA, 15.55; 324/77 R, 324/77 A, 77 B; 325/30, 38 B, 38 R, 15; 178/67; 340/148, 152

[56] **References Cited**

UNITED STATES PATENTS

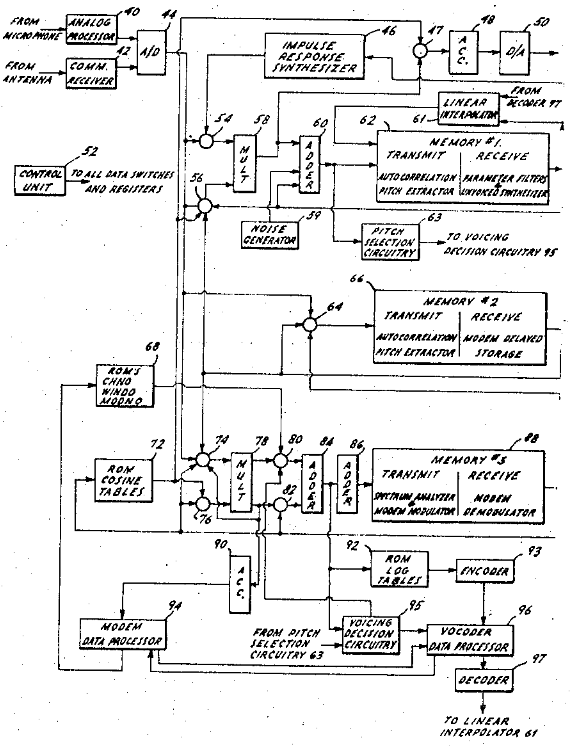
3,403,227	9/1968	Malm	179/1 SA
3,360,610	12/1967	Flanagan	179/1 SA

Primary Examiner—Albert J. Mayer
Attorney—Herbert Epstein

[57] **ABSTRACT**

A digital pipeline processing system for implementing both vocoder and data modem functions. The pipeline capability is provided by three circulating memories and two associated arithmetic units. The first circulating memory and its arithmetic unit implement the functions of vocoder spectrum analysis modem modulation and modem demodulation. The second circulating memory and its arithmetic unit implement the functions of vocoder pitch extraction, vocoder parameter filtering, and vocoder speech synthesis. The third circulating memory is used for temporary storage of data while computations are carried out by the other circulating memories and their arithmetic units. The processing system also comprises a control unit which provides timing and gating signals for control of the remainder of the processor, an impulse response synthesizer which provides sinusoids used in speech synthesis, encoding and decoding circuitry for formatting data, and a plurality of read-only memories for permanent storage of functions required by the processor.

4 Claims, 13 Drawing Figures



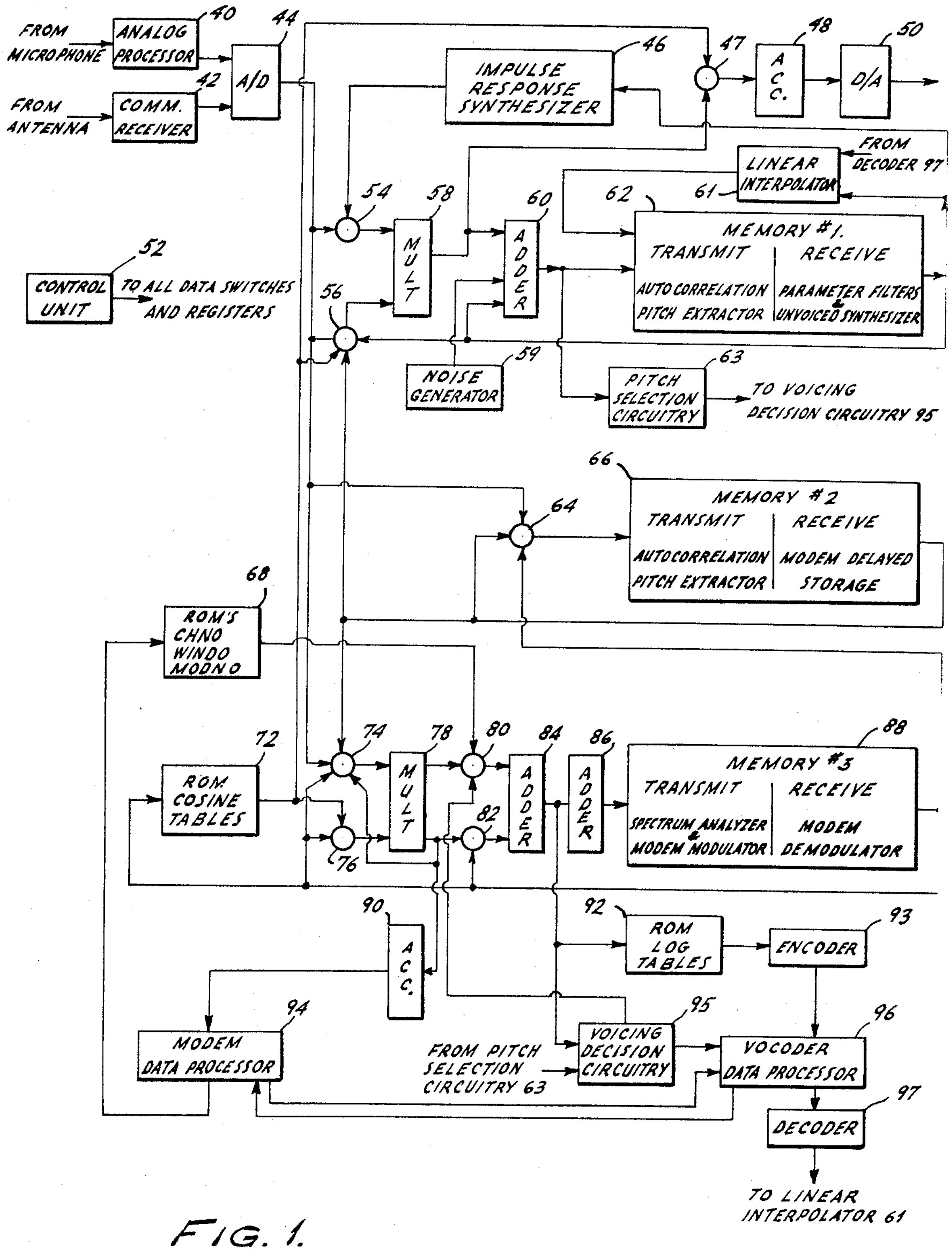


FIG. 1.

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

PATENTED DEC 19 1972

3,706,929

SHEET 2 OF 8

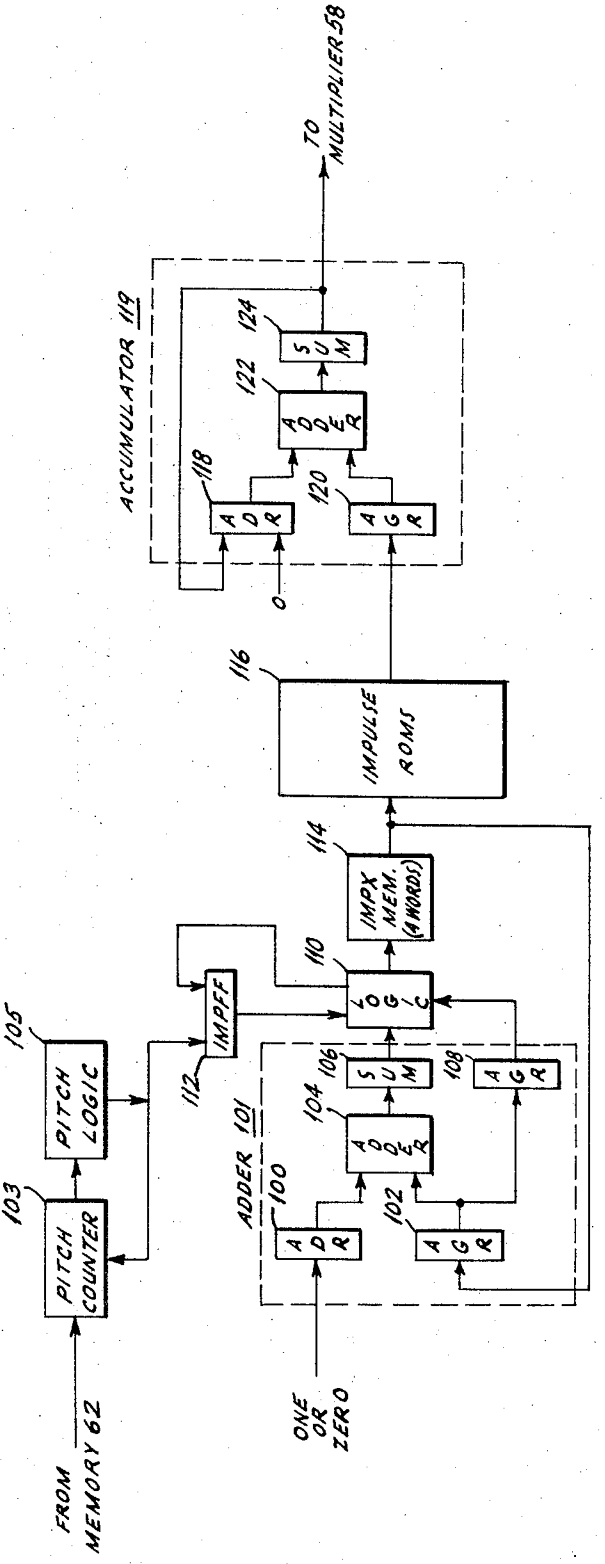


FIG. 2. IMPULSE RESPONSE SYNTHESIZER 46

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

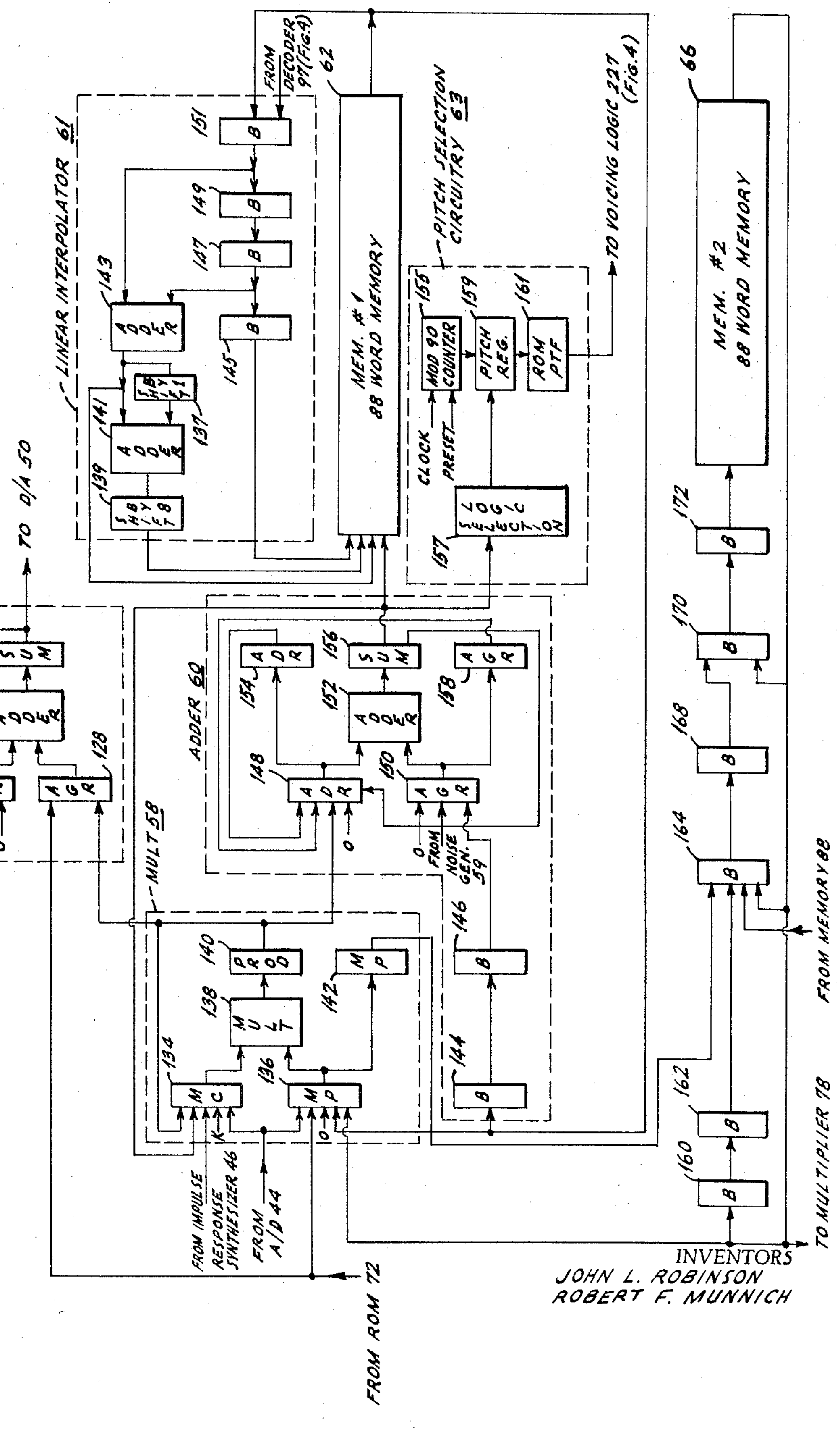


FIG. 3.

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

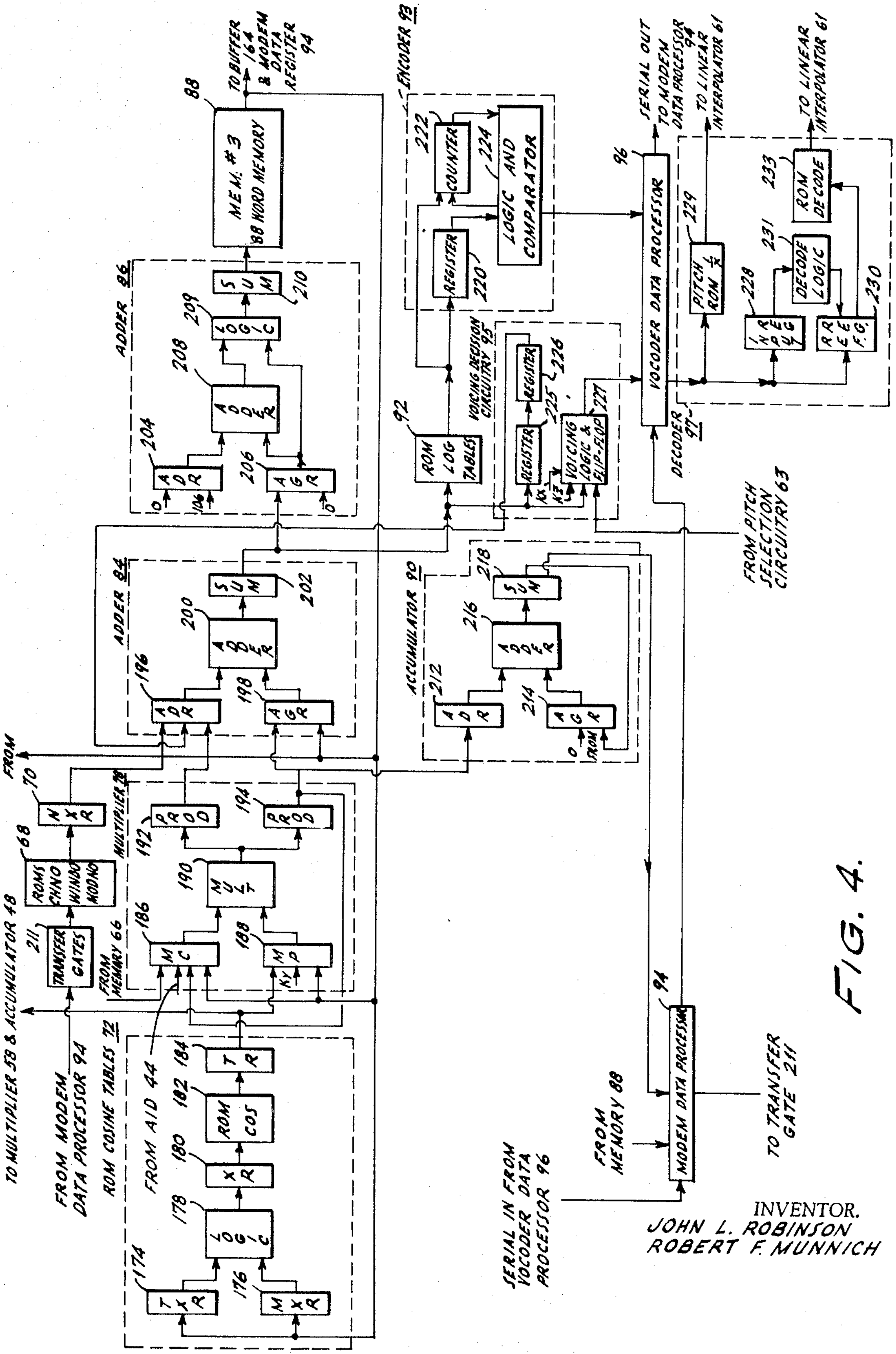


FIG. 4.

INVENTOR.
JOHN L. ROBINSON
ROBERT F. MUNNICH

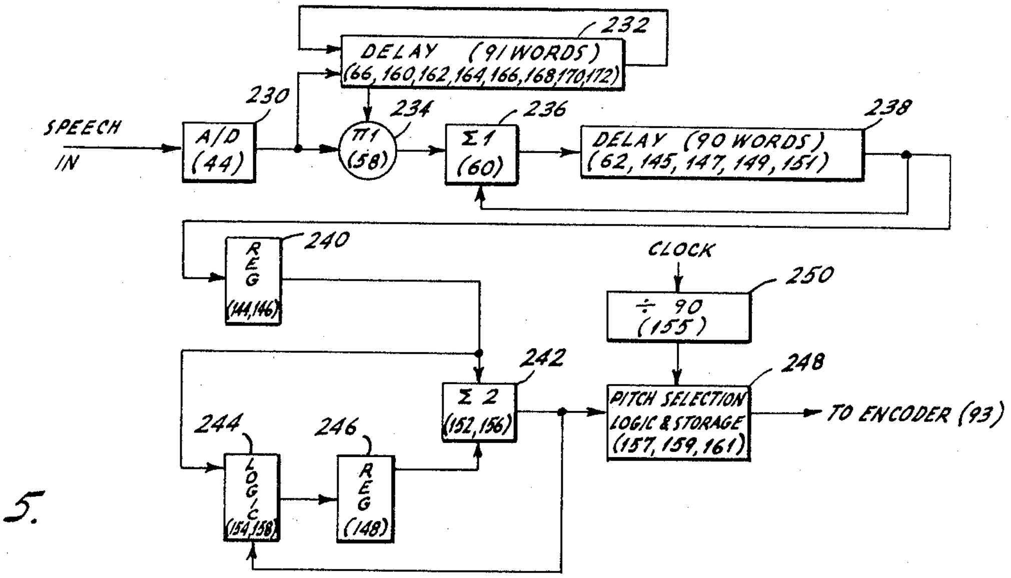


FIG. 5.

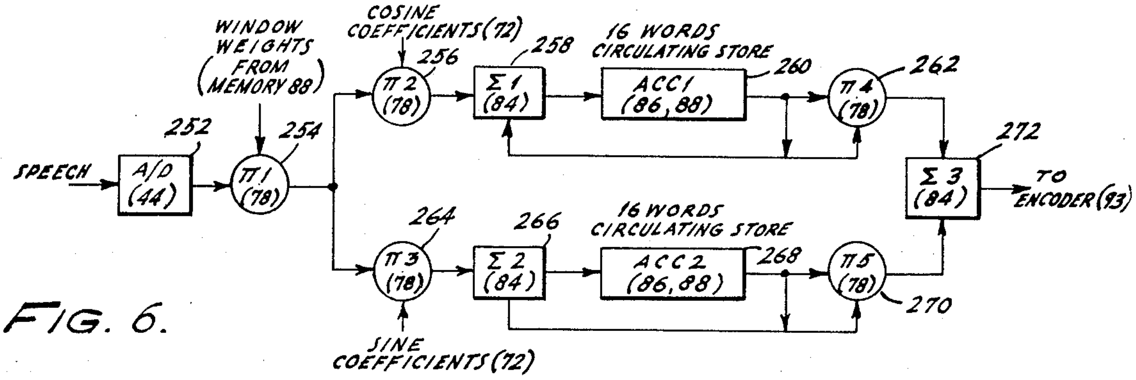


FIG. 6.

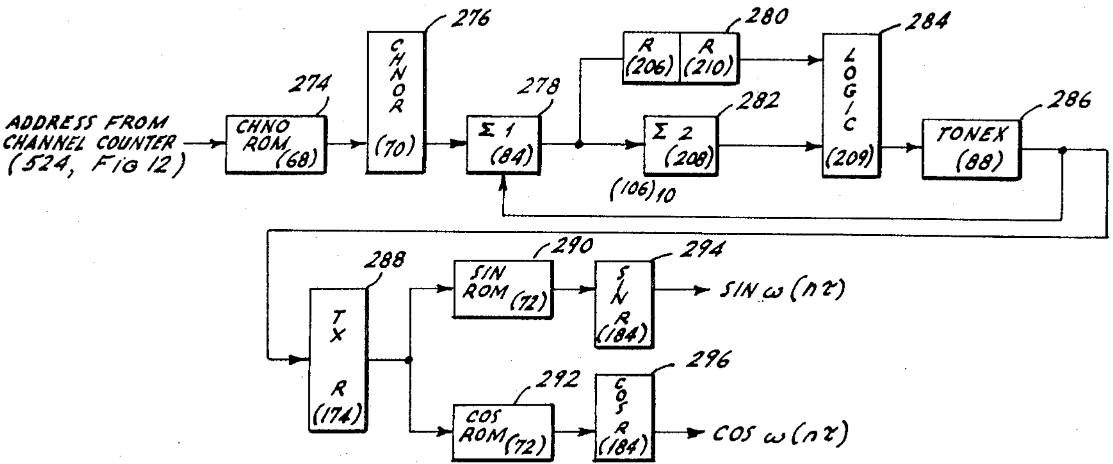


FIG. 7.

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

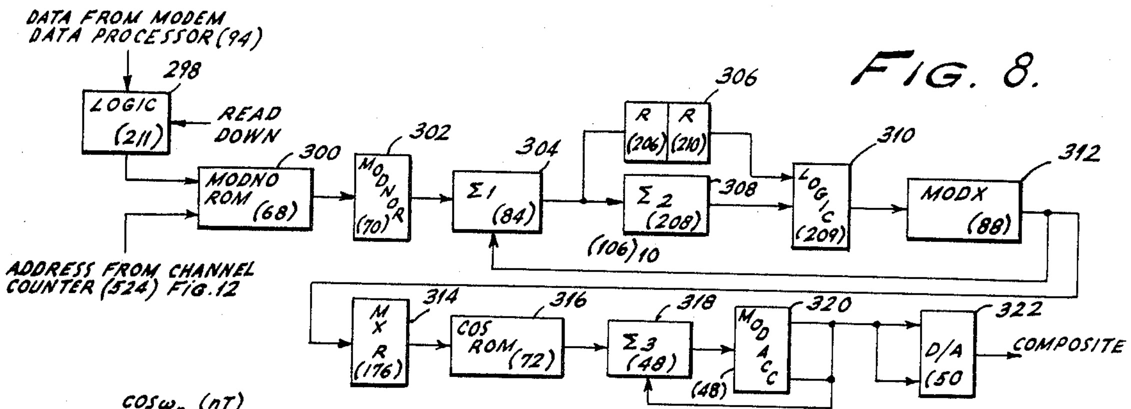


FIG. 8.

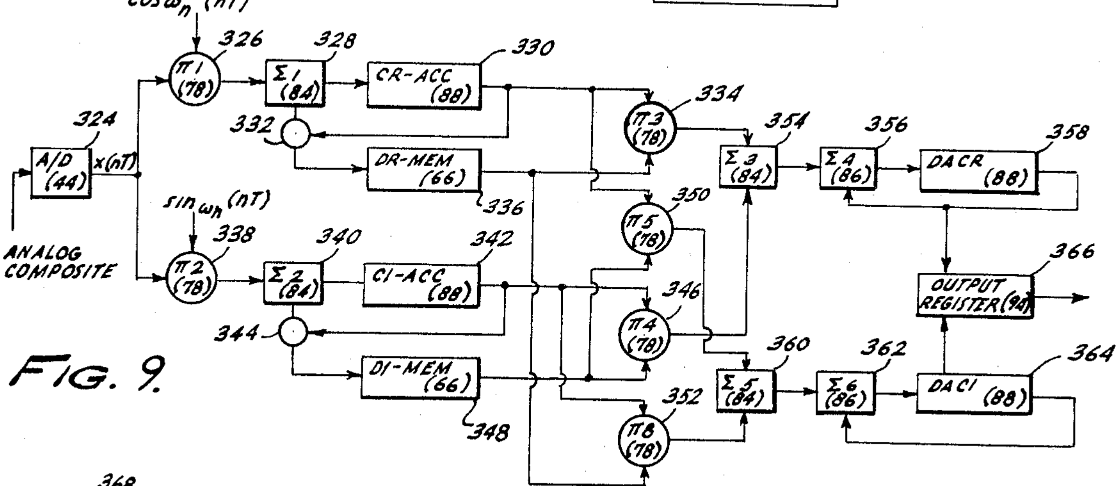


FIG. 9.

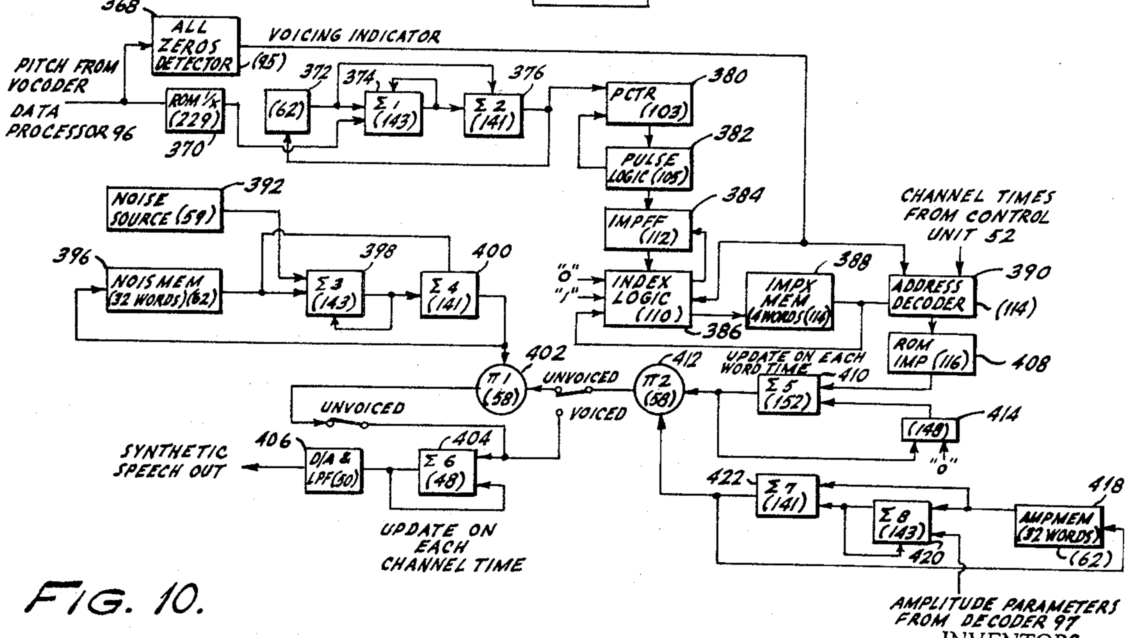


FIG. 10.

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

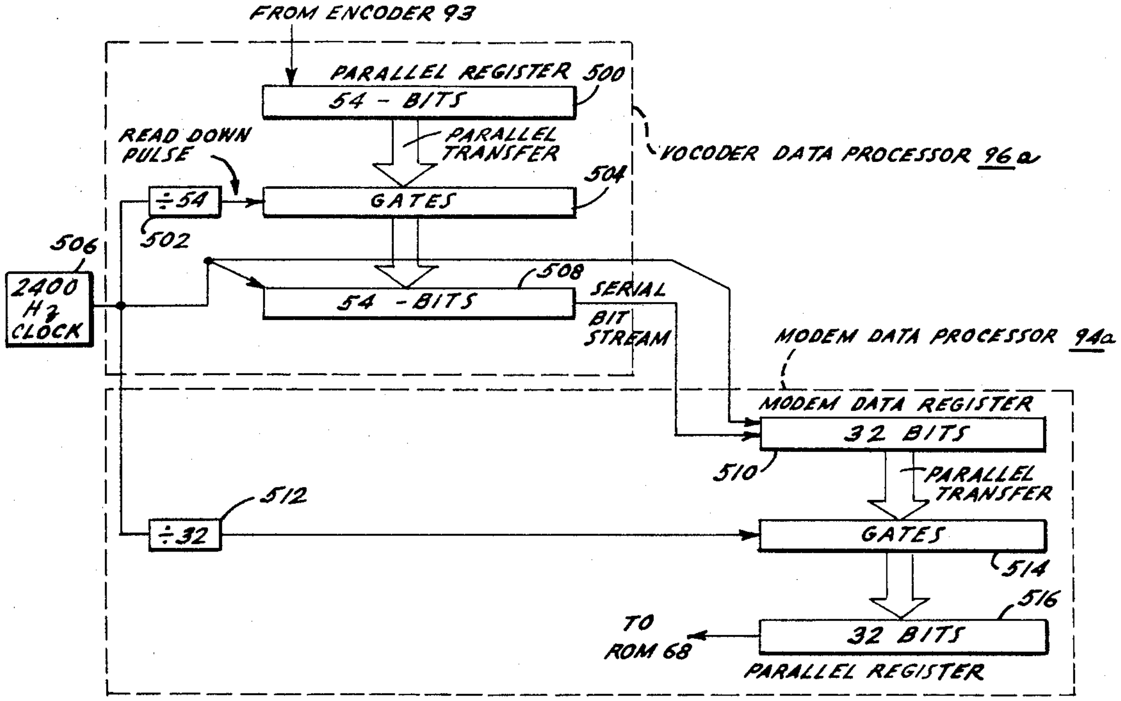


FIG. 11a. MODEM AND VOCODER DATA PROCESSORS

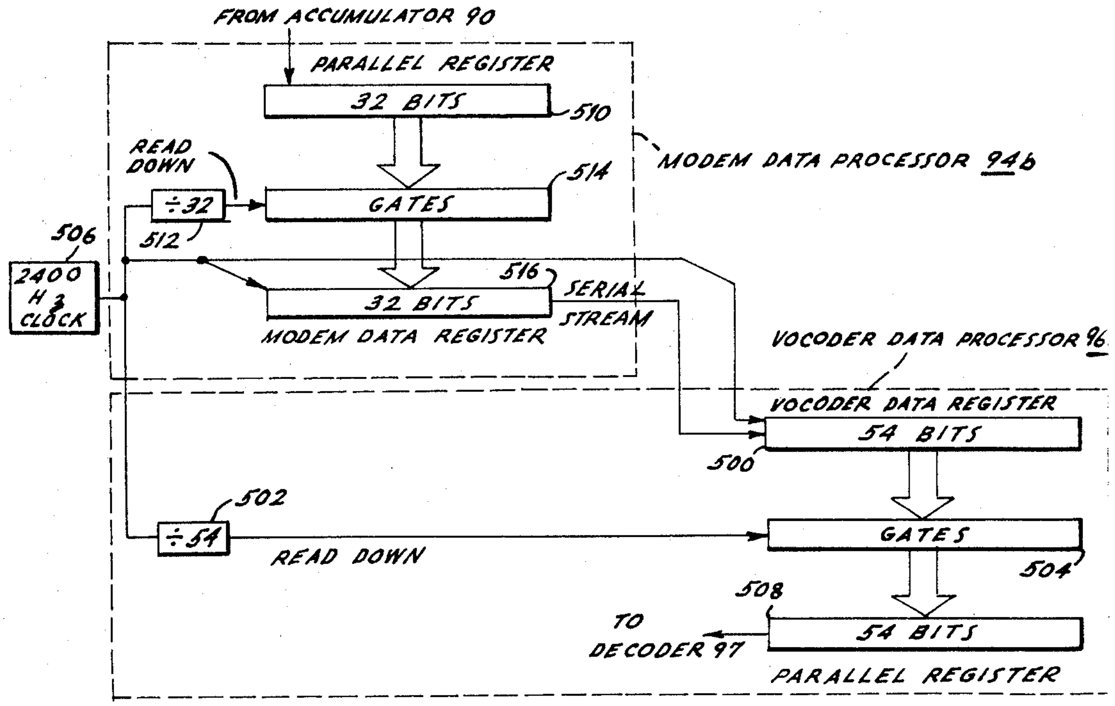


FIG. 11b. MODEM AND VOCODER DATA PROCESSORS

INVENTORS
JOHN L. ROBINSON
ROBERT F. MUNNICH

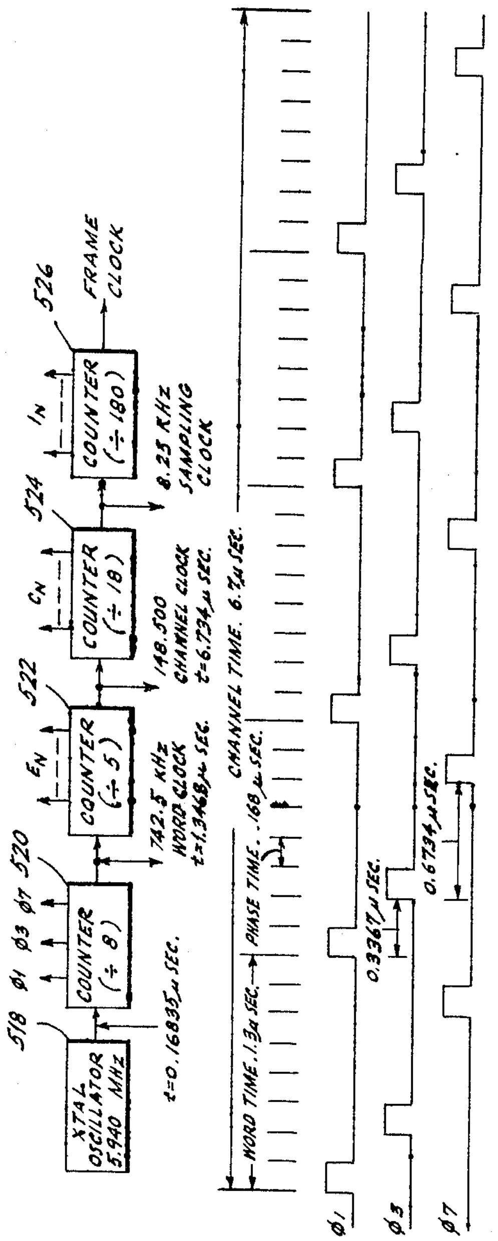


FIG. 12.

INVENTOR.
JOHN L. ROBINSON
ROBERT F. MUNNICH

COMBINED MODEM AND VOCODER PIPELINE PROCESSOR

BACKGROUND OF THE INVENTION

The invention herein described was made in the course of or under a contract or subcontract thereunder, with the U.S. Army Electronics Command, Fort Monmouth, N. J.

This invention relates to a pipeline digital processor for implementation of both vocoder and data modem functions.

Vocoder systems function to transmit speech signals in a coded manner to reduce the transmission bandwidth which would otherwise be required if the speech was to be transmitted in an uncoded manner. Thus a vocoder system includes both a "transmit" terminal to analyze the characteristics of the speech wave to be encoded and to encode the speech wave, and a "receive" terminal to synthesize from the coded signal sent to it a reconstruction of the original speech wave.

Data modems function to facilitate the transmission of data (for example from a speech vocoder) over a transmission medium. Thus a modem includes both a "transmit" terminal to convert the encoded data into a modulating signal used to modulate a carrier, and a "receive" terminal to demodulate the received signal and thereby recover the transmitted data.

Both vocoder and modem equipment are therefore required for transmission of speech signals in an efficient high performance system. Prior art systems have provided separate digital hardware to implement the vocoder and the modem functions. A digital processor which can use the same hardware to implement both these functions would result in a substantial savings in equipment. Moreover a substantial decrease in processing time can be achieved by performing these functions in a pipeline processor. The latter processor differs from a conventional digital computer where a complete cycle time to retrieve a number from memory, perform an operation on it and return it to memory is required before the next operation can begin. In the pipeline processor, data is continuously circulated through memory and multiple arithmetic units, i.e., retrieval of a second operand begins before the first result has been returned to memory, and the arithmetic unit starts working on a second set of operands before the results from the first set are returned to memory. Thus by choosing the sequence of mathematical operations or algorithms so that the pipeline can be kept full, processing time is greatly reduced.

A key factor in the construction of the pipeline processor which can implement both the vocoder and the modem functions is the choice of algorithms to be implemented. Only by proper choice of the algorithms which are to be implemented to synthesize the vocoder and modem can the simultaneous objectives of (1) producing digital apparatus which can implement both the vocoder and modem and (2) making possible a sequence of mathematical operations amenable to pipeline processing, be achieved.

Accordingly, an object of this invention is to provide a digital processor which can implement the functions required of both the vocoder and the modem.

Another object is to provide a pipeline digital processor adapted to function as a speech vocoder.

Another object is to provide a pipeline digital processor adapted to function as a data modem.

Another object is to provide a pipeline digital processor adapted to function as a combined vocoder and modem.

SUMMARY OF INVENTION

In accordance with the invention, these objects are achieved by a digital apparatus which functions in two operating modes. In the "transmit" mode the input signal to the apparatus is representative of a speech wave in the time domain and in the "receive" mode the input signal is representative of the spectral density and the pitch frequency of a speech wave. In either mode input signals are applied to an analog to digital converter where they are sampled and converted to binary form, and then applied to the pipeline processor portion of the apparatus. The pipeline processor comprises first and second portions each of which comprises a circulating memory and an arithmetic unit.

When the apparatus is arranged in the "transmit" mode, the first portion of the pipeline processor computes the spectral density of the speech wave and generates a binary representation of a modulated carrier containing the spectral density and the pitch frequency of the speech wave. The pitch frequency is computed in the second portion of the pipeline processor and is supplied to the first portion.

When the apparatus is arranged in the "receive" mode, the first portion of the pipeline processor demodulates the input signal and converts it into binary words representative of the spectral density and the pitch frequency of the speech wave. This information is then supplied to the second portion of the pipeline processor where the time domain representation of the speech wave is synthesized.

DRAWING

FIG. 1 is a simplified block diagram of the invention.

FIG. 2 is a more detailed block diagram of the impulse response synthesizer **46** of FIG. 1.

FIG. 3 is a more detailed block diagram of a first portion of the pipeline processor of FIG. 1.

FIG. 4 is a more detailed block diagram of a second portion of the pipeline processor of FIG. 1.

FIG. 5 is a flow diagram depicting a process for autocorrelation pitch extraction.

FIG. 6 is a flow diagram depicting a process for vocoder spectrum analysis.

FIG. 7 is a flow diagram depicting a process for modem tone synthesis.

FIG. 8 is a flow diagram depicting a process for modem modulation.

FIG. 9 is a flow diagram depicting a process for modem demodulation.

FIG. 10 is a flow diagram depicting a process for vocoder synthesis.

FIGS. 11*a* and *b* are more detailed block diagrams of vocoder and modem data processors **94** and **96**, respectively.

FIG. 12 is a more detailed block diagram of a portion of control unit **52** of FIG. 1.

DESCRIPTION OF THE INVENTION

Before entering upon a detailed description of the invention, the concept upon which the invention is based is described briefly.

In order to achieve a pipeline processor which performs the functions of both a vocoder and a data modem on a real time basis, it is necessary to choose those mathematical algorithms for implementation of these functions which allow use of digital apparatus compatible with both functions. In the preferred embodiment of the invention, implementation by the following algorithms of a channel vocoder and a differentially-coherent phase-shift-keyed frequency division-multiplexed (DCP SK/FDM) modem has been chosen:

Channel Vocoder:

Pitch Extraction: Autocorrelation pitch extraction

Spectrum Analysis: A discrete Fourier transform (DFT) with triangular weighting.

Voiced Synthesis: Impulse response synthesis using table lookup

Unvoiced Synthesis: Heterodyned noise algorithm

High Frequency Modem:

Modulation: Tone synthesis by table lookup

Phase detection: The discrete Fourier transform (DFT)

Differential phase Calculation: Vector multiplication

Diversity Combining: Vector addition

Channel vocoders attempt to reproduce the short time power spectrum of the speech waveform. The conventional channel vocoder comprises a pitch extractor to measure the pitch or fundamental frequency of the speechwave and a bank of filters (or its digital equivalent) to measure the spectral content of the speech wave. Presence or absence of the pitch signal or a test based on the speech energy in the filter bank can be used to indicate the presence of voiced or unvoiced sounds. The signals are then transmitted to the receiver for reconstruction of the speech waveform. Excitation derived from a pitch modulated pulse generator for voiced synthesis, or a broadband noise generator for unvoiced synthesis, is applied to a bank of filters in the receiver identical to that used in the transmitter. These filter outputs are amplitude modulated by the received signals which define the spectral content of the speechwave, and combined to provide a reconstruction of the speechwave.

In the digital processor of the invention, the functions of the channel vocoder are accomplished in the following manner:

Pitch extraction is accomplished by processing the speech signal in accordance with an autocorrelation pitch extractor algorithm wherein the autocorrelation function of the incoming signal is computed, and the pitch period is estimated by measuring the distance between autocorrelation peaks.

Spectrum analysis is accomplished by computing the discrete Fourier transform (DFT) of the incoming speech wave. As is well known, the computation of the DFT involves an integration process. To assure that significant temporal changes are accounted for, the spectrum is computed by analyzing a portion of the incoming signal as seen through a triangular time window or weighting function. This processing is equivalent to analyzing the speech waveform with a bank of 16 analyzing filters.

The voicing decision is made by a conventional energy balancing type of process wherein the energies in different portions of the spectrum are compared to preset thresholds and to each other. If the thresholds are exceeded and the energies in the different portions of the spectrum bear the correct relationship to each other, the decision that voiced signals are present is made.

Voiced synthesis is accomplished by synthesizing the impulse response of each channel by table lookup and then multiplying each channel by its corresponding amplitude parameter which is derived from the spectrum analysis performed on the original speechwave.

Unvoiced synthesis is accomplished by a heterodyne noise algorithm wherein each channel is modulated by a noise-like signal and then processed as in the case of voiced signals.

The algorithms used to implement the modem functions result in 16 frequency division multiplexed (FDM) channels of information carrying data representative of the results of the spectral analysis of the speech wave and the pitch frequency information. The data is carried in each channel by means of four phase differentially coherent phase shift keyed (DCP SK) modulation of the carrier.

Modulation is accomplished by a table lookup algorithm wherein the value of every quantized sample of each tone corresponding to each channel is calculated in advance and stored in permanent storage. The modulated signal bearing the voice information is then synthesized by computing the sequence of addresses required to generate each tone with its proper phase shift, and then retrieving the stored samples. The stored samples are then added to form the modem output.

In the demodulation process, the digital equivalent of supplying a filter bank to separate the transmitted signals is accomplished by computing the discrete Fourier transform (DFT) of the composite of received signals. This DFT algorithm results in a series of complex frequency coefficients representative of the amplitude and phase of each of the tones.

The DCP SK modulation of each tone is demodulated by a vector multiplication algorithm wherein the differential phase vector is computed by calculating the vector product of a complex frequency coefficient and the complex conjugate of the previously received coefficient.

Diversity combining, if required, is accomplished by vector addition, wherein the real and imaginary parts of the differential phase vectors of each of the channels to be combined are summed separately. In the preferred embodiment of the invention, maximal ratio combining is implemented. This technique is described in: D. G. Brennan, "Linear Diversity Combining Techniques," *Proceedings of the IRE*, Vol. 47, No. 6, pages 1075–1101, June 1959.

The pipeline processor of the invention operates half-duplex i.e., in either a transmitting or receiving mode.

In the "transmit" mode the input signal to the processor is a speech wave and the processor performs the vocoder function of speech wave analysis and the modem function of generating a modulating signal which carries the results of the speech wave analysis. This modulating signal becomes the output signal of the processor in this mode and can be used in the modulator of a conventional communication transmitting system.

In the "receive" mode the input signal to the processor is a speech information bearing communication signal such as may be derived from a conventional communications receiver. The same type of modulation used in the transmit mode, must be used in this mode. The processor performs the modem function of demodulation of the input signal and the vocoder function of synthesis of the speech wave. The output signal of the processor in this mode may be used to drive conventional voice reproduction circuitry.

FIG. 1 is a block diagram which shows the general organization of the pipeline processor of the invention.

At the heart of the processor are three circulating memories 62, 66 and 88, respectively designated "-memory No. 1," "memory No. 2" and "memory No. 3." Memory 62 along with its arithmetic unit comprising multiplier 58 and adder 60, and memory 88 together with its arithmetic unit comprising multiplier 78 and adders 84 and 86 function as first and second portions of the pipeline processor, respectively. Memory 66 is used to store data for the pipeline processor and has no arithmetic unit of its own. The functions performed by each portion of the pipeline processor in both the "transmit" and "receive" modes are listed within the blocks representative of memories 62, 66 and 88 and are described in detail hereinafter. In the preferred embodiment of the invention, these memories each have a capacity of 88 words. The word lengths are 12, 12, and 16 bits for memories 62, 66 and 88, respectively.

"Transmit" Mode

In the "transmit" mode, speech signals from a conventional transducer (not shown) may be processed in analog processor 40 which may be representative of a conventional VOGAD (voice operated gain adjustment device) and/or circuitry to provide preemphasis or limit the bandwidth of the transmission path for the speech wave. Analog processor 40 is coupled to analog-to-digital converter (A/D) 44 where the analog speech wave is sampled periodically and a binary representation of each analog sample is generated. In the preferred embodiment of the invention, the sampling rate is 8,250 Hz, and each sample is represented by an eight-digit binary word.

A/D circuit 44 is coupled to multiplier 58, adder 60 and memory 62 via data switches 54 and 56, and to memory 66 via data switch 64, all of which form a first portion of the pipeline processor. The data switches are gating circuits which function under control of control unit 52 to transfer data at the proper time and to the proper circuitry during each mode of operation. The first portion of the pipeline processor performs pitch extraction by computing the autocorrelation function of the incoming speechwave and measuring the period between autocorrelation peaks. Each autocorrelation coefficient, as will be discussed subsequently, is the sum of a number of currently-received speech samples multiplied by an equal number of preceding samples. Memory 66 stores samples of the incoming speech wave. Stored samples are then transferred via data switch 56 to multiplier 58 where they become the preceding samples to be multiplied by current samples of the incoming speech wave. The products are summed in adder 60, and the resulting autocorrelation coefficients are accumulated in memory 62. After all the 90 autocorrelation coefficients have been calculated, they are read out of memory 62 and sequentially passed through adder 60 which functions as a comparator to find the largest value (i.e., the peak of the autocorrelation function) among them. Adder 60 is coupled to pitch selection circuitry 63, where the pitch frequency is determined and assigned a six bit code. Pitch selection circuitry 63 is coupled to voicing decision circuitry 95, where the pitch bits are all changed to zero only if it has been determined that the speech signal is unvoiced. Otherwise the pitch bits are not changed from the six-bit code that was initially assigned. Voicing decision circuitry 95 is coupled to vocoder data processor 96.

A/D circuit 44 is also coupled, via data switch 74, to multiplier 78, adders 84 and 86, and memory 88 all of which form a second portion of the pipeline processor. This portion performs vocoder spectrum analysis simultaneously with the pitch extraction computations carried out in the first portion of the processor. Spectrum analysis is carried out by computing the DFT of the product of the incoming speech wave and a triangular weighting function (or window). The values of the window weighting functions are computed in advance and stored in memory 88 in the following manner: Window ROM (read only memory) 68, which stores values of window increments (one increment for each of the 16 vocoder channels) necessary to generate samples of the window, is coupled by data switch 80 to adder 84. Memory 88, which stores the instantaneous values of the window function, is also coupled to adder 84. Beginning, in memory 88, with an instantaneous window value of zero, the increment for each channel is sequentially added in adder 84 to and then subtracted from the instantaneous window values stored in memory 88 so that a complete set of sampled values of the triangular function are generated for each channel. The width of these windows, which differs for each channel, is controlled by control unit 52 which specifies the number of additions and subtractions to be performed for each channel. The output of memory 88 is coupled by data switch 76 to multiplier 78 where the samples of incoming speech signal are multiplied by the window function. The DFT computation, as will be discussed subsequently, requires a multiplication of the input signal-window function product by sine and cosine waves. Quantized samples of the required sine and cosine waves are stored in ROM 72. Memory 88, which stores the sequences of addresses necessary to select the correct sequence of samples, is coupled to ROM 72 for the selection of samples. ROM 72 is also coupled by data switch 76 to multiplier 78, where multiplication of the input signal-window product by the sine and cosine waves takes place. The result of the DFT processing is available at the output of adder 84 in the form of the sum of the squares of the real and imaginary parts of each of the complex frequency coefficients. Adder 84 is coupled to ROM 92 which stores tables of logarithms so that the vocoder amplitude parameters can be converted into three bit words each word representative of one-half the logarithm of the sum of the squares of the real and imaginary parts of each complex frequency coefficient; i.e., the frequency synthesizer output is representative of the logarithm of the magnitude of each of the complex frequency coefficients. Logarithmic steps are chosen because the human aural perception of the loudness is proportional to the logarithm of sound energy. A three-bit code is chosen because it provides quantum steps at approximately 4dB intervals, thereby conforming to conventional vocoder practice. This process is repeated for each of the 16 vocoder channels. ROM **92** is coupled to encoder **93** where the data is encoded into conventional 2,400 bits per second (Bps) or 1,200 Bps (delta coded) formats. Encoder **93** supplies its output signal to vocoder data processor **96** where the vocoder amplitude parameters for each vocoder channel are stored. Voicing decision circuitry **95**, which processes the vocoder data to determine the presence of voiced or unvoiced sounds, is also coupled between adder **84** and vocoder data processor **96**.

The data frame structure and transfer of data from vocoder operation to modem operation can best be described with reference to FIG. **11***a* which presents a more detailed block diagram of the vocoder and modem data processors **94** and **96** of FIG. **1**.

Each frame of vocoder data, consisting of a 54-bit word comprising (a) 48 bits of vocoder channel information (three bits per channel, except for the 16^{th} channel whose least significant bit is clamped to "One" and used as a synchronization bit) and (b) six bits of pitch information, is stored in parallel register 500. Upon occurrence of a "read down" pulse generated by a 2,400 Hz clock **506** and a divider circuit **502**, the vocoder frame is transferred in parallel via gate **504** into vocoder data register **508**. When the modem is acting as a modulator, the data in vocoder data register **508** is then shifted serially into modem data register **510**, under control of clock **506**, at a 2,400 bits per second (BPS) rate. When 32 bits of data, which comprise a modem data frame, have been accumulated in modem data register **510**, the entire 32-bit modem frame is transferred, under control of clock **506** and divider **512**, via gate **514** into parallel register **516**.

When modem modulation is required, each 32-bit modem frame is transferred to ROM **68** and the second portion of the pipeline processor, where modulation is performed. In this process, each 32-bit modem frame is divided into 16 pairs of bits. Each bit pair is then used to determine the differential phase shift to be applied to one of 16 modem tones. The binary representations of the 16 modem tones, each bearing vocoder information via the quadrature DCP SK modulation, are then combined to form a composite FDM modem output signal. In the preferred embodiment of the invention, the 16 modem tones are:

TONE	FREQ.
1	935 Hz
2	1,045 Hz
3	1,155 Hz
4	1,265 Hz
5	1,375 Hz
6	1,485 Hz
7	1,595 Hz
8	1,705 Hz
9	1,815 Hz
10	1,925 Hz
11	2,035 Hz
12	2,145 Hz
13	2,255 Hz
14	2,365 Hz
15	2,475 Hz
16	2,585 Hz

(It should be noted that because the 54-bit vocoder frames are converted first into a 2,400 BPS bit stream and then, in sequential order, into 32-bit modem frames, these 16 modem channels do not correspond with regard to data content on a one for one basis with the 16 vocoder channels.) Modem modulation is accomplished by cosine table look-up using the cosine table stored in ROM **72**. (See FIG. **1**). Memory **88**, which is coupled to ROM **72**, stores the ROM cosine table addresses for each modem channel. MODNO and CHNO ROM **68**, which contains the channel address increments for generating a tone representative of each channel as well as introducing the four-phase shifts required for four-phase DCP SK modulation are coupled by data switch **80** to adder **84** where the necessary increments are added to the cosine table ROM **72** addresses stored in memory **88**. The ROM **68** addresses necessary to access the proper channel address increments are determined from the 16 bit pairs transferred to ROM **68** from modem data processor **94** and the channel number transferred to ROM **68** from control unit **52**. ROM **72** is coupled by data switch **47** to accumulator **48** where the sample values of each of the required tones are accumulated. These sample values are combined in accumulator **48** to produce the samples of the composite modem output signal representative of the 16 FDM modem channels to be transmitted. In the preferred embodiment of the invention each sample is represented by an eight bit word. Accumulator **48** is coupled to D/A **50** where the composite modem signal is converted to analog form. D/A **50** may be coupled to the modulator of a conventional communications system (not shown).

"Receive" Mode

In the "receive" mode, analog signals modulated by speech information in the manner just described are received by communications receiver **42** (see FIG. **1**), which may be of conventional form. Communications receiver **42** is coupled to A/D **44** where the analog received signal is sampled periodically and a binary representation of each analog sample is generated. As in the "transmit" mode, the sampling rate is 8,250 Hz, and each sample is represented by an eight-digit binary word.

The interconnection of the major components of the pipeline processor of FIG. **1** is the same in the "receive" mode as it is in the "transmit" mode. However, in the "receive" mode different functions are performed and different data switches are activated.

In the second portion of the pipeline processor, comprising multiplier **78**, adders **84** and **86**, and memory **88**, modem demodulation is performed. The initial step in demodulation is the separation of the composite FDM signal into the 16 separate modem channels. This is accomplished by a DFT analysis of the composite signal and is carried out in the same manner and by the same apparatus (multiplier **78**, adders **84** and **86**, memory **88**, ROM **68**, ROM **72**, and data switches **76** and **80**) as is used for the spectrum analysis performed in the "transmit" mode. The resulting frequency coefficients which define each of the modem channels are stored initially in memory 88 and then in memory 66 which is coupled from memory 88 by data switch 64. This makes available at the same instant, a current set of frequency coefficients (in memory 88) and the previously received set of coefficients (in memory 66) which, as will be discussed subsequently, is necessary for execution of the differential phase algorithm used to demodulate the DCP SK modem signal and recover the vocoder channel information. Memories 66 and 88 are coupled to multiplier 78 by data switches 74 and 76 respectively for the multiplication steps required in the differential phase computation. Multiplier 78 is coupled to accumulator 90 where the addition steps required in differential phase computation are carried out. Accumulator 90 is coupled to modem data processor 94 for storing of each 32-bit modem frame recovered by the demodulation process.

The transfer of data from modem to vocoder operation can best be described with reference to FIG. 11*b* which presents a more detailed block diagram of the vocoder and modem data processor 94 and 96 of FIG. 1. The processing components of FIG. 11*b* are the same as those of FIG. 11*a* and are therefore similarly numbered. However the logical interconnection (via data switches which are not shown) is different. Accumulator 90 (FIG. 1) is coupled to parallel register 510. Upon occurrence of a "read down" pulse generated by clock 506 and divider circuit 512, each 32-bit modem frame is transferred in parallel via gate 514 into modem data register 516. Modem data register 516 is coupled to vocoder data register 500 into which data is transferred at a 2,400 BPS rate under control of clock 506. When 54 bits of data, which comprise a vocoder frame, have been accumulated in vocoder data register 500, the entire 54 bit frame is transferred, under control of clock 506 and divider 502, via gate 504 into parallel register 508.

Vocoder data processor 96 (FIG. 1) is coupled to decoder 97 which converts pitch frequency (which by convention is transmitted) to pitch period (which is required in impulse response synthesizer 46). In addition, when the 1,200 BPS delta coded mode is used, decoder 97 decodes the delta modulation.

Decoder 97 is coupled to linear interpolator 61 where parameter filtering, the first step in the speech synthesis process, is performed. Parameter filtering is necessary to remove a 44.4 Hz noise component which results from vocoder data being supplied to the synthesizer at a rate of 44.4 frames per second (2,400 BPS/54 bits per frame).

Impulse response synthesizer 46 (shown in more detail in FIG. 2), which synthesizes the impulse response of each vocoder channel, is coupled to multiplier 58 by data switch 54. Memory 62, which stores the vocoder amplitude parameters, is also coupled by data switch 56 to multiplier 58 where the product of the impulse response and the amplitude parameter of each vocoder channel is obtained. Multiplier 58 is coupled by data switch 47 to accumulator 48 where digital samples representative of the composite synthesized voice signal are formed. Accumulator 48 is coupled to D/A 50 where the analog composite of the speech signal is formed. D/A 50 may be coupled to a speech transducer (not shown) of conventional form.

Unvoiced sounds are indicated by the presence of zeros in all of the six pitch-representative bit positions of the vocoder frame. Those zeros are detected by circuitry in impulse response synthesizer 46, which gates the first portion of the processor into the unvoiced synthesis mode. In this mode, the channel impulse responses are not used. Instead binary samples of sine waves at the center frequency of each channel are generated by cosine ROM 72, which is coupled to multiplier 58 by data switch 56. Memory 62, which stores the vocoder amplitude parameters, is also coupled by data switch 56 to multiplier 58, where the samples of the sine waves are modulated by the vocoder amplitude parameters. Noise generator 59 is coupled to adder 60 for further modulation of each of the sine waves. This processing produces, in effect, a band of noise centered at each vocoder channel frequency, and modulated by the appropriate channel amplitude parameter. The remainder of unvoiced synthesis processing by accumulator 48 and D/A 50 is the same as in voiced synthesis.

Control unit 52, which contains gating and timing circuitry of conventional form, is coupled to all data switches and shift registers to control the flow of all data in the system.

Structural Details of Components of FIG. 1 System

FIGS. 2 through 4 and 12 show in more detail the structure of the major components of the system of FIG. 1. (The constant inputs shown in these figures, viz, "One or Zero," "0," "K," "106" are internally generated by connecting the gate inputs to appropriate constant voltage levels.)

FIG. 2 — Impulse Response Synthesizer 46

The impulse response synthesizer 46 of FIG. 1 is shown in detail in FIG. 2. Pitch counter 103 receives samples of filtered pitch from memory 62 (FIG. 1). Pitch counter 103 is coupled to pitch logic network 105 which transmits pulses to impulse flip-flop IMPFF 112 when each new pitch period should commence. Impulse ROM 116 stores the impulse responses for each of the 16 vocoder channels. In the preferred embodiment of the invention, the impulse response for each vocoder channel is represented by 80 samples which are read out of memory at a 8,250 Hz rate. The addresses in ROM 116 which are to be accessed are generated by combining a channel time signal supplied by control unit 52 (FIG. 1) to addend register (ADR) 100, with index numbers which circulate in the five-word circulating memory consisting of adder 101, index logic network 110 and memory 114. Each word in this loop circulates once per channel time. The channel time, 6.7 microseconds, is the time it takes to perform five calculations and make five data shifts. (See subsequent discussion of control unit 52 for further discussion of timing.) Memory 114 is coupled to and controls the readout of data from impulse ROM 116. Pitch pulses supplied by pitch logic circuitry 105 to impulse flip-flop (IMPFF) 112 cause IMPFF 112 to change to the set condition. IMPFF 112 which is coupled to index logic network 112 then causes the addresses in the circulating memory to be incremented by 1. IMPFF 112 is then cleared. Each address of the circulating memory continues to be incremented by one until all the impulse responses are read out of ROM **116**. At that time the index numbers in the circulating memory are reset to correspond to the “0” addresses until the arrival of the next pitch pulse. Impulse ROM **116** is coupled to accumulator **119** which stores the samples of the impulse responses generated during each channel time.

FIG. 3 — First Portion of Pipeline Processor

FIG. **3** shows the first portion of the pipeline processor in more detail. The interconnection of addend register (ADR) **126**, augend register (AGR) **128**, adder **130**, and summer **132** which together are comprised in accumulator **48**; buffers **144** and **146**, ADR **148** and ADR **154**, AGR **150** and AGR **158**, adder **152** and summer **156** which together are comprised in adder **60**; and multiplicand register (MC) **134**, multiplier registers (MP) **136** and **142**, multiplier **138** and product register **140** which together are comprised in multiplier **58** are shown. Buffers **160, 162, 164, 168, 170** and **172**, which are connected to memory **66** serve to provide along with memory **66**, a 91-word circulating loop for computation of the autocorrelation coefficients during pitch extraction, and to allow for reorganization of words in memory **66** during modem differential phase computations. Reorganization of the words is required to change from calculating the real part of the differential phase vector to calculating the imaginary part.

Pitch selection circuit **63**, which comprises selection logic network **157**, modulo 90 counter **155**, pitch register **159** and ROM **161**, determines, during the “transmit” mode, the pitch frequency from the autocorrelation data transferred to circuit **63** from the first portion of the pipeline processor.

Parameter filtering during voice synthesis operation in the “receive” mode is accomplished by linear interpolator network **61** in conjunction with memory **62**. As will be discussed subsequently, linear interpolation is accomplished by determining the difference between successive transmitted samples of vocoder data and then adding a portion of the difference to subsequent samples within a frame. The vocoder data frames are supplied to buffer **151** by decoder **97**. Buffer **151** is coupled in turn to buffers **149, 147**, and **145** which provide suitable storage and time delay necessary for the interpolator processing. Buffers **151** and **147** are also coupled to adder **143** where the difference between successive samples is determined. Shift networks **137** and **139** and adder **141** then compute the required fraction of this difference, which is to be added to subsequent vocoder data. Shift network **139** is coupled to memory **62** where this addition takes place.

FIG. 4 — Second Portion of Pipeline Processor

FIG. **4** shows the second portion of the pipeline processor in more detail. Cosine ROM **72** comprises ROM **182** which stores 150 binary samples representative of a sinusoid. The frequency of the sinusoid produced is dependent upon the order in which the samples are read out, and the phase is dependent on which sample is selected as the starting point. Tone index register (TXR) **174**, which contains the address of the next sample to be read out, is coupled to logic network **178**. Modulation index register (MXR) **176** which contains the next address necessary to generate a tone having the proper phase to implement modem DCPSK modulation is also coupled to logic network **178**. Logic network **178**, which is coupled to index register (XR) **180**, selects (depending on whether a vocoder or modem tone is required) either the contents of TXR **174** or MXR **176** to be loaded into XR**180**. XR **180** is coupled to ROM **182** for selection of the next sample to be read out. ROM **182** is coupled to tone register TR1**184** where the sequence of samples necessary to generate each required tone is stored.

The interconnections of MC**186**, MP**188**, multiplier **190** and product registers **190** and **194**, which are comprised in multiplier **78**; ADR **196**, AGR **198**, adder **200** and summer **202** which are comprised in adder **84**; ADR **204**, AGR **206**, adder **208** and summer **210**, which are comprised in adder **86**; and ADR **212**, AGR **214**, adder **216** and summer **218** which are comprised in accumulator **90** are also shown.

Adder **84** is coupled to and provides spectrum analyzer filter outputs to voicing decision circuitry **95**. Voicing decision circuitry **95** comprises scratch-pad registers **225** and **226** where data is held temporarily during the voicing computations, and flip-flop circuitry **227** where the voicing decisions are made. The decision as to presence of voiced or unvoiced sounds, based on energy in the spectrum analyzer filters, is made in the following manner:

First, by use of adder **84** and registers **225** and **226**, a summation is made of the outputs from analyzer channels **1** through **5**. These registers act as “scratch pads” in which data can be held temporarily for later reinsertion into the adder. The sum of the five lowest frequency channels, designated TOT5, is transmitted to memory **88**. The summation process continues with the remaining 11 channels, so that the sum of all 16 channels is also formed. This quantity, designated TOT16, is also transmitted to memory **88**. TOT5 is then compared in logic circuitry **227** with a constant designated KZ, which is permanently “wired in” by connecting to appropriate constant voltages inputs to voicing logic and flip-flop circuitry **227**. If TOT5 is greater than or equal to KZ, voicing logic circuit **227** recognizes that fact as a partial requirement for a voiced condition. TOT **5** is also multiplied in multiplier **78** by a second permanently stored constant which is designated as KY, and the product is compared with TOT **16**. If TOT **16** is less than the product of KY and TOT **5** and if the condition TOT **5** greater than or equal to KZ has already been fulfilled, the voicing logic circuit **227** produces a “1” indicating a voiced frame. TOT **16** is also compared with a third constant designated KX. If TOT **16** is greater than or equal to KX the frame of data is also treated as voiced. If neither criterion for voicing is fulfilled, the voicing logic produces a “0.” When a frame is unvoiced the pitch extractor output is forced to an all-zero condition. When a frame is voiced, the pitch extractor output is gated into the vocoder bit stream and stored in vocoder data processor **96**.

Since an unvoiced sound is produced by turbulent air passing through a constriction of the mouth or throat, a large amount of high frequency noise will be present. Therefore, in the preferred embodiment of the invention, the test adopted to determine presence of voiced or unvoiced sounds makes use of the presence of a large amount of energy in the high portion of the frequency spectrum during unvoiced sounds.

The physical significance of the parameters KX, KY and KZ is as follows:

KX is a high-threshold parameter, KZ is a low threshold parameter, and KY is a constant of proportionality. In the preferred embodiment of the invention, each vocoder amplitude parameter can have an integer value between 0 and 127 (six bits). KX is set at 150, KY at 1.9 and KZ at 20. These values were chosen empirically by examining different values of TOT **5** and TOT **16** in simulation work. TOT 5 represents the energy in the low-frequency portion of the speech spectrum, and TOT **16** represents the total energy in the speech spectrum. If TOT **16** is greater than or equal KX, the total speech energy is high, indicating presence of voiced sounds. If TOT **16** is less than the product of KY and TOT 5, low-frequency energy constitutes a significant portion of the total speech energy. In addition, if TOT 5 is greater than or equal to KZ, the low frequency energy content of the speech wave exceeds at least a minimum amount. The processor then will determine the presence of voiced sounds whenever the speech energy is very high (TOT **16** $\geq$ KX) or when the speech energy is of medium amount and is concentrated in the low frequency region (TOT **5** $\geq$ KZ and TOT **16**$<$KY $\times$ TOT **5**). An unvoiced condition will occur whenever the total speech energy is either very low (TOT 5$<$KZ and TOT **16**$<$KX) or is at medium strength concentrated in the high frequency region (TOT 5 $\geq$ KZ and TOT **16**$>$KY $\times$ TOT **5**).

FIG. 4 also shows the circuitry used to encode the vocoder output data in the "transmit" mode and the circuitry to decode the vocoder input data in the "receive" mode.

Encoder unit **93** which comprises register **220**, counter **222** and logic and comparator network **224**, operates in conjunction with ROM **92** and vocoder data processor **96** to encode the vocoder output data into a 2,400 BPS or a 1,200 BPS standard format. The output of ROM **92** is a three-bit word representative of the amplitude parameter of each channel. The least significant bit for channel **16** is forced to assume a "1" value, making an effective two-bit description for that channel with the constant "1" acting as a synchronization bit. For 2,400 BPS operation, no further processing is carried on in encoder **93** and the three-bit words are inserted directly into vocoder data processor **96**, where they become part of the vocoder data frame.

However, for operation at 1,200 BPS, delta coding is required to maintain compatibility with conventional vocoder equipment. Channels **1**, **2**, **3**, and **10** are processed as in the 2,400 BPS case, in that the three-bit codes are inserted directly into vocoder data processor **96**. The codes for channels **3** and **10** are also inserted into counter **222** which is an up/down counter with preset capabilities and a "round-off" feature. This feature causes the counter to remain unchanged if it contains a minimum count and receives a step-down signal, or if it contains a maximum count and receives a step-up signal. ROM **92** is coupled to register **220** to which the three-bit word for each of the remaining channels (**4** through **9** and **11** through **15**) is transferred. Register **220** and counter **222** are coupled to logic and comparator network **224** where their contents are compared. Network **224** is also coupled to vocoder data processor **96**. If the contents of register **220** are greater than the contents of counter **222**, a "1" is gated to the vocoder data processor as the one-bit delta code for that channel. Counter **222** is then stepped-up by "1" subject to round-off. If the contents of counter **222** are greater than register **220**, a "0" is gated to vocoder data processor **96**, and counter **222** is stepped down subject to round-off. After all channels have been processed, a "1" is gated into the vocoder data processor as a synchronization bit.

Decoder unit **97**, which comprises pitch ROM **229**, input register **228**, reference register **230**, decode logic **231**, and decode ROM **233**, converts received vocoder data into a format suitable for vocoder synthesis. Vocoder data processor **96** is coupled to pitch ROM **229** where the six pitch bits which are representative of pitch frequency, are converted to a six-bit word representative of pitch period. When the 2,400 BPS format is used, this is the only decoder function performed. When the 1,200 BPS delta coded format is used, the remainder of the decoder circuitry functions to convert the delta coded information into the standard three bits per channel format.

FIG. 12 — Control Unit **52**; Timing

FIG. **12** shows a portion of control unit **52** of FIG. **1**, and timing diagrams which illustrate basic system timing.

Crystal oscillator **518** provides the basic 5.94 MHz clock source from which all processor timing pulses are derived. Oscillator **518** is coupled to counter **520** which divides the 5.94 MHz frequency modulo **8**. The outputs of the three stages of divider **520**, designated ϕ**1**, ϕ**3**, and ϕ**7** are used to control operation of all arithmetic units. They each provide outputs at 1.347 microsecond intervals, which is designated as the system word time. This is the time between processor calculations and data shifts. Counter **520** is coupled to counter **522** which counts modulo **5** and thereby provides time slots for the execution of five complete consecutive operations in each arithmetic unit of the processor within a 6.734 microsecond interval designated as "channel time." Counter **522** is coupled to counter **524** which counts modulo **18** to provide time slots for groups of 18 channel times. A complete cycle of counter **524** takes place every 0.1212 milliseconds and corresponds to the system sampling rate of 8.25 KHz. Thus, since there are 18 channel times during each sampling interval, computations for the 16 vocoder channels or the 16 modem channels can be performed consecutively with two additional channel times available for auxiliary functions. A complete processing cycle for vocoder analysis, and pitch extraction takes place during 180 sampling intervals. Counter **526** which is coupled from counter **524** provides capability for counting each such processing cycle.

Theory of Operation

Mechanization of the algorithms used to implement the vocoder and modem functions will be explained with the aid of flow diagrams shown in FIGS. **5** through **10**. The reference numerals shown in parentheses within the logic boxes of the flow diagrams refer to the particular apparatus of FIGS. **1** through **4** and **11** and **12** by which the particular logical operation is carried out. The unparenthesized reference numerals designate respective steps of the algorithm.

Autocorrelation Pitch Extractor

$$R_{(\tau)}=\frac{1}{\tau}\int_0^{\tau} f(t)\cdot f(t+\tau)\,dt \qquad (1)$$

The input signal, $f(t)$, is multiplied by a stored replica of itself delayed by τ, $f(t+\tau)$. The product is time-integrated over the interval 0 to τ, and the integral is averaged over τ. The function R(τ) is evaluated for various values of τ. The value of τ which yields the largest value of R(τ) is taken to correspond to the fundamental pitch period of the speaker's voice. In the actual mechanization of this algorithm, the autocorrelation function of equation (1) is approximated by:

$$R_{(\tau)}=\sum_{n=1}^{m} f(nT+\tau), \qquad (2)$$

where T is the sampling interval (1/8250 Hz) and m is chosen so that mT equals the maximum expected pitch period. In the preferred embodiment of the invention, m is set to the maximum value at the beginning of each voiced interval and thereafter adjusted to the period previously found for the speaker's voice.

FIG. 5 — Flow Diagram of Algorithm for Computing Pitch Period

FIG. 5 shows the flow diagram for this algorithm. The algorithm consists of two phases: computing the autocorrelation function as approximated by equation (2), and determining the value of τ for which the autocorrelation function peaks. To compute the autocorrelation function, the input speech wave, in step **230**, is sampled at the system sampling rate (8,250 Hz) by A/D converter **44** and the samples are converted to digital form. Each pitch extraction interval (or frame) comprises 180 sample times. During the first half-frame (90 samples), each incoming sample is sequentially multiplied, step **234**, by each of the preceding samples in the frame. The delay is obtained by circulating, step **232**, the preceding samples in a 91-word input sample delay line (ISDL). The products which are obtained are accumulated, step **238**, in a 90-word correlation accumulator delay line (CADL). During this process the two delay lines are recirculated synchronously. Because of the one word difference in delay line lengths a "slippage" between samples being correlated occurs at the rate of one sample per delay cycle. This allows the 90-word delay line to accumulate, step **236**, in successive words the cumulative sums of autocorrelation products taken between samples separated by 1 to m. The additional word in the 91-word delay line also permits the insertion of each incoming sample into that delay line during the first half-frame. At the end of the first half-frame, the 90-word delay line contains sums of from one to 89 terms representing correlation products of samples separated by a delay time of from one to 89 sample times. During the second half-frame, multiplication continues and one word per sample time is transferred to correlation accumulators located in memory **62**. Thus at the end of 180 sample times, the correlation accumulators each contain the sum of 90 correlation products representing pitch periods of from one to 90 sample time intervals. In the peak picking phase, each autocorrelation sum is transferred, step **240**, to a first comparison register. The contents of the first comparison register and a second comparison register, which is initially set to 0, step **246**, are then compared **242** by subtraction. If the number in the first register is greater than the number in the second register the contents of the first register are transferred into the second register for subsequent comparisons and a pitch count **250** corresponding to τ is also gated, step **248**, into the pitch register. Thus when all correlation sums have been processed, the maximum value of all stored values of R(τ) will reside in the second comparison register. The corresponding value of τ is equal to the pitch period. Logic step **244** is provided to: insert the greater input from each comparison into second comparison register **246**, to set the second comparison register **246** to "0" at the beginning of each frame, and to disable comparison **242** except during a prescribed interval optimized to minimize false autocorrelation peaks.

During the first voiced frame, the prescribed interval is selected from accumulated sums $n = 27$ to $n = 90$ corresponding to the upper pitch frequency of 305 Hz (8,250/27 and a minimum pitch of 92 Hz (8,250/90) respectively. During the remainder of the voiced interval the searched region is limited to within plus or minus 20 samples of the τ at which a peak was found in the last frame.

The range of measurable pitch periods can be changed to cover the pitch frequency range of 70 Hz to 300 Hz conventionally used in vocoders by adding an additional delay between A/D conversion **230** and one word delay **232**. If for example a 27-word delay were to be inserted, pitch periods corresponding to frequencies from 70 Hz to 300 Hz would be measurable.

Spectrum Analyzer

A spectral analysis equivalent to that performed by a conventional channel vocoder analyzer is performed by using a computation of the discrete Fourier transform (DFT) of the speech wave.

The DFT is characterized by:

$$Ar=\sum_{n=0}^{m-1} x(nT)\ \text{exp}\ (-j2\pi rn/m) \qquad (3)$$

where

A_r = the rth Fourier coefficient,

$x\,(nT)$ is the sampled waveform to be analyzed,

T = the time between sample points (1/8250 Hz),

n = the nth sample, and

m = the number of sample points to be analyzed (i.e., mT is the analysis frame time).

Analysis of a fixed number of unweighted samples (i.e., no adjustment of the amplitude of the samples) of the speech input is equivalent to analyzing the speech input as seen through a rectangular window in the time domain. The equivalent vocoder filter that would result would have a (sin x)/x shape (i.e., the Fourier transform of a rectangular time function). In order to reduce the spectral contamination between vocoder filters a triangular window or weighting function, $w_v(nT)$, where v represents the vth channel of the analyzer, is used. The resultant filter has a (sin x/x)^2 shape in the frequency domain (i.e., the Fourier transform of a triangular time function), which results in lower spectral sidelobes and therefore less contamination between vocoder filters. In particular, the filter envelope shape for the vth channel of the analyzer for a triangular weighting function $w_v(nT)$ of length m_vT symmetrical about $(m_vT)/2$ and having a height of unity at the center, is

$$O(m_vT)=\frac{m\,T}{2}\left|\frac{\sin\,\pi\Delta f_v\,\frac{m_vT}{2}}{\pi\Delta f_v\,\frac{m\,T}{2}}\right|^2 \qquad (4)$$

where Δf_v is the frequency difference from the filter's center frequency. The magnitude of the function of equation (4) falls to within about 3dB of its peak value when

$$\pi\Delta f_{v_{3dB}}\,\frac{m_vT}{2}=1$$

Therefore the relationship between the bandwidth, B_v, of the vth filter and the length of the analyzer frame, m_vt, necessary to achieve this bandwidth is:

$$B_v=2\,\Delta f_v\;=4\,/\pi\;m_vT. \qquad (5)$$

Table I shows the required frame times and number of samples necessary to simulate the 16 channels of the vocoder analyzer. The values chosen for $f$and $B$are those conventionally used in vocoder practice.

TABLE I

Parameters for DFT Analyzer

channel no.	fv center frequency (Hz)	Bv bandwidth (Hz)	m_vT window length (msec)	m_v samples (T_{sam}=0.1212 msec)
1	263	132	9.7	80
2	393	132	9.7	80
3	525	132	9.7	80
4	660	132	9.7	80
5	791	132	9.7	80
6	925	132	9.7	80
7	1,060	143	8.9	73
8	1,225	165	7.7	64
9	1,390	185	6.9	57
10	1,590	215	5.9	49
11	1,820	245	5.2	43
12	2,080	280	4.6	38
13	2,380	320	4.0	33
14	2,720	365	3.5	29
15	3,115	420	3.0	25
16	3,565	490	2.6	21

Use of this algorithm for vocoder analysis permits direct achievement of filter banks having non-equal bandwidths (which conforms to conventional vocoder practice) without recourse to the combining of the outputs of many equal-bandwidth filters, as is commonly done in other systems.

Thus by substituting in equation (3) the sampled waveform to be analyzed for the vth filter,

$$x_v\,(nT)=w_v\,(nT)\cdot f(nT),\text{ where } f(nT)$$

is the sampled speech wave, the required computation is:

$$A_{rv}=\sum_{n=0}^{m-1} f(nT)w_v(nT)\;\exp\;(-j2\pi rn/m_v) \qquad (6)$$

r is the ratio of the vth channel center frequency, fv, to the basic frequency spacing of the Fourier series $1/m_vT$. Therefore by substituting $r=f_vm_vT$ in equation (6), the DFT coefficients can be represented by:

$$A_{rv}=\sum_{n=0}^{m-1} f(nT)w_v(nT)\;\exp\;(-j2\pi nf_vT) \qquad (7)$$

In order to avoid computation in complex arithmetic and thereby minimize equipment complexity, the relationship

$$e^j=\cos\Theta+j\sin\Theta \qquad (8)$$

is inserted into equation (7), to yield:

$$A_v=\left[\left(\sum_{n=0}^{m_v-1} f(nT)w_v(nT)\;\cos\;2\pi rf_vT\right)^2+\left(\sum_{n=0}^{m_v-1} f(nT)w_v(nT)\;\sin\;2\pi rf_vT\right)^2\right]^{1/2} \qquad (9)$$

FIG. 6 shows the flow diagram for implementation of equation (9). In step **252**, the input speech wave, $f(t)$, is sampled and the samples are converted to digital form. In step **254**, the train of samples representative of the speech wave, $f(nT)$, is multiplied for each channel by the triangular weighting function, $w(nT)$, and in steps **256** and **264** respectively the latter product is multiplied in one branch by the cosine coefficients and in another branch by the sine coefficients. In steps **258** and **266** respectively the results of steps **256** and **264** are then added by circulating, steps **260** and **268**, the contents of the accumulators through the adders. After the required number of samples in the analysis frame has been processed, the contents of the accumulators are squared **262** and **270**, and added **272** in pairs. The result represents the output of each of 16 vocoder analyzer channels, which is then encoded by encoder unit **93** into the standard 54-bit vocoder format.

Modem Modulation

Modem modulation consists of three processes:

a. tone synthesis, which is the generation of binary words respectively representative of the 16 modem carrier tones (sine and cosine functions for the DFT processes used in modem demodulation and vocoder analysis are generated in the same manner),

b. modulation, which consists of imparting the information carrying four-phase DCP SK modulation to the tones, and (c) generation of a modem preamble.

One hundred and fifty samples representative of one cycle of a sinusoid are permanently stored in ROM **72**. If these samples were continuously read out, in order, at the systems sampling rate of 8,250 Hz, a 55 Hz sine or cosine wave would be synthesized. However, if at each sampling time, the ROM address were incremented by p, instead of 1, a sinusoid of p times 55 Hz would be generated. By choosing p, any tone which is a multiple of 55 Hz can be synthesized.

FIG. 7 — Tone Synthesis

FIG. 7 is the flow diagram for tone synthesis.

The ROM 72 (FIG. 1) addresses necessary to generate the tones for each channel are called TONEX and are stored in memory **88**. The channel address increments corresponding to p which are necessary to generate all the different frequencies are called channel numbers (CHNO) and are stored in CHNO ROM **68**. In order to synthesize the tone, the ROM **68** address corresponding to the required CHNO is supplied to CHNO ROM **68** by the channel counter (counter **524** of FIG. 12, which counts modulo **18**). The CHNO is then selected (step **274**) and stored (step **276**) in CHNO register **70** (FIG. 4). The CHNO is then added (step **278**) to the current value of TONEX stored (step **286**) in memory **88**, and the sum is stored as the new TONEX. TONEX is then used to access (step **288**) the cosine table ROM. Although access (steps **290** and **292**) to both cosine and sine ROM's are shown in the flow diagram, from an apparatus standpoint these steps represent access to the same apparatus, viz, ROM **72**. From the 150 sinusoid samples stored therein either sine waves or cosine waves can be generated merely by choosing the correct order of readout of samples.

Since one cycle of a sinusoid is represented by up to 150 samples, the addition step is performed modulo **150**. This is accomplished by performing two series additions (steps **278** and **282**) and testing the sums (step **284**). In the first addition (step **278**), CHNO is added to the current TONEX value. The number 106 is then added (step **282**) to that sum, to form a second sum. If the second sum does not exceed 255 the first sum which has been temporarily stored (step **280**) is loaded into memory (step **286**) and becomes the new TONEX. If the second sum exceeds 255, the eight least significant bits of the second sum (which is of length nine bits) are loaded into memory (step **286**) and become the new TONEX.

FIG. 8 — Modulation

FIG. **8** is the flow diagram for modulation. As can be seen, the sequence of operations is similar to that depicted in the flow diagram for tone synthesis (FIG. 7).

The ROM 72 (FIG. 1) addresses necessary to generate the tones for each channel are called MODX and are stored in memory **88**. The channel address increments necessary to generate the required tone with the four phase changes are called modulation numbers (MODNO) and are stored in MODNO ROM **68**. In order to synthesize a tone with the required phase shift, the ROM **68** address corresponding to the required MODNO is determined by combining the CHNO supplied by the channel counter (**524** of FIG. **12**) with the bit pair corresponding to that channel, which is supplied (step **298**) by modem data processor **94**. The MODNO is then selected (step **300**) and stored (step **302**) in MODNO register **70** (FIG. **4**). The MODNO is then added MODULO **150** (steps **304**, **306**, **308**, **310** and **312**) to the current value of MODX and the sum is stored (step **314**) as the new MODX. MODX is then used to access (step **316**) the cosine table ROM. In order to form the composite MODEM signal, the samples of each individual tone are summed (step **318**) and the sums are accumulated (step **320**). The digital composite is then converted (step **322**) to analog form to form the analog modem composite.

The processor also has the capability for generating a MODEM preamble which can be transmitted for synchronization purposes prior to the transmission of data. In the preferred embodiment of the invention, the MODEM preamble comprises a 605 Hz Doppler tone and a synchronization tone at either 1,705 Hz or 2,915 Hz having 1,800 phase shifts. This preamble conforms to conventional modem practice. Since the tones required by the preamble are multiples of 55Hz, the processing steps are similar to those shown in FIGS. **7** and **8** with the only change being the prevention of the unwanted data tones from being accumulated (FIG. **8**, step **320**).

FIG. 9 — Demodulation

The demodulation processing consists of essentially two steps, the separation of the composite MODEM signal into 16 separate tones by apparatus performing a DFT filtering algorithm, and demodulation of the DCPSK modulation by apparatus performing a vector multiplication algorithm.

Filtering is accomplished by computing the 75-point DFT of the sampled composite received signal. The rth frequency coefficient of the DFT, A_r, is given by:

$$A_r = \sum_{k=0}^{N-1} (x_k) \exp\left[\frac{-j2\pi rk}{N}\right] \qquad (10)$$

where x_k is the kth sample of the composite and N is the number of sample points to be analyzed. Using equation (8), equation (10) can be transformed to:

$$A_r = \sum_{k=0}^{N-1} (x_k) \left[\cos \frac{2\pi rk}{N} - j \sin \frac{2\pi rk}{N}\right] \qquad (11)$$

Multiplying the numerators and denominators of the arguments of the trigometric functions of equation (11) by T, the system sampling time, and substituting $x(nT)$ (samples of the continuous function, $x(t)$) for x_k and W_r for $(2\pi r)/NT$ yields:

$$A_r = \sum_{n=0}^{N-1} x(nT)[\cos W_r nT - j \sin W_r nT] \qquad (12)$$

By letting the first sample within a transform be represented by the Kth sample of the continuous function, equation (12) may be written:

$$A_r = \sum_{n=K}^{K+74} x(nT)[\cos W_r(nT) - j \sin W_r(nT)], \qquad (13)$$

where $x(nT)$ represents the nth sample of the composite analog signal. Separating equation (13) into real and imaginary parts yields:

$$A_v \text{ (real)} = \sum_{n-K}^{K+74} X(nT) \cos W_r(nT) \qquad (14)$$

and

$$A_r\ (\text{imag}) = -\sum_{n=K}^{K+74} X(nT) \sin W_r(nT). \qquad (15)$$

Equations (14) and (15) indicate that the real and imaginary parts of a frequency coefficient can be obtained by multiplying samples of the composite by samples of the cosine and sine functions at that frequency and summing the products in accumulators for 75 samples.

This algorithm is illustrated in the left-hand portion of the flow diagram of FIG. 9. The received modem composite signal is sampled and the samples are converted to digital form (step **324**). Each sample is multiplied by the corresponding sample of the appropriate cosine wave (step **326**) and sine wave (step **338**). The products are accumulated (steps **328** and **340**) as per equations (14) and (15) and the real, *CR*, and imaginary, *CI*, parts of the current coefficients are stored (steps **330** and **342**) in the current real accumulator (CR–ACC) and the current imaginary accumulator (CI–ACC) respectively. At the end of each modem frame the contents of the accumulators are transferred (steps **336** and **348**) to the delayed real memory (DR–MEM) and delayed imaginary memory (DI–MEM) for differential phase calculation.

Calculation of differential phase makes use of the principle that the complex product of a first vector and the complex conjugate of a second vector yields a third vector whose magnitude is the product of the magnitudes of the first and second vectors and whose phase is equal to the difference in phase between the first and second vectors.

$$\text{i.e.}\quad \vec{A} = A\underline{/\theta A}$$
$$\vec{B} = B\underline{/\theta B}$$
$$\vec{A} \times \vec{B}^* = AB\underline{/\theta A - \theta B}$$

Thus the differential phase algorithm requires the computation of the vector product of the current frequency coefficient, $A_r = CR + jCI$, and the complex conjugate of the previously received coefficient, $A^*_{r-1} = DR - jDI$. The product thus obtained results in a differential phase vector, $\Delta\phi$, where, $\Delta\phi = Ar \cdot A^*r{-}1 = (CR \cdot DR + CI \cdot DI) + j(CI \cdot DR - CR \cdot DI)$. (16)
The phase of this vector $\Delta\phi$ is equal to the difference in phase between Ar and $Ar{-}1$.

The flow diagram for the differential phase calculation is shown in the right hand portion of FIG. 9. The real part of $\Delta\phi$ is calculated by multiplying *CR* and *DR* (step **334**) and *CI* and *DI* (step **346**) and then adding (step **354**). The imaginary part of $\Delta\phi$ is calculated by multiplying *CR* and *DI* (step **350**) and *CI* and *DR* (step **352**) and adding (step **360**).

The remaining steps shown in FIG. **9** are used for diversity combining. Either "in-band" or "out-band" diversity may be used. In "out-band" diversity, two 2,400 BPS modem composites are received and combined. In "in-band" diversity, a 1,200 BPS transmission rate is used, with the 32 bits transmitted in each modem frame actually consisting of two identical sets of 16 bits each. The $\Delta\phi$ vector is computed for each channel as previously described. Then the real parts of duplicate channels are summed (step **356**) and stored (step **358**) and the imaginary parts of duplicate channels are summed (step **362**) and stored (step **364**). The most significant bit, which is the sign bit, of each of the real and imaginary parts of the $\Delta\phi$ vector are combined (step **366**) to form a bit pair containing the four-phase information for that channel. This process is continued for each channel until the 16 bit pairs constituting a complete modem frame are available in the output register of the modem data processor **94**.

FIG. 10 — Voiced and Unvoiced Synthesis

The method of voiced synthesis uses a time-domain version of the inverse Fourier transform which, like the DFT analyzer, produces the effect of a vocoder filter bank. Impulse synthesizer **46** forms a sinusoidal oscillation, for each of the 16 channels, at the channels center frequency with a triangular window function imposed upon it. The effect, for each channel, is a sampled-data equivalent of the result of ringing with an impulse, a bandpass filter having a triangular envelope characteristic. Each channel oscillation is multiplied by its corresponding amplitude parameter which has been suitably filtered. All channels are then summed together to form the equivalent impulse response of a vocoder filter bank. New response waveforms are generated at intervals determined by the speaker's pitch frequency and added into the remaining portions of waveforms which have been generated but have not finished ringing.

Unvoiced synthesis is accomplished by generating 16 white-noise waveforms, low-pass filtering each one, and heterodyning each with a sine wave at the center frequency of a vocoder channel. The result is a spectral distribution in which a symmetrical noise distribution occurs around each channel center frequency, but the noise in each channel band is unrelated to any other band. As in the voiced case, each channel signal is modulated by a filtered amplitude parameter.

FIG. **10** is the flow diagram for voiced and unvoiced synthesis.

The initial processing steps describe the conversion of received pitch words into pitch pulses to be used for vocoder synthesis. The pitch bits received from vocoder data processor **96** are detected (step **368**) to determine the presence of voiced or unvoiced sounds. If the pitch bits are all zeros, unvoiced sounds are determined to be present and pitch pulse generation is inhibited. If the pitch bits are not all zeros, voiced sounds are indicated and pitch pulses are generated in the following manner:

The six-bit pitch frequency code is converted (step **370**) by a $1/x$ function into a number denoting the pitch period in terms of a number of sample times (i.e., $n \times 1/8.25$ KHz). In steps **372** through **376** a linear interpolation filtering operation is performed on the number to produce a smooth pitch variation (with time). The pitch signal is then gated into a six-bit digital count down circuit (step **380**). Once during each sample time, the count down circuit is decremented by "one." When its contents equal zero, a pitch pulse is generated and transmitted (step **384**) to impulse flip-flop **112** (FIG. **2**). Generation of the pitch pulse also enables gating of the next pitch word into the count down circuit (step **380**). Steps **386**, **388**, and **390** illustrate the generation of the addresses necessary to read the channel impulse responses out of impulse ROM **116** (FIG. **2**). Impulse response samples are read out (step **408**) from the ROM and accumulated (step **410**) during each channel time. These impulse response samples are updated once each word time. A filtered amplitude parameter is then used to modulate (step **412**) the summed channel impulse response. Filtering of the amplitude parameter is accomplished in steps **418, 420,** and **422.** New channel amplitude parameters arrive once per vocoder frame or approximately once every 185 sample times (54-Bits/2,400 BPS) × 8,250 samples/sec). Lowpass filtering is accomplished by interpolating linearly between successive frames of channel amplitude parameters. During each frame, 1/185th of the difference between the value of each amplitude parameter during the last frame and the new value during the current frame is computed. This amount is then added to the filtered value which is generated at each sample time within the frame. Linear interpolation is carried out by determining the difference (step **420**) between successive amplitude parameters, adding (step **422**) one-half of the difference to the difference to obtain 1.5 times the difference and then shifting (step **418**) 9 times (i.e., effectively dividing by 2^9) to obtain 1/185th of the difference. This portion of the difference is then added (step **418**) to each sample of the amlitude parameters read out of memory, to obtain a smoothed estimate of the amplitude parameter over each frame time.

For voiced synthesis, the result of the modulation process (step **412**) is then accumulated (step **404**). After all 16 channels have been similarly processed, the accumulated composite sample of synthesized speech is converted (step **406**) to analog form, and is available as the voiced synthesized output. The accumulated composite sample is updated once each channel time.

For unvoiced synthesis, this processing is modified slightly. Unvoiced sounds are detected by an all "0's" detector in the impulse response synthesizer **46** (FIG. **1**). After modulation (step **410**) of a sinusoid at the center frequency of each voice channel by the filtered amplitude parameter, the modulated signal is multiplied (step **402**) by samples of low-pass filtered white noise. The noise signal supplied (step **392**) by noise generator **59** (FIG. **1**) is filtered as shown in steps **396, 398** and **400** in the same manner that filtering of the amplitude parameters is accomplished (steps **418, 420,** and **422**). The result is accumulated (step **404**) and converted to analog form as in voiced synthesis.

We claim:

1. A digital apparatus having a first operating mode wherein said apparatus is responsive to a first input signal representative of a first speech wave in the time domain, and a second alternative operating mode wherein said apparatus is responsive to a second input signal (i) representative of the power spectrum and the pitch frequency of a second speech wave and (ii) comprising a plurality of differentially coherent phase shift keyed tones, said apparatus comprising:

first means for

a. computing from said first input signal the power spectrum of said first speech wave,

b. generating a binary representation of a modulated carrier comprising a plurality of differentially coherent phase shift keyed tones, said modulation bearing the power spectrum and the pitch frequency of said first speech wave, and

c. recovering from said second input signal the power spectrum and the pitch frequency of said second speech wave,

said first means comprising: first and second memories, a first adder having an output connected to the input of said first memory, a first multiplier, means for coupling the outputs of said first multiplier to an input of said first adder, first and second read-only memories, means for coupling the output of said first read-only memory to said input of said first adder, means for coupling the output of said second read-only memory to an input of said first multiplier, means for connecting the output of said first memory to said input of said first adder and to inputs of said first multiplier and to the input of said second read-only memory and to the input of said second memory, and means for connecting the output of said second memory to the input of said second memory and to an input of said first multiplier; and

second means for

a. computing the pitch frequency of said first speech wave,

b. supplying said pitch frequency of said first speech wave to said first means, and

c. generating from the power spectrum and pitch frequency recovered by said first means a binary representation of said second speech wave in the time domain,

said second means comprising: a third memory, a second adder having its output connected to the input of said third memory, a second multiplier having its output connected to an input of said second adder, means for connecting the output of said third memory to an input of said second multiplier and to an input of said second adder, means for connecting the output of said second memory to an input of said second multiplier, means, having an input coupled to the output of said third memory, for synthesizing the impulse response of a plurality of filters each responsive to pass a different one of said plurality of tones, means for connecting the output of said impulse response synthesizer means to an input of said second multiplier, means for generating a plurality of noiselike signals each having a center frequency corresponding to a different one of said plurality of tones, and means for connecting the output of said noise generator means to an input of said second adder.

2. A digital apparatus according to claim **1**, wherein said means for coupling the outputs of said first multiplier to said first adder comprises first and second data switches, means coupling one output of said first multiplier to an input of said first data switch, means coupling another output of said first multiplier to an input of said second data switch, a third adder, means coupling the respective outputs of said first and second data switches to respective inputs of said third adder, and means coupling the output of said third adder to an input of said first adder.

3. A digital apparatus according to claim **2**, further comprising pitch selection means and voicing decision means, means coupling the output of said second adder means to the input of said pitch selection means, means coupling the respective outputs of said pitch selection means and said third adder to respective inputs of said voicing decision means; vocoder data processor means and modem data processor means, means coupling the output of said voicing decision means to said first data switch and to an input of said vocoder data processor means, means connecting an output of said modem data processor means to an input of said vocoder data processor means, means connecting an output of said vocoder data processor means to an input of said modem data processor means, first accumulator means having an input coupled to an output of said first multiplier and an output coupled to an input of said modem data processor means, means coupling an output of said modem data processor means to the input of said first read-only memory, said means for coupling the output of said first read-only memory to said first adder comprising said first data switch and said third adder; a third read-only memory having an input coupled to the output of said third adder, encoder means having an input coupled to the output of said third read-only memory and having an output coupled to an input of said vocoder data processor, decoder means having an input coupled to an output of said vocoder data processor, linear interpolator means having an input coupled to the output of said decoder means and an input coupled to the output of said third memory and an output coupled to an input of said third memory; a third data switch having one input coupled to the output of said second read-only and having another input coupled to the output of said second multiplier means, second accumulator means having an input coupled to the output of said third data switch; first analog-to-digital converter means alternatively responsive to said first input signal and said second input signal, a fourth data switch having respective inputs coupled to the outputs of said impulse response synthesizer means and said first analog-to-digital converter and having an output coupled to an input of said second multiplier, a fifth data switch having respective inputs coupled to said output of said first analog-to-digital converter, said output of said second read-only memory, said output of said second memory and said output of said third memory and having an output coupled to an input of said second multiplier; a sixth data switch having respective inputs coupled to said output of said first analog-to-digital converter, said output of said first memory and said output of said second memory, and having an output coupled to said input of said second memory; a seventh data switch having respective inputs coupled to said first analog-to-digital converter, the output of said first memory, the output of said second memory, and that output of said first multiplier which is coupled to an input of said second data switch, and having an output coupled to one input of said first multiplier; and an eighth data switch having respective inputs coupled to the output of said second read-only memory and the output of said first memory, and having an output coupled to another input of said first multiplier.

4. A digital apparatus according to claim 3, wherein said first read-only memory stores values of a weighting function, and channel address increments, said second read-only memory stores values of a sine wave, and said third read-only memory stores values of logarithms, and wherein said apparatus also comprises a second digital-to-analog converter having its input coupled to the output of said second accumulator means.

* * * * *