# MICROPROCESSOR-BASED ANALOG VOICE SCRAMBLING TECHNIQUES

**Sergei Udalov**
**Axiomatix**
**9841 Airport Blvd., Suite 912**
**Los Angeles, California 90045**

## SUMMARY

Analog voice privacy techniques provide the advantage of being compatible with the 3 kHz audio bandwidth of the existing radio and telephone channels. The degree of privacy provided by an analog voice scrambling technique, however, is proportional to the number of time and frequency elements into which the voice signal can be divided as well as to the number of permutation patterns according to which the elements are scrambled. This implies the requirement for a high degree of signal-processing capability. Microprocessor-based implementations of an analog voice scrambling device provide a large potential for signal processing and scrambling. Furthermore, they provide this potential at a reasonable cost, small volume and moderate power consumption. In addition, a single microprocessor-based analog voice privacy device can be configured in software to yield various degrees of privacy, depending on a particular use and circumstances. Also, a variety of auxiliary functions such as timing, code generation, synchronization and analog-to-digital conversion can be time-shared within the same microprocessor chip, thus minimizing the requirement for support hardware.

The purpose of this paper is twofold: (1) to provide an overview of the existing analog voice privacy techniques, and (2) to specifically outline the capabilities of the microprocessor-based analog voice privacy system design, with a particular emphasis on achieving an analog scrambled signal compatible with the 3 kHz nominal audio bandwidth of the existing radio and telephone channels. Also, workable algorithms used for microprocessor-based analog voice scrambling in frequency as well as in time domain are described. Tape recordings of the voice scrambled and recovered with these algorithms are presented for comparison.

## INTRODUCTION

The requirement for voice privacy, once an exclusive prerogative of the military and diplomatic communities, is expanding rapidly into the civilian sector of the population. Not only law enforcement and public service agencies are expressing a strong interest in voice privacy, but the commercial and industrial users are as well. The reasons for this steadily increasing interest in voice privacy are twofold: (1) the ever-growing dependence of the civilian sector on radio and telephone communications, and (2) the concomitant proliferation of the inexpensive and readily available signal intercept equipment [1]. A typical list of the potential civilian sector users of voice privacy may read as follows:

- Law Enforcement Community
  - City
  - County
- Emergency Services
  - Fire
  - Ambulance
- Private Sector Users
  - Public utilities
  - Trucking firms
  - Oil companies
  - Business radiotelephones
  - Fishing fleets
  - Private security companies.

The degree of privacy required by a potential user depends on (1) the type of information communicated, and (2) the basic intent of the eavesdropper. Often, the monitoring of, say, emergency services, is simply a matter of idle curiosity. The equipment used is generally rather simple (typically, a standard scanner receiver) and, thus, the damage can result only if the interceptor attempts to interfere with the functioning of the emergency service. The use of a relatively simple voice scrambling technique can eliminate the majority of these eavesdroppers. Realistically, however, many cases of radio eavesdropping involve monitors whose intent is to gain a specific advantage over the communicator. Furthermore, those practicing such purposeful eavesdropping are generally better equipped than the average curiosity-driven listener and, consequently, the communicator must rely upon a rather sophisticated method of voice scrambling.

Having decided that he requires a voice scrambler for his communication, the user has to make the selection of the equipment which will meet his requirements in a most cost-effective manner. Quite likely, a user has already made a considerable investment in his communication gear and, therefore, he will be looking for the scrambling equipment which is most compatible with already existing equipment. Another important factor influencing the user's selection of the voice scrambling method is the bandwidth of the communication channel available to him. The conventional radio and telephone channels, for example,

limit the usable bandwidth to that of 3 kHz, i.e., the audio bandwidth of the human voice signal.

Fortunately for the user, there exists a rather extensive line of voice privacy devices and techniques which are compatible with the nominal 3 kHz bandwidth of a "standard" telephone-like communication channel. The devices which achieve communication privacy within the audio bandwidth are known as analog voice scramblers.

The purpose of this paper, therefore, is to present an overview of the typical analog voice scrambling techniques, with a particular emphasis on the microprocessor-based implementations. It is the microprocessor—a result of large-scale integration (LSI) development efforts of recent years—which brings new powers and capabilities to the field of the analog voice scrambler design. With a microprocessor, the signal-encoding methods hitherto impractical because of implementation difficulties are becoming available to the analog voice privacy users at an affordable cost and within small size packages.

## ADVANTAGES OF ANALOG VOICE SCRAMBLING

Basically, there are two methods for achieving voice privacy. The first is an all-digital technique. The second method involves analog voice processing and, depending on the degree of privacy required, may also employ digital technology.

With an all-digital voice scrambler, the voice signal is first digitized by the conventional methods such as PCM or Δ-Mod [2]. The digitized voice is then combined with a digital (normally, a binary) encryption code and applied to the transmission channel. Within the transmission channel, therefore, the composite signal remains in a digital format which requires a transmission bandwidth compatible with the original voice signal digitizing technique. This implies a transmission bandwidth requirement for nonsynthetic speech much higher than 3 kHz and more likely on the order of tens of kilohertz [3]. Figure 1 shows, qualitatively, the time and frequency domain relationships between the input and output signals of an all-digital scrambler. As can be seen from the figure, the output (i.e., encoded) digital waveform D(t) has no resemblance to input analog waveform A(t). The spectrum of the output waveform is accordingly modified and widened so that $f_c >> f_H$, where $f_H$ is typically 3.2 kHz. The wide bandwidth of the digitized and scrambled voice signal not only precludes its utilization with conventional transmission channels but also increases the accuracy requirements for the synchronization at the receiving end. With $f_c$ being in tens of kilohertz, the synchronization aperture requirement is in tens of microseconds. Therefore, although relatively simple in its implementation and virtually free of the time delays associated with scrambling, the all-digital voice privacy technique is incompatible with the standard 3 kHz-wide audio bandwidth of the existing radio and telephone equipment. It is also incompatible with the single-sideband (SSB) operation.

In comparison, the analog voice scrambling techniques are specifically designed for compatibility with the standard, telephone-like channel. Figure 2 shows the input/output relationships of the time waveforms and the spectra of an analog voice scrambler. Because of the variety of the implementation techniques of the analog voice scramblers, the following general statements can be made with respect to information shown in Figure 2:

- Scrambled waveform is modified for all techniques, yet it remains "speechlike"
- Time delay may or may not be present
- Spectral composition (i.e., frequency ordering) may or may not be altered
- Spectrum width is essentially unaltered
- Synchronization requirement is proportional to the inverse of the audio bandwidth.

It is the latter two features of the analog voice scrambling which make it compatible with the existing telephone and radio equipment as well as with the SSB operation.

The ease of interfacing with the host communication equipment is another salient feature of an analog voice privacy device. A simple interfacing at the audio terminals of the host equipment is all that is needed in most cases to provide a voice privacy capability to the user's communication gear. Figure 3 shows an example of how a mobile transceiver can be equipped with an analog voice scrambling device. As shown there, the "scrambled" output and input terminals of a voice scramble/descramble device (VSDD) are plugged into the microphone and speaker jacks, respectively, of the host radio transceiver. The microphone and speaker are then plugged into the "clear" input and outputs, respectively, of the VSDD. Depending on the type of transceiver, either half-duplex or full-duplex scramble operation is possible. A voice scrambler, particularly a microprocessor-based one, can easily accommodate a full-duplex operation because of time-sharing and multiplexing capabilities inherent in its architecture. Furthermore, a "clear mode" override, which is required by many emergency-service-oriented users, can also be safely accommodated by an analog VSDD.

In summary, the advantages of the analog voice scrambling technique over that of an all-digital method, can be summarized as follows:

- Does not require bandwidth expansion (thus, provides for bandwidth conservation)
- Compatible with standard telephone and radio link channels
- Compatible with single-sideband operation
- Affected by channel distortion in the same manner as conventional analog voice
- Can be used with existing analog communication equipment without requiring equipment modification.

These intrinsic advantages of the analog voice privacy techniques, combined with a microprocessor-based implementation, provide the potential user with an immediately realizable voice privacy capability.

## OVERVIEW OF ANALOG VOICE PRIVACY TECHNIQUES

Thus far, the discussion has been centered around the advantages of the analog privacy techniques in general, without regard to any specific implementation. Consequently, to provide for better appreciation of how a microprocessor-based implementation can enhance the capabilities of a particular analog voice scrambling technique, a description of several fundamental analog voice privacy techniques is in order.

The techniques used for analog voice scrambling fall into two basic categories: (1) frequency (i.e., spectrum) manipulation, and (2) voice signal manipulation with time [4]. To these two basic categories can be added secondary techniques such as amplitude modification and masking. Because the latter two techniques alter the overall dynamic range requirement for the transmission of the signal, their application must be exercised with care unless the user has full control over the characteristics of the communication channel available to him. On the other hand, from the two basic techniques, i.e., frequency and time scrambling, a large number of systems can be derived with many and varied levels of privacy available to the user [5].

### A. Frequency Inversion With and Without Hopping

Frequency inversion is one of the oldest methods used for achieving voice privacy. With this method, the spectrum of the prefiltered audio signal is inverted; in other words, the high frequencies become low frequencies and vice versa. Figure 4 shows the relationship between the spectra of the clear and inverted speech. The implementation of the frequency inversion is rather simple—the prefiltered clear speech signal is suppressed/carrier modulated onto a "carrier" (i.e., a tone), whose frequency is above the voice spectrum and the lower sideband is selected by filtering. This lower sideband constitutes the "scrambled" voice signal.

For the spectrum inversion illustrated in Figure 4, the frequency of the carrier tone is 3.8 kHz. Thus, the 3.5 kHz signal of original speech becomes a 0.3 kHz signal in the inverted spectrum and, similarly, the 0.3 kHz frequency remaps into 3.5 kHz, with the intermediate frequencies being remapped according to their relationship to the 3.8 kHz carrier tone.

This method of providing voice scrambling provides privacy only against a casual, disinterested listener. The very simplicity of generating the inverted speech makes it

vulnerable to even a relatively unsophisticated "tinkerer" eavesdropper who can use a balanced modulator/filter combination to provide descrambling. In fact, simply passing the inverted spectrum through the device used to generate the inversion in the first place restores the natural frequency order to the scrambled speech. Because of this simple relationship between "encoding" and the "decoding" process of a simple frequency inversion process, this scrambling technique is generally referred to as the "single-code" method. In other words, spectrum is either inverted or not; hence, a single code which, for the scrambler described above, is also a fixed code.

To increase the degree of privacy provided by the spectrum inversion method, frequency hopping can be added to the basic inversion. With frequency hopping, the inverted spectrum shifts in frequency with time, as shown in Figure 5. The frequency displacement ranges from 50-300 Hz and the frequency hop rate is typically in the range of 10-100 Hz. The increased privacy obtained with frequency hopping occurs because the intelligibility of a voice signal drops with an increased displacement from the normal frequency ordering of the voice spectrum [6]. Consequently, the eavesdropper equipped with a fixed-frequency recovery device will have to cope with a considerably high degree of "garbling" due to frequency hopping. The degree of such intelligibility reduction depends on the rate and degree of frequency displacement contained in the "recovered" hopped spectrum, such as shown in the right-hand portion of Figure 5. In comparison, the authorized listener has a frequency inverter (i.e., descrambler) which hops its carrier tone in synchronization with the hopping pattern of the carrier tone of the transmitting scrambler. Adding this additional degree of privacy to a simple frequency inverter introduces an element of time which, in turn, brings in the requirement for code synchronization between the transmitting and receiving stations. Because of the relatively slow hopping rates involved in the synchronization process, however, the accuracy requirement is not severe, and is on the order of tens of milliseconds.

The simplicity of a voice spectrum inverter and its frequency-hopped version still makes these devices a cost-effective alternative to some of the contemporary users. As shown later in this paper, a microprocessor-based analog scrambler can achieve frequency inversion and spectrum hopping with only a few commands in the assembly language.

### B. Bandsplitting with Permutations

The degree of privacy offered by a scrambler operating on the frequency structure of the voice signal can be increased considerably by dividing the spectrum into several subbands, as shown in the left-hand portion of Figure 6. Such frequency segmentation can be accomplished by passing the clear voice through a bank of contiguous bandpass filters. The outputs of the individual filters can then be transposed in frequency by an appropriate set of mixers and translation oscillators, followed by a bank of filters identical to the one

used ahead of the frequency transposition circuits. In the process of frequency transposition, some of the subbands can also be inverted in frequency, thus adding an additional degree of permutability to the scrambled analog signal. Such a combination of frequency transposition and inversion is shown qualitatively in the right-hand portion of Figure 6.

As in the case of simple frequency inversion, the bandsplitting followed by transposition and accompanied by occasional subband inversion does not increase the overall bandwidth of the scrambled signal, thus placing the bandsplitting technique into a class of true analog voice privacy devices. The degree of voice privacy provided by the bandsplitting and frequency transposition and inversion is far greater than that provided by simple frequency inversion and hopping. The reason for the higher degree of security is the much higher number of combinations and permutations to which the analog signal can be subjected during the encoding process.

It is important to note, however, that, although it is natural to assume that the larger the number of filters, the better the potential scrambling capability of the bandsplitting device, practical considerations set the number of subbands at five [5] and the typical bandwidth of the individual filters at about 500 Hz. Furthermore, despite the fact that potentially there are 3,820 permutations of five subbands with and without inversion ($5!x2^5$), only 11 are useful for providing good scrambling transpositions [4]. This is due to the well established fact that the voice signal possesses extremely high redundancy in its frequency spectrum.

The logical solution for increasing the degree of security of a bandsplitter/frequency inversion scrambler is to introduce the element of time into the permutation matrix. Specifically, the frequency transpositions and inversions are changed in time (i.e., “rolled”) according to a pseudorandom pattern determined by the key setting of a particular device. The period between permutations may range from seconds to milliseconds, with 0.25 to 0.5 seconds being typical.

Because of the introduction of the element of time, i.e., changing the permutations of the transpositions and frequency inversions, the rolling code bandsplitters provide a higher degree of privacy as compared to the static bandsplitters. Thus, it is the added element of time-varying permutation which imparts an additional and significant dimension to the analog voice privacy techniques based on bandsplitting.

### C. Time Segment Permutation (TSP)

So far, the scrambling techniques which operate on the frequency characteristics of an analog signal were discussed. The element of time was utilized by these methods solely for the purpose of changing the pattern of frequency permutation. The continuity of voice

signal in time, however, provides for another dimension in which the transposition of digitized speech elements can take place. One of the most common techniques for voice scrambling in time is known as time segment permutation, or TSP [7,8].

Figure 7 shows qualitatively the functioning of a TSP encoding. As shown there, of a clear voice is first loaded into a memory device and then, upon completion of the storage phase, is read out as a sequence of time-permuted segments of the original "clear voice" signal. The order of the permutation and its rate of change is determined by a preselected pseudorandom sequence which, in turn, may be changing with time. The duration of the stored, unchopped voice signal is typically 250 ms and the duration of the segments is 20-30 ms, which implies that they are shorter than the duration of a syllable [8].

Because the TSP is carried out in the time domain, it is particularly suitable for microprocessor-based implementations. Of specific advantage is the microprocessor's capability to store in and retrieve from its memory a large amount of digital data. Thus, a microprocessor-based analog voice privacy TSP system converts an analog system signal into a digital format, operates on the digital samples (i.e., permutes them) and reconverts the processed signal back to an analog format. Figure 8 shows the typical memory load and readout sequences of a TSP encoding frame.

The sequence of the frame permutation shown in Figure 8 serves as an additional explanation of the TSP encoding process shown in Figure 7. Specifically, the analog voice is digitized by an analog-to-digital converter, then stored in the random access memory (RAM) buffer. As shown in Figure 8, the loading of the buffer proceeds in a sequential manner, i.e., the load scan address increases linearly over the memory segments 1-6.

Upon completion of the load, the buffer is then read out in the permuted sequence, whereby segments 1-6 are delivered to the digital-to-analog converter (DAC) in sequence 4, 6, 2, 1, 3 and 5. At the receiving end, the inverse procedure takes place and the segments arriving in a scrambled sequence are rearranged to provide the original clear voice signal. Similar to the limitations inherent in the bandsplitting technique, the TSP has its limitations. Specifically, the limitation of TSP relates to the shortest usable segment length and number of "good" permutations, i.e., the permutations which make the scrambled speech most unintelligible.

The solution to the permutation "quality" problem lies in the use of permutations which have been generated and "screened" by a computer prior to their application in the scrambling process. On the other hand, the limitation on the minimum usable segment length is a more fundamental one because it relates to the transmission characteristics of the voice bandwidth channel [9]. Specifically, as shown in part (a) of Figure 9, a typical audio bandwidth channel has a considerable amount of time delay variation relative to the

time delay at its center frequency. This differential delay, when of the same order as the duration of the shortest speech segment, causes distortion over a significant portion of the speech segment. This distortion results in smeared boundaries between the permuted segments which, in turn, lead to loss of intelligibility of the received signal. Part (b) of Figure 9 shows this phenomena qualitatively. Consequently, with channel time delay variation being on the order of 3-10 ms, the minimum time segment duration is limited to the aforementioned 20-30 ms duration.

### D. Reversed Time Segmentation (RTS)

The problem associated with the limit on the shortness of the time segment duration may be alleviated if the voice signal segments themselves can be made unintelligible. If the segments are unintelligible, their duration can be increased without compromising the privacy of the device. The increased length of the segments, in turn, considerably reduces the distortion associated with the time delay variation of the communication channel.

One of the most efficient methods for destroying the intelligibility of speech segments is to reverse their delivery in time. Figure 10 shows qualitatively the voice signal scrambling pattern obtained with such a reversed time segment encoding method. Part a of the figure shows a portion of a speech signal which is progressing in a normal direction, with its segments being generated in the 1, 2, 3, 4, 5... sequence. Part (b) of the same figure shows how the direction of each of the segments (i.e., time intervals) has been reversed, yet the order of their delivery, i.e., that of outputting to the channel, has not been altered. The typical duration of these reversed segments can be in the 50-400 ms range, depending on the memory capacity of the system and other implementation considerations. Some of the salient characteristics of this technique reported in the literature [10] and confirmed by the author using microprocessor-based algorithms, are as follows:

- The ability to understand the contents of the encoded speech depends on the speech delivery rate
- For very slow delivery rates, it is necessary to increase the duration of the time interval
- A time duration interval of about 150 ms is sufficient to render conventional speech rates unintelligible
- The quality of the recovered voice signal suffers little from the artifacts of the coding/decoding process.

Another important feature of the RTS is that the spectral characteristics of the scrambled signal differ very little from those of the original signal. This makes the RTS particularly attractive from the standpoint of audio channel characteristics compatibility. Furthermore, as shown later in this paper, the implementation of the RTS is relatively easy, particularly

with a microprocessor. The RTS is also compatible with other aforementioned scrambling techniques and, thus, when combined with them, may offer a high degree o privacy.

## SALIENT FEATURES OF MICROPROCESSOR-BASED IMPLEMENTATIONS

Because the degree of privacy provided by a voice scrambler device is proportional to the number of permutations of time and frequency segments of the voice signal, a good scrambler generally combines several of the techniques described so far [11]. This implies the requirement for a considerable amount of signal processing. The microprocessor is therefore an ideal device for providing the signal processing necessary for obtaining a good scrambler implementation. With a microprocessor, the signal handling power of a high-speed digital computer is available to the designer of an analog voice privacy device at a reasonable cost, within reasonable size, and an acceptable level of power consumption.

Furthermore, microprocessor-based techniques allow many support functions to be implemented in software, rather than hardware, thus reducing the overall component count. For example, such functions as timing, code generation [12] and synchronization as well as the analog-to-digital conversion [13] can be time-shared within the same microprocessor chip, thus minimizing the requirement for support hardware. From the development and utilization standpoint, microprocessor-based implementations also offer the following advantages:

<u>Development Phase</u>
- Various techniques can be tried out with relative ease
- Modifications and refinements are easily implemented

<u>Utilization Phase</u>
- Powerful algorithms can be employed to maximize effectiveness
- Multilevel programs can be used within the same device
- Technique updates can be carried out simply by changing the ROM units.

Figure 11 shows a typical microprocessor-based microcomputer system which can form the heart of an analog voice scrambler unit. As shown in the figure, the main component of the system is the microprocessor unit (MPU) or, as it is alternatively referred to, μP. The unit shown is an eight-bit processor with an addressing capability of 16 bits. The latter provides an addressing capability of up to approximately 64 thousands of eight-bit words,i.e., a 64 kbyte capability. The architecture includes a read-only memory unit (ROM), a random access memory (RAM), and a peripheral input/output controller chip (PIO). A clock unit is shown external to the MPU. For most modern processors, the "clock" is actually a frequency-determining element such as a quartz crystal or a combination of R, L and C elements. The program which controls the overall functioning

of the microcomputer resides in ROM, while the dynamically changing data is stored in RAM. The PIO provides for the control and buffering of the digitized data flowing in and out of the microcomputer.

Figure 12 shows a functional block diagram of an analog voice scrambler which utilizes a microcomputer as its central signal processing unit. The configuration shown provides for a full-duplex operation of the scrambler. Basically, the scrambler shown converts an analog signal to a digital format, operates on the digital samples, then reconverts the processed signal back to an analog format. The duplex operation is made possible by the inherent time-multiplexing capability of the microcomputer.

For timing and code synchronization, a special synchronization receiver and generator are included in the scrambler unit. The synchronization process itself is generally a two-stage process, i.e., there is an initial phase, followed by continuous tracking and synchronization updates. The most commonly used techniques for these two phases [14] and their salient, features are as follows:

Initial

- Preamble followed by code information

Continuous

- Tone supplied along with the signal (subtracted at the receiving end)
    - Stable synchronization signal available during transmission (advantage)
    - Takes away from the total dynamic range (disadvantage)

- Synchronization information provided during pauses
    - Does not use up-channel dynamic range (advantage)
    - Requires good clock stability at the receiving end (disadvantage).

With an efficient μP-based design, a major portion of these functions is performed by the microprocessor software, thus increasing the function's efficiency and flexibility as well as providing for an optimum trade-off between the advantages and disadvantages of the various synchronization techniques.

## EXAMPLES OF MICROPROCESSOR-BASED ANALOG VOICE SCRAMBLING ALGORITHMS

### A. Frequency Inversion Algorithm

The algorithm for frequency inversion of a voice spectrum serves as a good introductory example of a μP-based voice privacy device implementation. This algorithm is based on

the fact that polarity inversion of alternate Nyquist-rate samples taken of a lowpass-filtered signal results in a spectrum whose lower half component is the original lowpass signal with its frequency components inverted [15].

Figure 13 shows the flow chart for the spectrum inversion algorithm. For implementation of this algorithm, it is assumed that the analog signal applied to the analog-to-digital converter (ADC) is lowpass-filtered by either a hardware or software implemented filter, as indicated by the line and microphone input filters in Figure 11. It is also assumed that the output of the digital-to-analog converter (DAC) is also lowpass-filtered, as indicated by the output filters in Figure 11.

As indicated in the flow chart of Figure 13, the system, after cold startup, continues through the initialization and setup of an executive mode. For the case of no frequency hopping, i.e., simple inversion, the executive mode will be minimal and may consist primarily of supporting the lowpass filter implementation if such implementation is incorporated into the software.

The implementation of the frequency inversion algorithm per se starts with loading a REGISTER with a value of 2. After this load, the processor executes auxiliary functions which may range from simple functions such as setting up a timing loop, if the processor is internally timed, to complex functions such as computation of the next frequency hop increment, if frequency hopping is used.

If the software lowpass filter implementation is not utilized, the next steps are then ADC conversion and data fetch, i.e., the sampling and temporary storing of the ADC output word. Following reading and storing of the ADC output, the contents of the REGISTER are decremented and tested for zero. Failure to pass the REG = 0? test results in the polarity inversion of the word obtained from the ADC. After the polarity inversion, the digital word is then delivered to the analog channel via the system output DAC. The REG = 0? test, or its previously obtained result, is then used for directing the program to either the inner loop (i.e., REG $\neq$ 0) or outer loop (i.e., REG = 0).

Consequently, if the first REG = 0? test resulted in a polarity inversion of the signal sample output by the scrambler, the second pass through the inner loop will ensure that the next sample is delivered to the output without inversion. Following this duplet of output samples, the program passes through the second REG = 0? test (or its equivalent) and returns to the SET REG = 2 step which initiates the next pass through this frequency inversion subroutine.

The presence of the “auxiliary functions” subroutines allows for addition of a frequency hopping to the aforementioned algorithms of simple frequency inversion. The exact

implementations of the hopping in software will depend on whether the system is self-timed or interrupt-driven. In either case, sufficient time is generally available for execution of the "auxiliary functions" during the intersample period of the basic spectrum inversion routine. It must also be pointed out that, if the system is indeed frequency-hopped, in addition to the inversion, provisions for the generation (at transmitter) and detecting/tracking (at receiver) of the synchronization signal must be included. In this case, the sharing of the synchronizing functions between the executive and auxiliary portions of the program must be optimized.

## B. Time Segment Permutation (TSP) Algorithm

As stated previously, the manipulation of a digitized signal in time is one of the salient features offered by a μP-based implementation of an analog voice scrambler. Figure 14 shows the flow chart for implementing a generic TSP algorithm. As shown in the figure, the cold startup is followed by an initialize phase which then leaves the program in the Run Executive mode.

Because the TSP is a highly synchronization-dependent program, one of the major functions of the EXEC mode is to generate the synchronization initiate pattern for a transmitter unit and to search for the corresponding synchronization pattern for the case of the receiver unit.

Assuming that the unit is in the RECEIVE mode, as implied by the SYNC? test, the descramble program is not initiated until the synchronization signal is detected. Upon detection and verification of the synchronization signal, the buffer addresses are selected (assigned) and the buffer load is initiated. If the program is set up to load in a scrambled fashion and to unload linearly, the addresses of the LOAD buffers are selected to provide the loading in a permutated sequence identical to that of the transmission readout sequence shown in Figure 8.

Having selected the randomized load addresses, the program initiates the actual load. For the load on which the ADC conversion is performed, the data is fetched from ADC held in temporary storage and, if necessary, the auxiliary functions are performed. The latter may include, as stated earlier, the random code generation for the next cycle, timing cycle generation and synchronization update and tracking. Upon completion of these auxiliary functions, the stored input signal (digitized) is distributed to an appropriate address in RAM and the next RAM storage address is computed.

Upon computation of the next RAM address, an END OF FRAME test is performed. If the test is not passed, the program returns to the CONVERSION phase and the fetch/distribute cycle is repeated until all RAM segments have been loaded according to the incoming

pattern. If the test is passed, thus indicating the end of a frame, the program tests for synchronization. If that test is also affirmative, it proceeds to modify the buffer addresses for the next load and current readout. This implies that (1) for the readout, the RAM addresses are sequence-ordered (i.e., linearized) and (2) for load, the addresses are selected to correspond to the next incoming randomized sequence.

This pattern of loads and readouts is repeated until the message is completed and the last RAM buffer unloaded to the analog output terminal via the DAC. Again it must be emphasized that, because of the synchronization requirements, an optimum "work load" division must be chosen between the executive and auxiliary functions.

### C. Reversed Time Segment (RTS) Permutation Algorithm

Figure 15 shows a block diagram which represents a generic system for the reversed time segmentation encoding. As shown in this figure, the system is comprised of an ADC, a block of RAM and a DAC. On the first pass (i.e., scan) through the memory, the RAM is loaded by the output of the ADC. During this pass, no signal readout is performed. On the second pass, however, the RAM is already loaded and the signal readout can commence. During the second scan, the first load is read out in a time-reversed manner and, immediately upon readout of each memory byte, the read-out address is filled with a new sample of a digitized speech signal. Thus, when the readout of the inverted speech is completed, the RAM is full and ready for the next readout of the time-inverted speech. In this manner, the alternate scans of the RAM produce the required time inversion of the voice signal delivered to the output of the scrambler.

An algorithm for performing an RTS scramble is shown by the flow chart in Figure 16. The structure of this algorithm is very similar to the one for the TSP, with the exception of the aforementioned timing differences and synchronization requirements. Thus, the executive functions, in conjunction with the auxiliary functions, perform the tasks of initial synchronization detection and subsequent update, respectively. Also, auxiliary functions can be used to provide additional degrees of signal randomization such as frequency inversion over random intervals of inverted speech readout. Other techniques mentioned earlier can also be "mixed in" via the auxiliary functions of the RTS program to provide for a higher degree of privacy.

## CONCLUSIONS

Analog voice scrambling techniques can satisfy the immediate as well as future needs of a variety of civilian users. The chief advantage of the analog voice privacy techniques is their bandwidth compatibility with the existing radio and communication equipment. Microprocessor-based implementation of the analog voice privacy, in particular, offers the

user the advantages of a high degree of privacy at a realtively low cost, small size and moderate levels of power consumption. Consequently, microprocessor-based implementations of several analog voice privacy techniques have been carrier out for demonstration purposes. Based on the generated and available tape recording data, the salient features of these implementations are discussed. Furthermore, the examination of a potential effect of new chip development on the existing microprocessor-based analog voice privacy device designs is examined in terms of achieving advanced configurations and improved performance.

## REFERENCES


1. Middleton, R. G., Scanner Monitor Servicing Guide, Howards W. Sams and Co., Inc. Indianapolis, Indiana, 1975.

2. Jayant, N. S., "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers," Proceedings of IEEE, Vol. 62, No. 5., May 1974.

3. Flanagan, J. L., et al, "Speech Coding," IEEE Transactions on Communications, Volume COM-27, No. 4, April 1979, pp 710-737.

4. Kahn, D., The Codebreakers, The Macmillian Company, New York, 1967.

5. McCalmont, A. M., "Communications Security for Voice-Techniques, Systems and Operations," Telecommunications, April 1973, pp 35-42.

6. Pappenfus, E. W., Bruene, W. B., and Schoenike, E. Q., Single Sideband Principles and Circuits, McGraw Hill, New York, 1964.

7. Leitich, A., "A Survey of Available Techniques for Voice Privacy," National Electronics Conference, 1977, pp 180-185.

8. Hartmann, H. P., "Analog Scrambling vs. Digital Scrambling In Police Telecommunication Network," 1978 Carnaham Conference on Crime Countermeasures, University of Kentucky, Lexington, Kentucky, May 1979, pp 47-51.

9. Baschlin, W., "The Integration of Time Division Speech Scrambling Into Police Telecommunication Networks," Proceedings, 1977 International Conference on Crime Countermeasures, pp 141-145.

10. Belland, E. and Bryg, N., "Speech Signal Privacy System Based on Time Manipulation," 1978 Carnaham Conference on Crime Countermeasures, University of Kentucky, Lexington, Kentucky, May 1978, pp 37-40.

11. McCalmont, A. M., "Achieving and Measuring High Security in Analog Speech Communications Security Devices," 1979 Carnaham Conference on Crime Countermeasures, University of Kentucky, Lexington, Kentucky, May 1979.

12. Vouga, C. A., "Speech Scrambling in Radio Communication," First International Electronic Crime Countermeasures Conference, Edinburgh, Scotland, July 1973.

13. Lesea, A. and Zaks, R., Microprocessor Interfacing Techniques, Second Edition, Sybex, Berkeley, California, 1978, Chapter 5.

14. Goode, G. E., "New Developments in Data and Voice Security," 1973 IEEE Electronics Security Systems Seminar Conference Record, pp 83-97.

15. Kak. S. C. and Jayant, N. S., "On Encryption Using Waveform Scrambling," BSTT Journal, Vol. 56, No. 5, May-June 1977, pp 781-808.

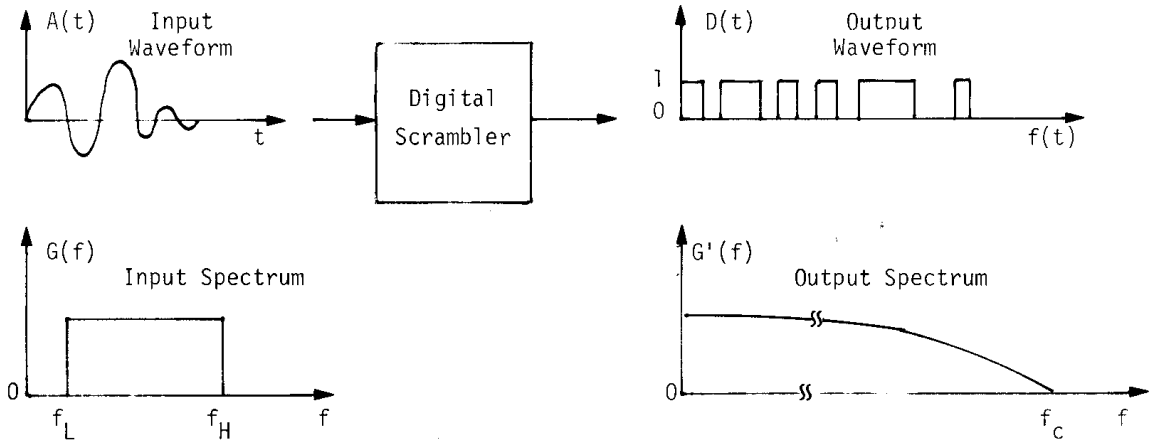


**Figure 1. Input/Output Relationship of the Time Waveform and the Spectrum of an All-Digital Voice Scrambler**

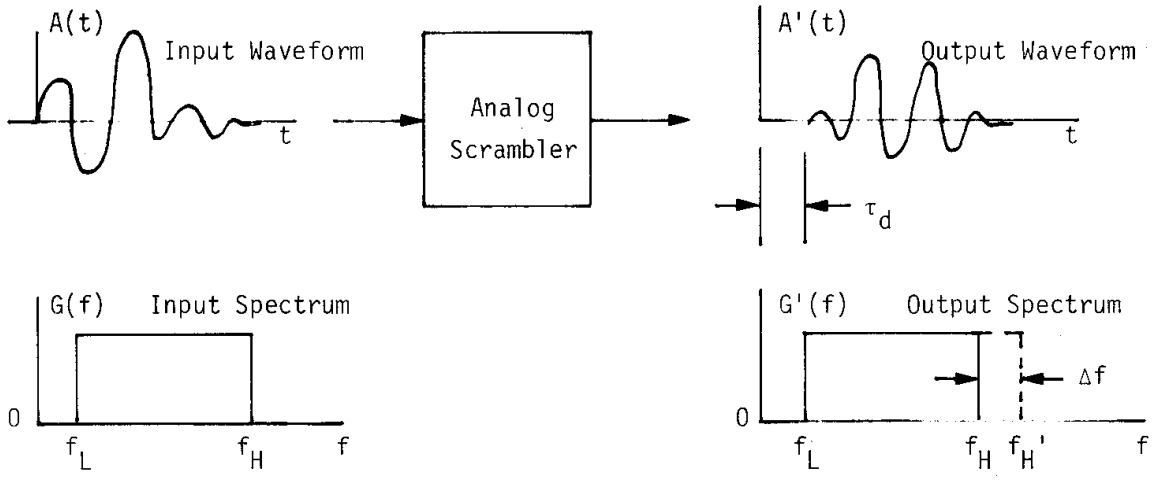


**Figure 2. Input/Output Relationship of Time Waveforms and the Spectra of an Analog Voice Scrambler**

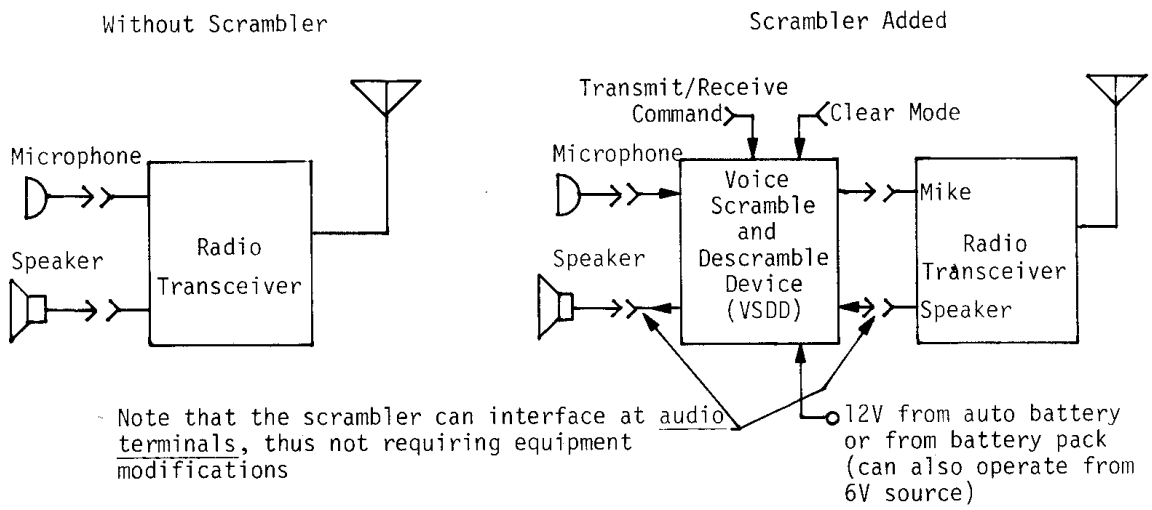


**Figure 3. Analog Voice Scrambling Device Provides for Simple Interfacing at Audio Terminals of Host Equipment**

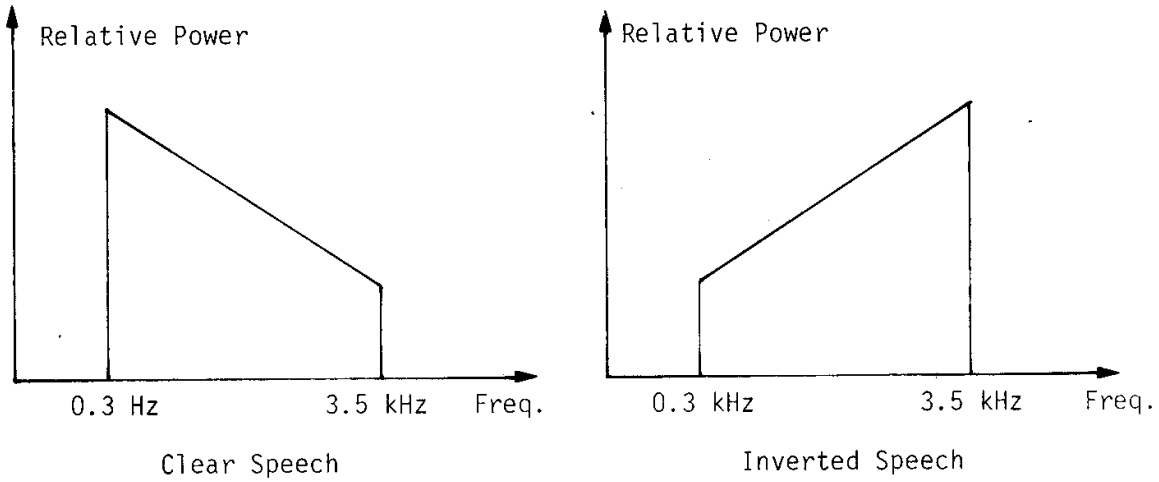


**Figure 4. Frequency Inversion Spectra**

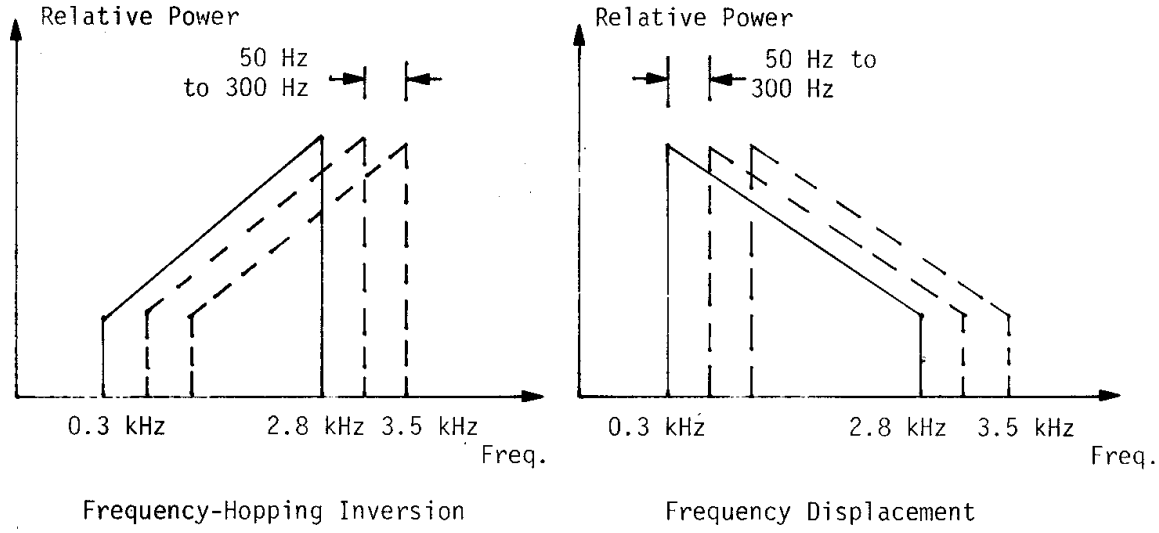


**Figure 5. Frequency-Hopping and Inversion Spectra**

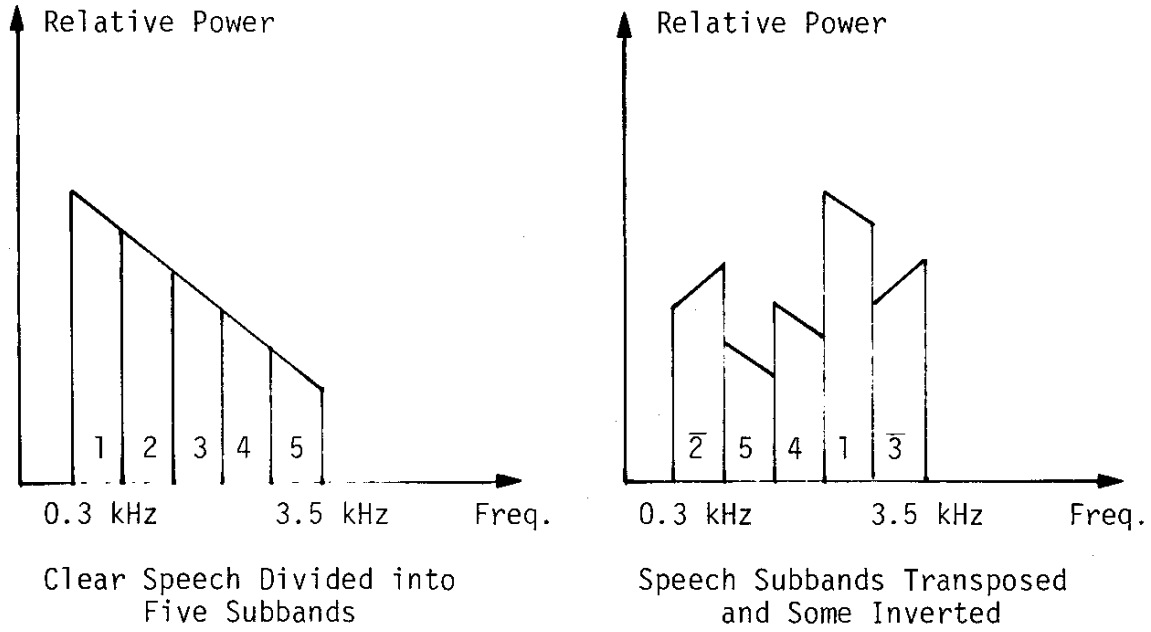


**Figure 6. Bandsplitting Combined with Inversion**

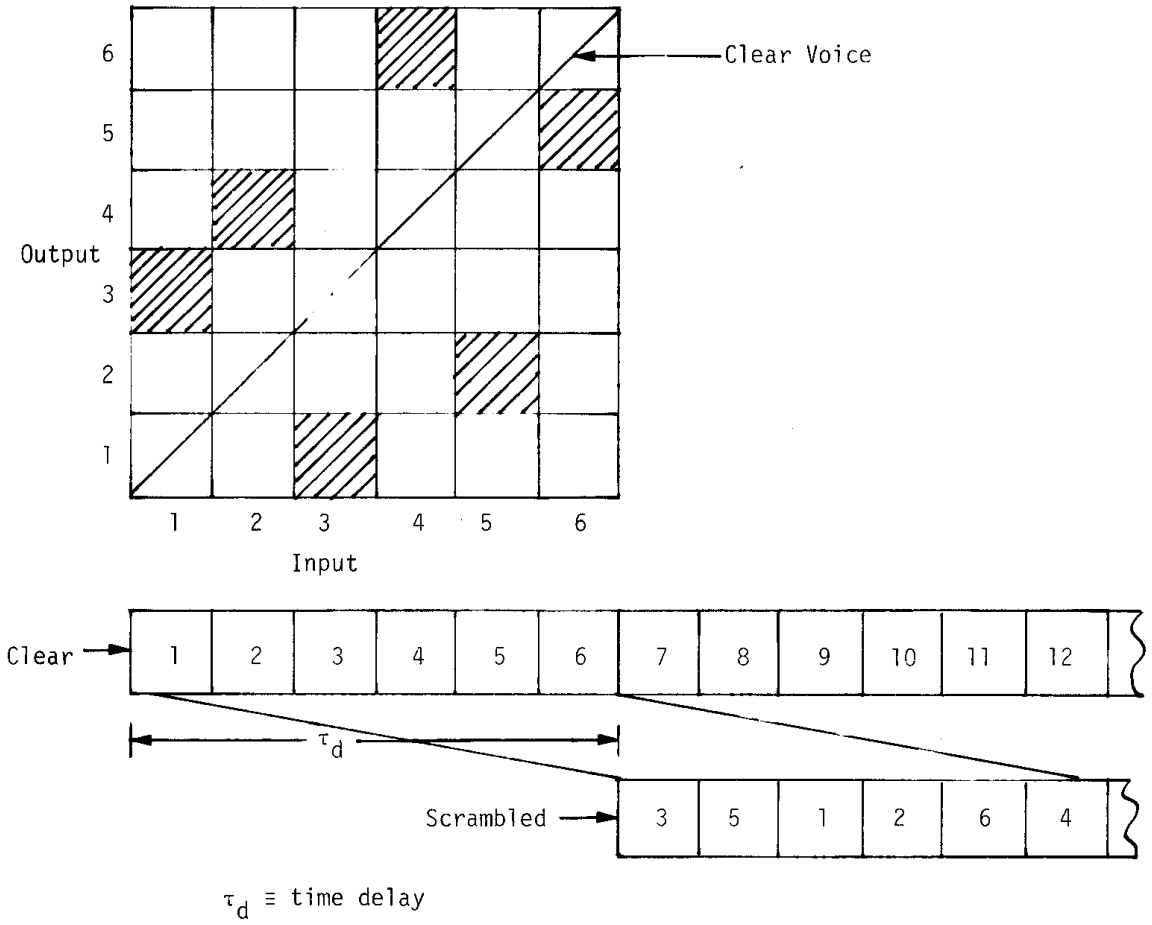


**Figure 7. Time Segment Permutation (TSP)**

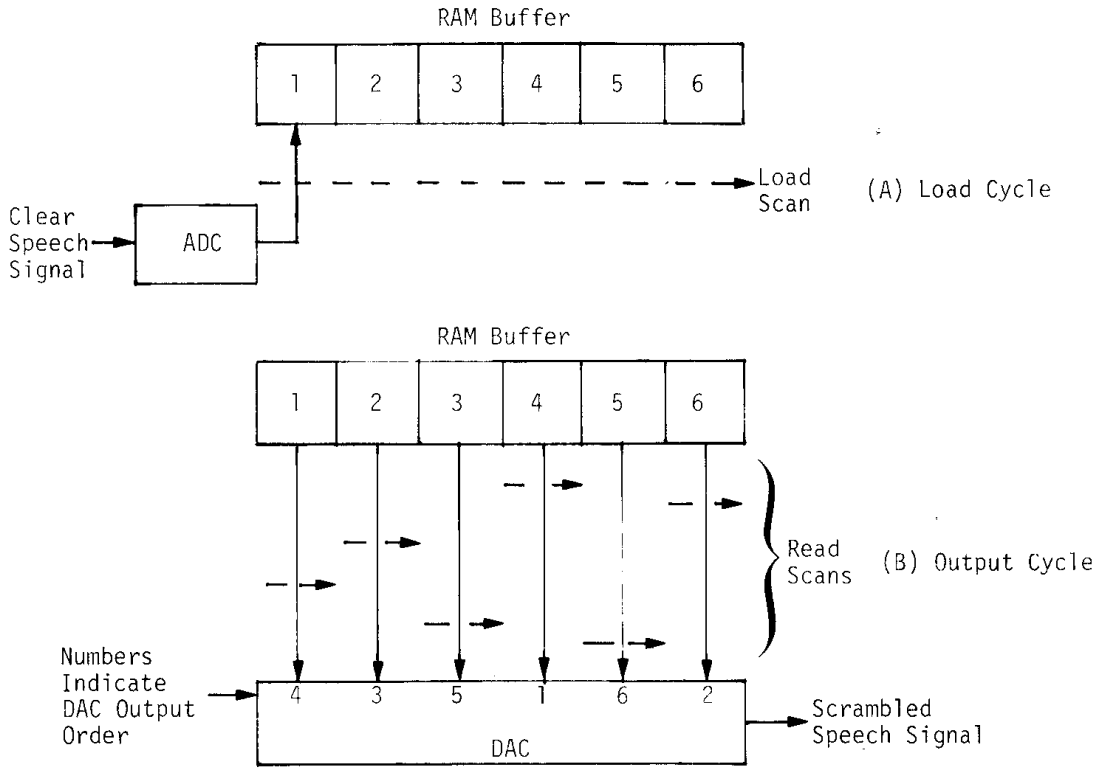


**Figure 8. Various Stages of the Time Segment Permutation Cycle**

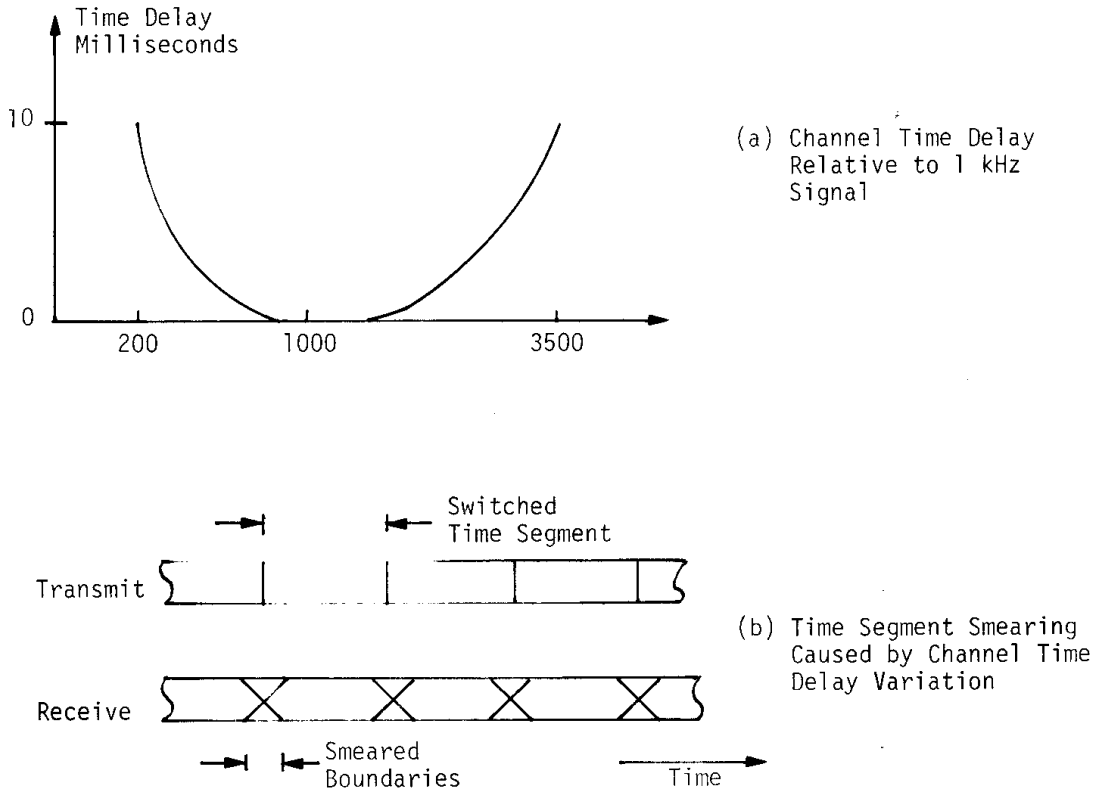


**Figure 9. Effect of Channel Group-Delay Variation on Time Segment Boundaries**

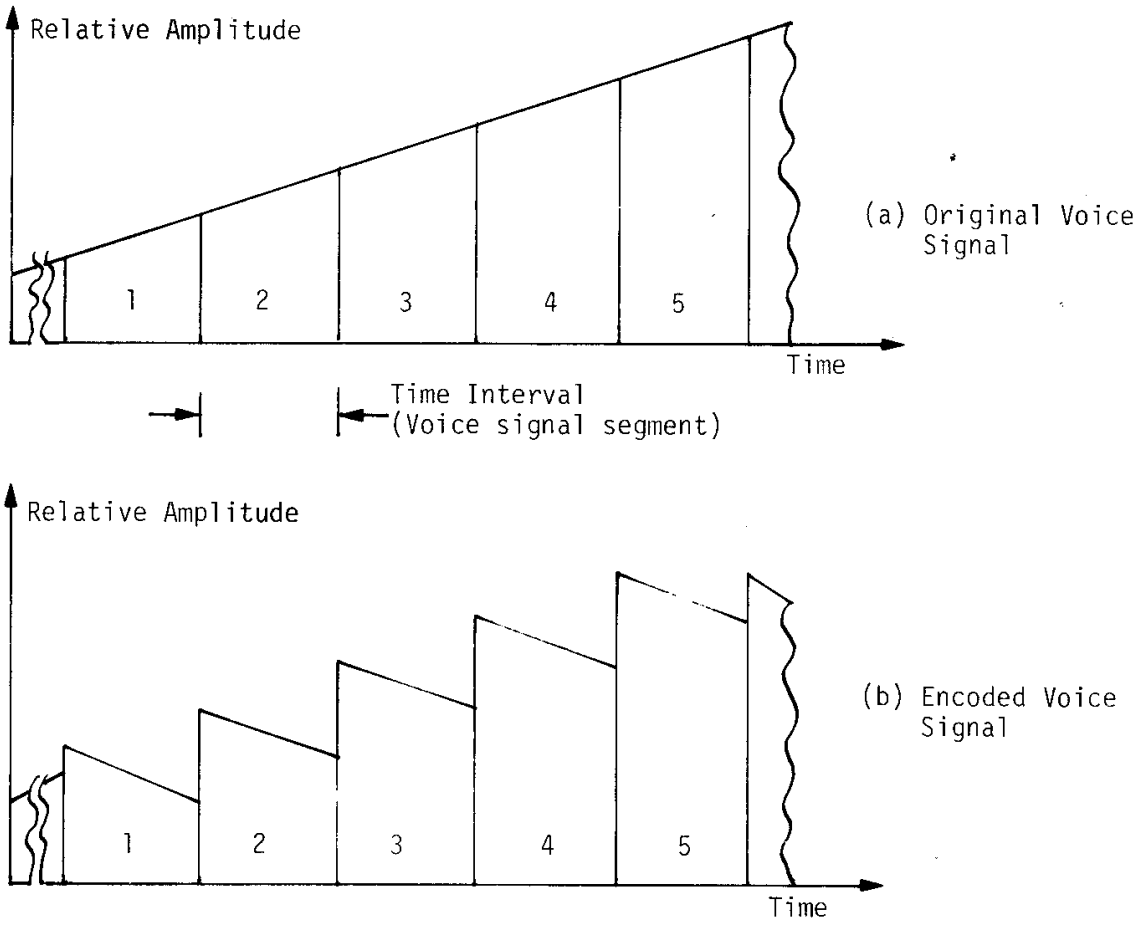


**Figure 10. Reversed Time Segment Encoding**

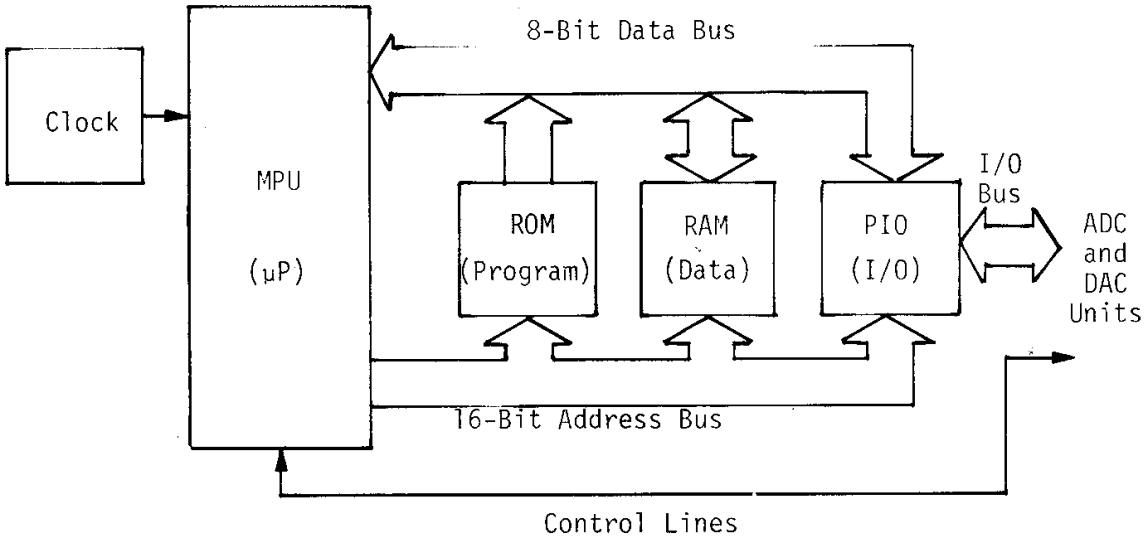


**Figure 11. Typical Microcomputer System Architecture**

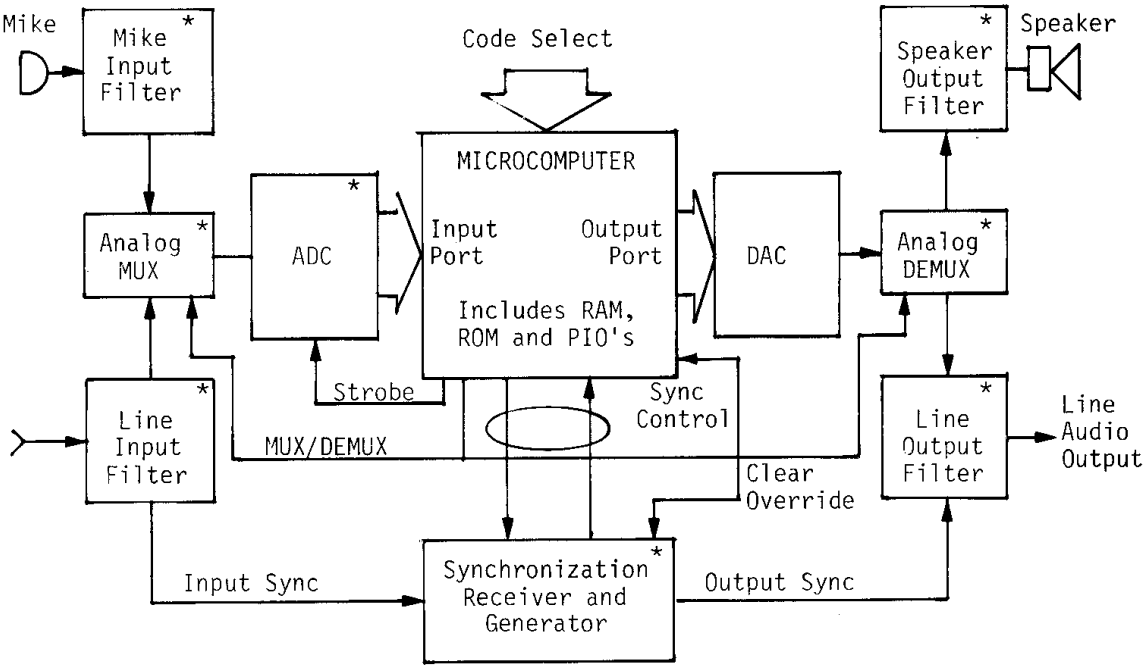


**Figure 12. Functional Block Diagram of a Full Duplex Analog Voice Scrambler/Descrambler**

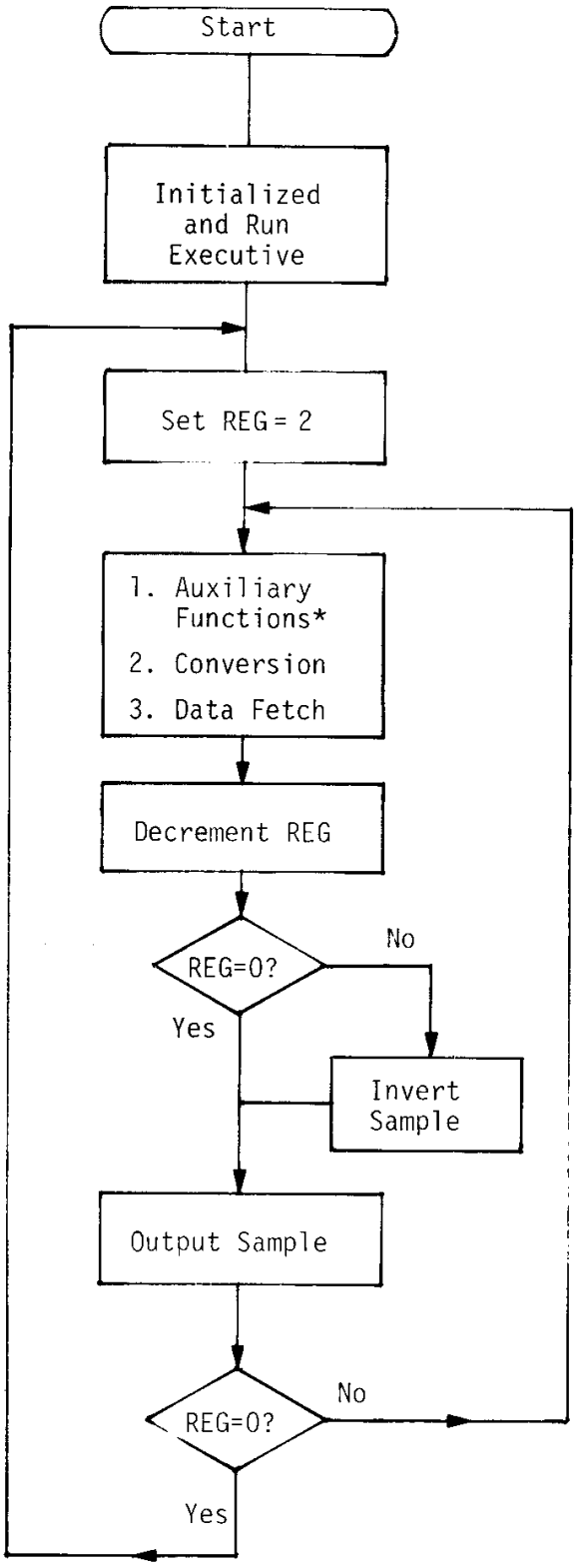


**Figure 13. Flow Chart for Spectrum Inversion**

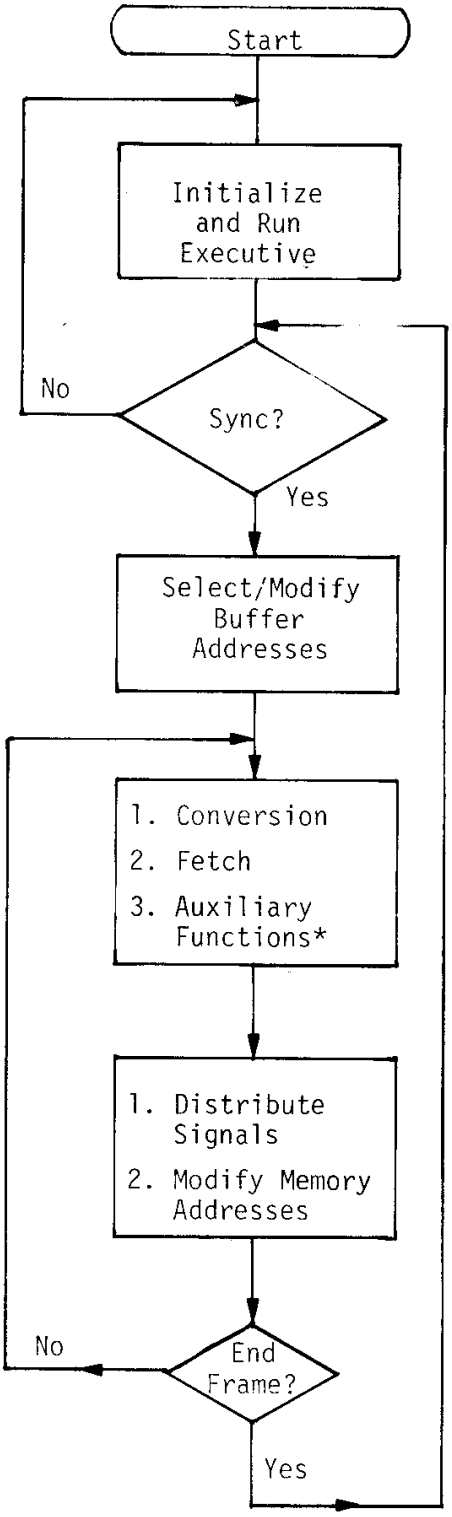


**Figure 14. Flow Chart for Time Segment Permutation (TSP)**

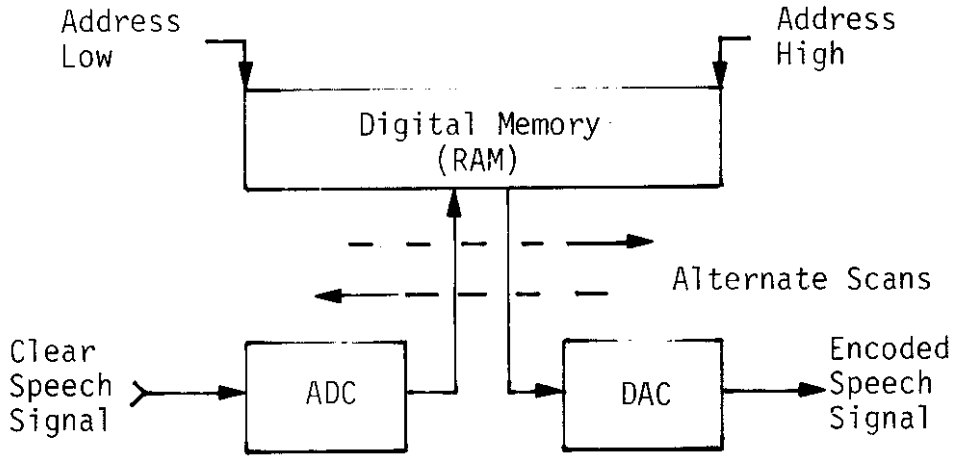


**Figure 15. Generic System for Reversed Time Segment Encoding**

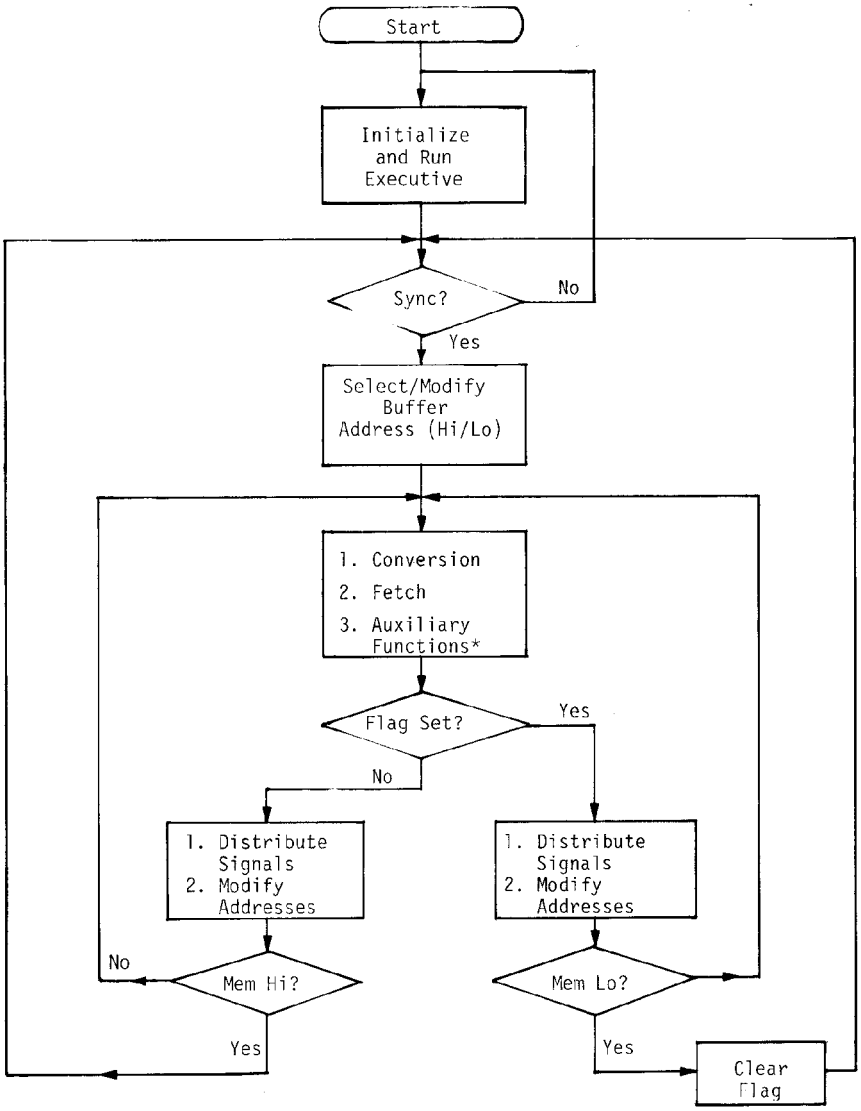


**Figure 16. Flow Chart for Reversed Time Segmentation (ITS)**