VOTAN
VPC-2000
TECHNICAL SPECIFICATION

VPN 74500029
REV. D (05/23/85)

Change History

Rev 1.0 - Initial Release 8/7/84
Rev 1.1 - Release 11/19

Section I - Model 2020 (no telephone)
- Model 2040 (no telephone, no compression)
Section 2.1.1 - 2.1.9 Add new flags applicable to voice functions.
Section 2.2.1 - Add description of new flag options:
Bit 2 - Enable Voice Detect
Bit 6 - Enable State Change Interrupt
Bit 7 - Enable DTMF Interrupt
Section 2.3 - Add new status bytes :
0C - DP Full
10 - Voice Detected
11 - DTMF Detected
12 - Deskset Off-Hook
14 - Function Not Available
15 - Deskset On-Hook
Section 3 - General renumbering of sections and adjustment of table of contents to match.
Section 3.1 - Add Telephone Monitor function
Section 3.2 - Add State Change usage of flag bit
Section 3.3 - Add input and output gain TCR parameters

Rev 1.2 - Release 4/7/85

Section 1.0.7 - All unused status bits are indeterminant
Section 2.3.11 - Template Load Error Status
Section 3.8 - Internal DTMF tone buffer
Appendix 2 - Add compressed speech format appendix.

Rev 1.3 - Release 05-30-85

Section 1, Figure 1.1a - Add strap drawing for 01000043
Section 1, Figure 1.4 - Chng audio path drawing

VPC2000 SPECIFICATION

TABLE OF CONTENTS

I. PRODUCT DESCRIPTION

The VOTAN PC board (VPC2000) provides a mix of voice technologies and complementary telephone functions on a single IBM PC compatible board. These technologies include :

* Isolated word Speaker Dependent Recognition

* Continuous speech recognition (speaker dependent).

* Real time speech compression and decompression (record and playback).

* FCC approved telephone loop interface with autodial, auto-answer, DTMF generation and detection capability.

The VPC2000 is similar in nature to many other peripheral devices in that it presents a few I/O locations to the system processor. Operations are initiated by the system processor (8088), data is supplied to or taken from the board by the system processor using programmed I/O. The general sequence of operation is that the system processor initiates a voice command by writing to a command register on the VPC. The board then requests the detailed command information from a control record structure called the VCR (Voice Control Record) and the required voice data to and from the system processor using interrupts. The command is executed and a final interrupt is set to the host processor followed by transfer of status. This sequence, as well as the form of the data records required for the data, control and status information is described in detail in subsequent sections of this specification.

Two additional models of the board are available with reduced feature sets. The VPC2020 eliminates all telephone functions and is not equipped with a plug-on telephone board. The VPC2040 does not have either a telephone board or speech compression. Application software has access to the board model number via the NULL function as described in section 2.1.1.

1.0 Hardware Considerations

The board hardware is in the form of a single IBM PC compatible board along with a piggyback board if telephone functions are installed (VPC2000 only). The physical interfaces for both software and analog connections are detailed below.

1.0.1 I/O Registers

The VPC presents a block of 16 contiguous I/O locations (registers) to the system processor. These are assigned the functions described below.

RELATIVE ADDRESS	FUNCTION
0	DATA REGISTER (READ ONLY)
1	STATUS REGISTER (READ ONLY)
2	DATA REGISTER (WRITE ONLY)
3	HARDWARE CONTROL REGISTER (WRITE ONLY)
4	RESET REGISTER (WRITE ONLY)
5	INTERRUPT VPC (WRITE ONLY)
6	INTERRUPT ACKNOWLEDGE (WRITE ONLY)
B	AUDIO CONTROL REGISTER (WRITE ONLY)

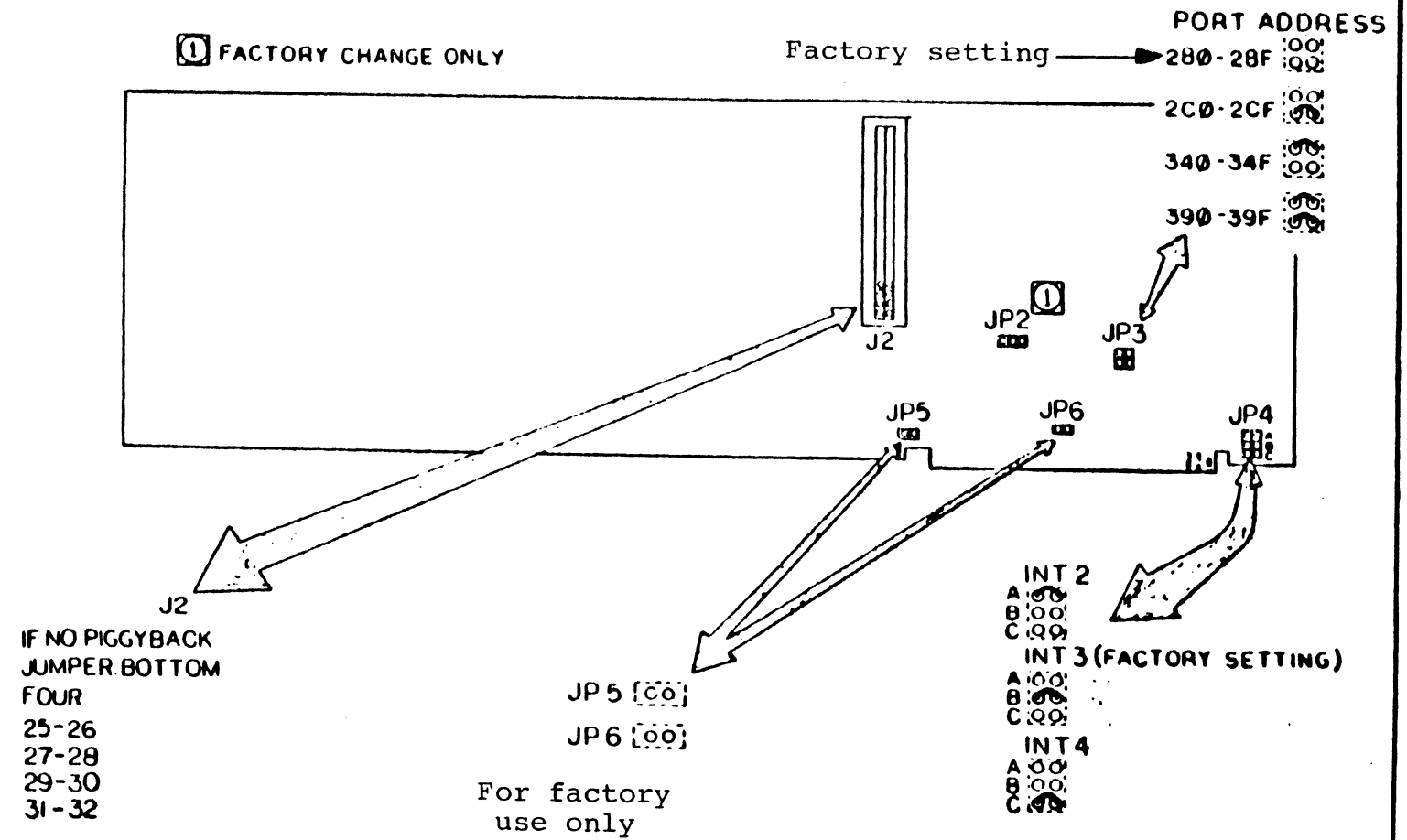
The start of this block of sixteen I/O locations may be any one of four jumper selectable base I/O addresses. These are 280, 2C0, 340 and 390 (all in hexadecimal). The base I/O address is selected by jumpers as shown in the board drawing in figure 1.1.

1.0.2 Programmed Reset

A write by the HOST processor to the RESET REGISTER location causes a reset of the VPC. The data associated with a write to this register is ignored by the board. This reset may be identical to that caused by activation of the INIT/ line on the backplane (a HARD RESET) or may cause the board to retain some variable information and current hardware state (a SOFT RESET). Prior to issuing a write to the RESET REGISTER the WRITE DATA REGISTER must be written with a byte of data to indicate the type of reset desired. Once a programmed reset is provided to the board, the VPC reads the byte of data present in the WRITE DATA REGISTER. If this byte is non-zero, the VPC assumes a soft reset and retains the appropriate variable information and telephone interface state. Otherwise all variable infomation is discarded and the telephone interface is placed on-hook.

1.0.3 Interrupt Acknowledge

A write to the Interrupt Acknowledge I/O location clears the interrupt latched on the board. This must be done once the interrupt for the board has been processed by the system and prior to starting another command. The data written to this register is ignored.



REV	ECO
X-1	
A	E0161
	08-07-84

VPC 2000 JUMPER/STRAP DWG

scale N/A

tol .xx N/A .xxx

Votan

size A

86600034

NXT Assy- 0100034-XXX

drwn by

date 08-08-84

dwg no 86600034

DWG. NO.	SH	REV.

FIG 1.1

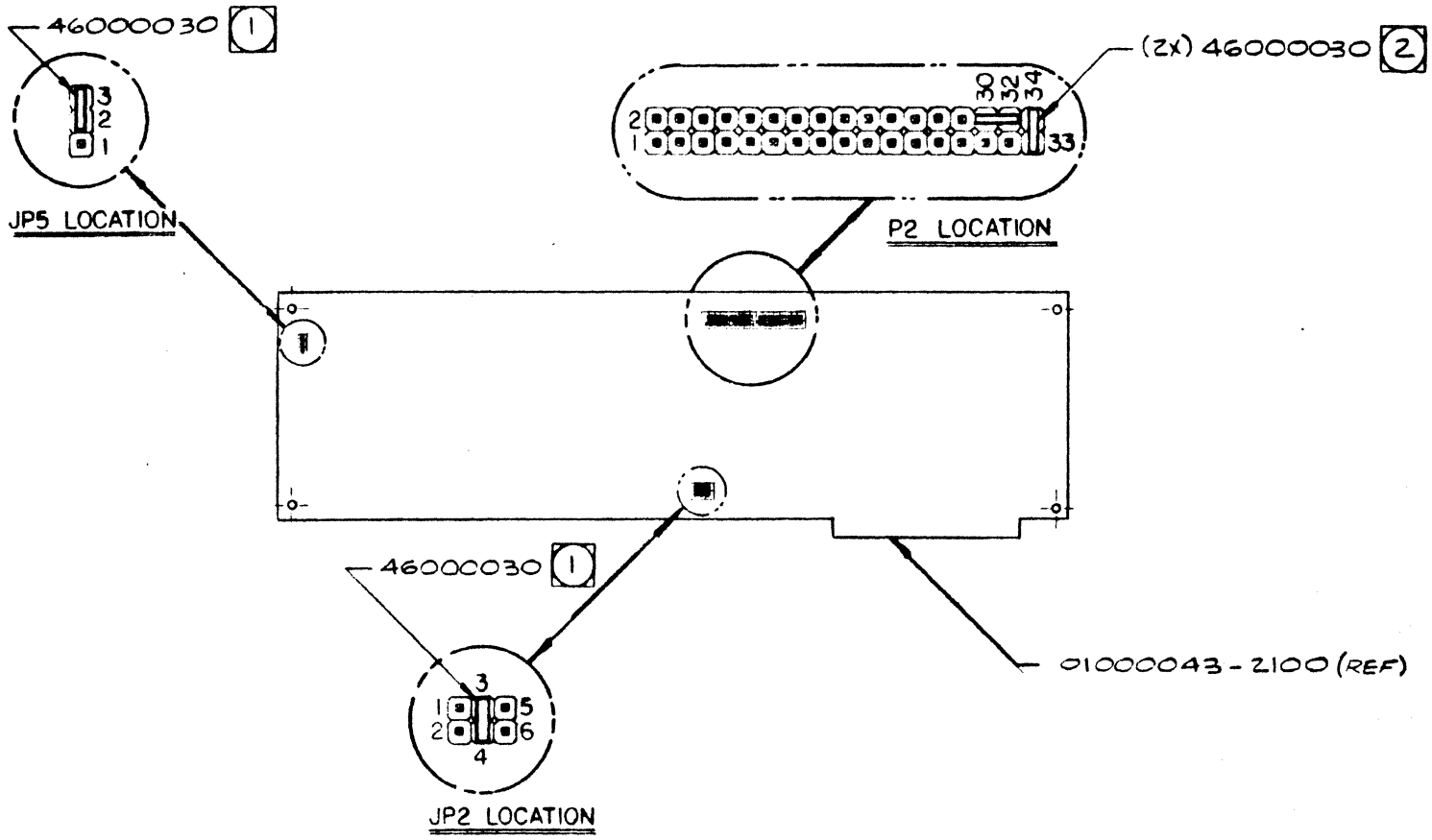


Figure 1.1a

NOTES:

(1) LOCATIONS JP2 & JP5 TO BE JUMPERED ON 01000043-2100 WITH AND WITHOUT PIB OPTION.

(2) LOCATION P2 TO BE JUMPERED AS SHOWN ON 01000043-2100, WITHOUT PIB OPTION ONLY.

REV	ECO
X1	1/30/85
X2	2/27/85

REF. VPC BOARD JUMPER / STRAP DRAWING

scale NONE | tol .xx ~ .xxx ~ | 1/31/85 | dwn by A. Velez Jr | date

Votan | sht 1 of 1 | NEXT ASSY: 01000043-2100 01000043-2300

86600043 | dwg no 86600043

REV | SH | DWG. NO.

1.0.4 Interrupt VPC

A write to this register interrupts the control processor on the VPC board. This is a standard part of the process of initiating a function on the board. If an operation is already in progress it will be aborted. The data written to this register is ignored.

1.0.5 Hardware Control Register

The data written to this register controls various elements of the hardware configuration of the board. The bits are assigned as follows:

BIT	FUNCTION
0	SPARE (MUST BE ZERO)
1	AGC DISABLE (IF = 1)
2	RESERVED (MUST BE ZERO)
3	INTERRUPT TRI-STATE
4	INTERRUPT DISABLE
5	RESERVED (MUST BE ZERO)
6	RESERVED (MUST BE ZERO)
7	RESERVED (MUST BE ZERO)

AGC DISABLE - This control bit has effect only if the telephone board is attached. It enables AGC circuitry for audio input. This AGC is an aid especially in the telephone environment where there is considerable variability between the audio level of telephone connections.

INTERRUPT TRI-STATE - When this bit is set the output of the interrupt is set to high impedance state. This allows the board to be used in a polling mode if an interrupt is unavailable.

INTERRUPT DISABLE - Masks the interrupt if set to a one.

1.0.6 Audio Control Register

This register allows software control of the available audio paths available with the telephone interface hardware. The following audio paths are available:

* Telephone Subscriber Loop (Tip/Ring)
* Telephone Set (Deskset) Interface
* Microphone and Speaker

The configurations possible and corresponding control codes are shown in figure 1.4.

Only the least significant four bits of this byte are used as described above. The remaining bits are reserved for future enhancements and must be set to zero.

1.0.7 Status Register

This read-only register provides status concerning the state of the read and write data registers and the VPC system interrupt. The bits are defined as follows:

BIT	MEANING
0	INDETERMINATE (MAY BE EITHER ZERO OR ONE)
1	INDETERMINATE (MAY BE EITHER ZERO OR ONE)
2	INDETERMINATE (MAY BE EITHER ZERO OR ONE)
3	SYSTEM INTERRUPT SET (IF = 0)
4	INDETERMINATE (MAY BE EITHER ZERO OR ONE)
5	INDETERMINATE (MAY BE EITHER ZERO OR ONE)
6	READ DATA REGISTER FULL (IF = 1)
7	WRITE DATA REGISTER FULL (IF = 1)

Bits 6 and 7 provide a means of synchronizing data transfers between the system processor and VPC. Bit 7 becomes active (1) when the system writes a byte of data to the WRITE DATA REGISTER and becomes inactive (0) when the VPC control processor reads the data. In order to prevent an overrun condition, the processor can sample Bit 7 until it goes to a zero prior to writing another byte.

When data is being sent from the VPC to the system, Bit 6 is set when the VPC control processor places data in the READ DATA REGISTER and is reset when the system reads this register. The system can sample this status bit to read a sequence of data bytes.

1.0.8 Interrupts

The VPC interrupt to the system processor (8088) may be set to one of three interrupt levels in the PC. They are:

PC Network Adapter choice
- ✓ INT2 - Color graphics interrupt
- ✓ INT3 - Comm2 interrupt
- INT4 - Comm1 interrupt

The factory setting for this interrupt is Interrupt 3. See figure 1.1 for the jumper selection of this interrupt.

1.0.9 Power Requirements

The VSP requires three voltage supplies at the tolerances and power levels given below.

+5 VDC (+10% , -10%) at 2.5 Amps (typical)
+12 VDC (+20% , -20%) at 0.08 Amps (typical)
-12 VDC (+20% , -20%) at 0.06 Amps (typical)

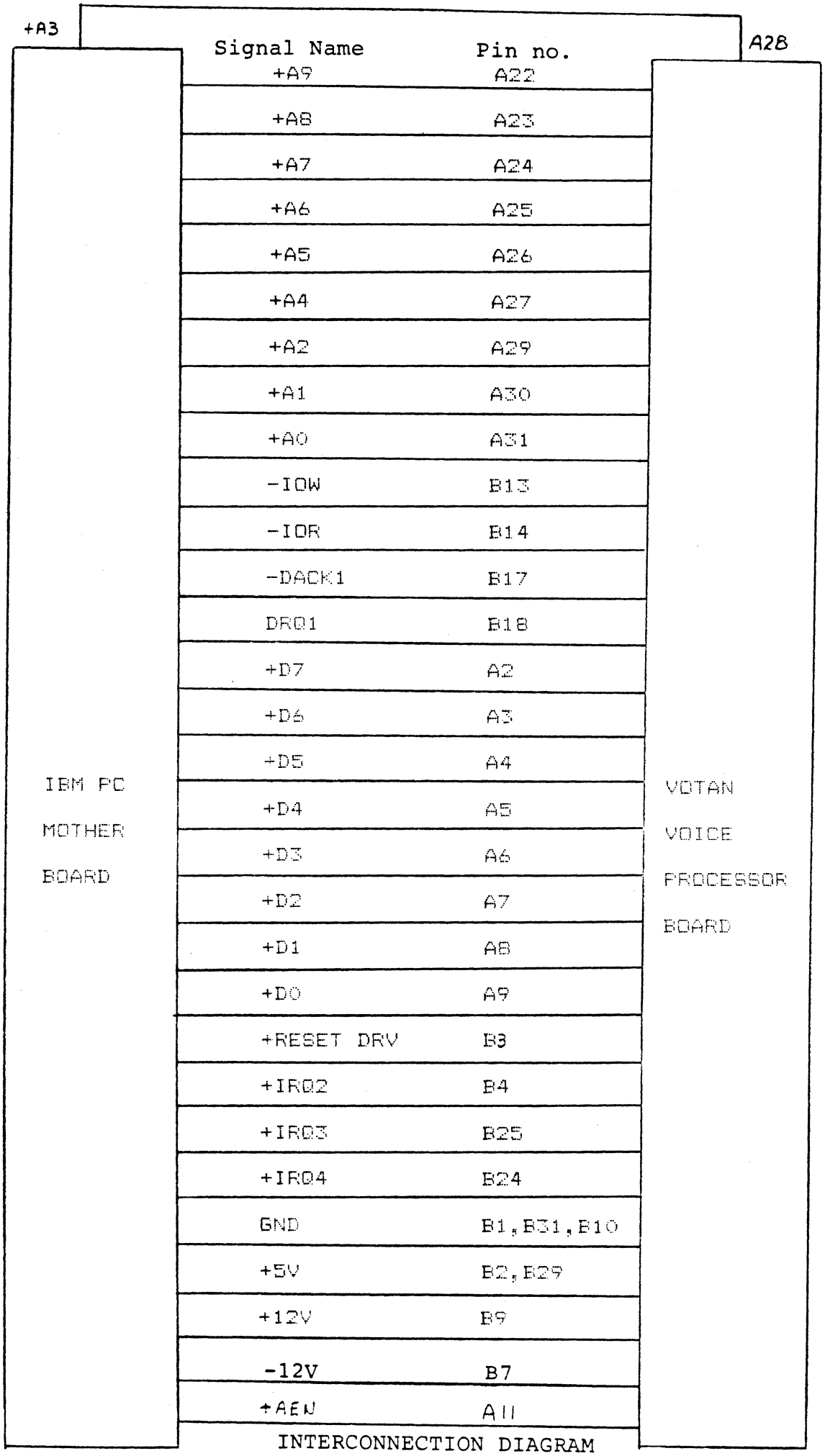
1.0.10 Audio Input and Output

The impedance and signal levels associated with the microphone and speaker audio I/O is given below:

Audio I/O	Max Level	Impedance
Microphone Input	1 mv. RMS	1k ohms
Speaker Output	2 watts	8 ohms

1.0.11 Digital and Power Connections

These pin assignments are compatible with the PC bus and are given in figure 1.2 for reference.



INTERCONNECTION DIAGRAM

FIGURE 1.2

1.1 Initialization

The board is initialized by either the hardware reset within the system or programmably by software. For a software reset the system should write appropriate initialization data values to the HARDWARE and AUDIO CONTROL REGISTERS. A byte must then be written to the WRITE DATA PORT to indicate whether to execute a hard or soft reset. Finally a data byte of any value is written to the INTERRUPT PORT. Note that the bytes written to the board may be written without respect to examining the WRITE BUSY bit since the VPC will be reset immediately following this sequence of actions. Since the reset operation takes some time to accomplish on the board, the software needs to wait a minimum of 100 microseconds before issuing another command to the board.

A hard reset clears all of the data on the board and puts the telephone interface on-hook (hangs up). The soft reset retains both the on-board data and the state of the telephone interface.

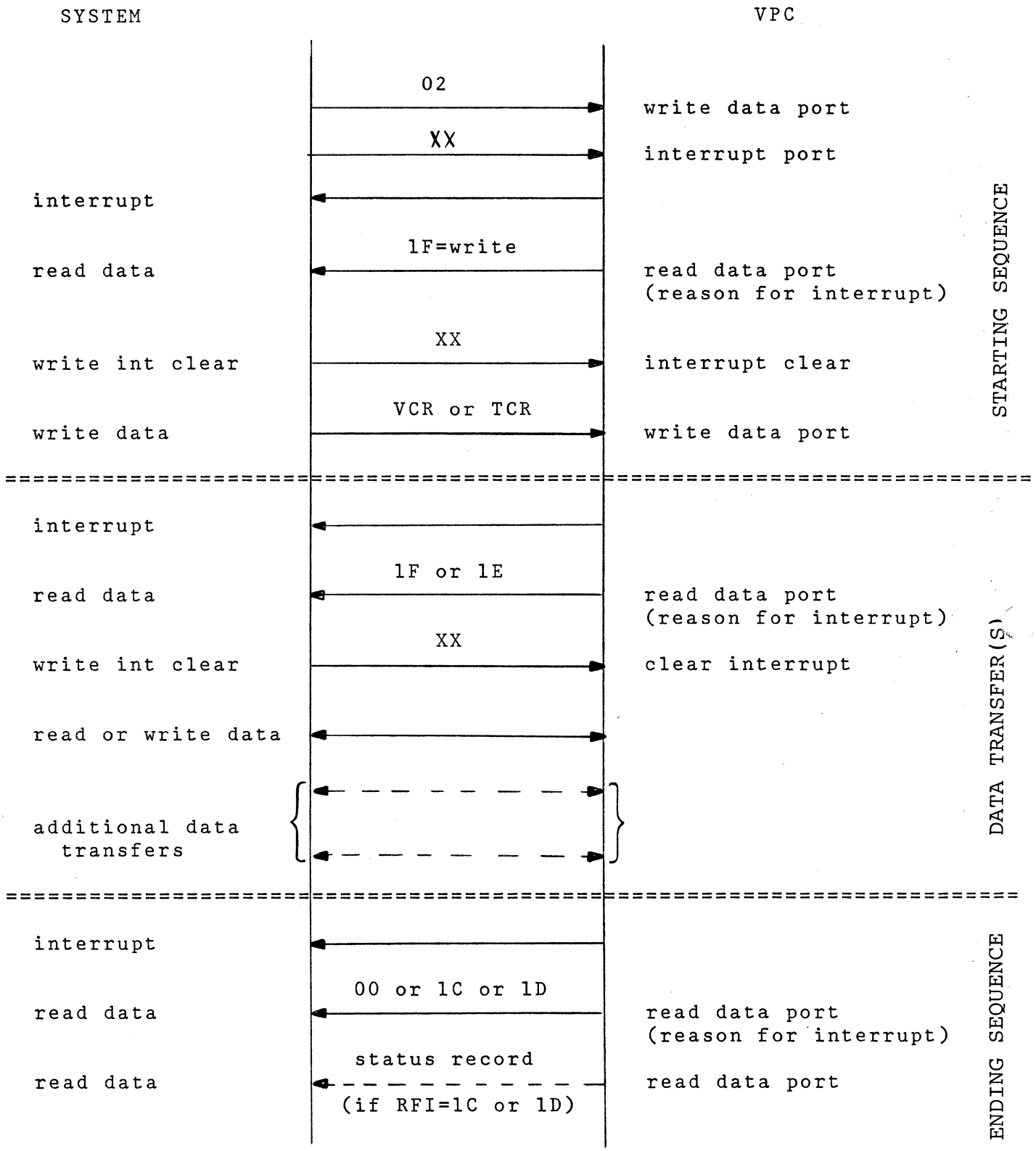
1.2 Command Sequence

The generalized sequence for executing a command to the VPC2000 consists of starting the function, transfering data to or from the board upon request and finally responding appropriately to the ending condition for the function. This ending condition may require reading an ending status record from the board upon command termination if the Reason for Interrupt (see section 1.4) is TERM or STAT. If the Reason for Interrupt is AOK then the function has completed normally with no further status to be transferred. For some commands no data transfer is required and only the command initiation and termination sequence is required. See figure 1.3 for a visual representation of a command sequence.

1.3 VPC Interrupt

In order to initiate any sort of action by the VPC the system must first write a command byte to the WRITE DATA PORT on the VPC followed immediately by a write to the INTERRUPT PORT. Only three valid values are defined for the command byte as listed below.

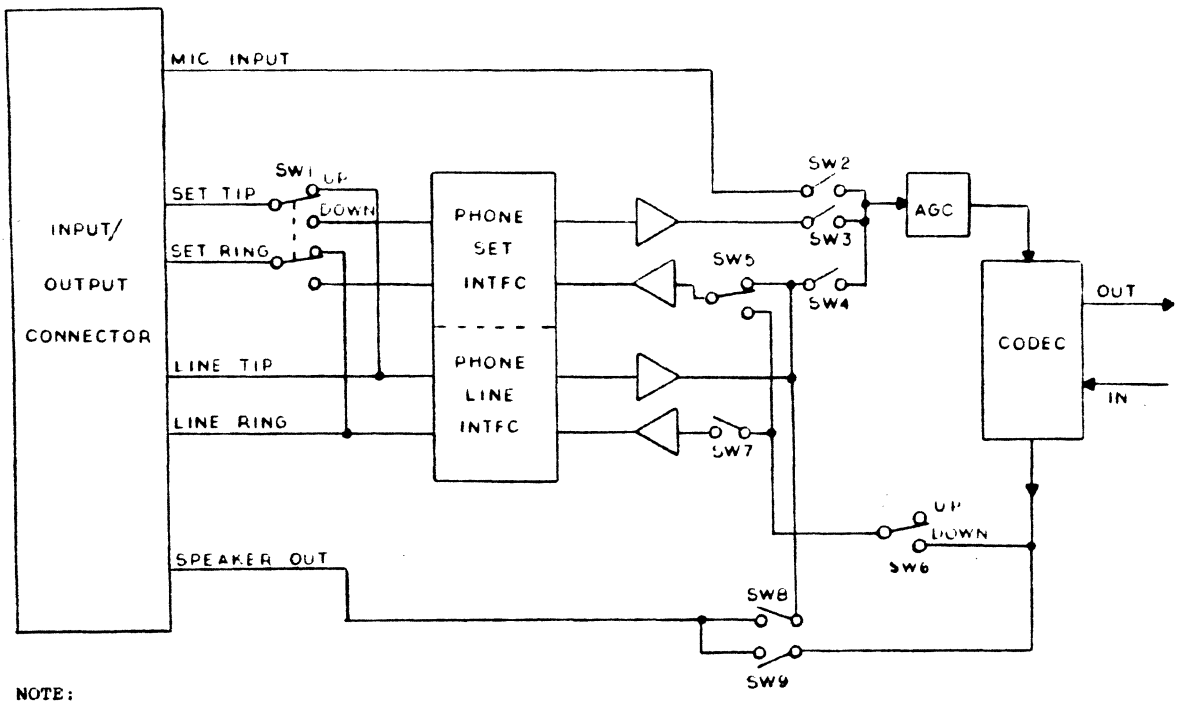
COMMAND	OPERATION
00	NOP
01	START VOICE OR TELEPHONE FUNCTION
02	ABORT CURRENT OPERATION



TYPICAL COMMAND SEQUENCE

FIGURE 1.3

CTLO				CORRESPONDENT SWITCH									DESCRIPTION
3	2	1	0	SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	
0	X	X	X	ON	OFF	ON	DOWN	UP	OFF	OFF	ON	UP	Normal operation
1	0	0	0	OFF	OFF	ON	UP	UP	OFF	ON	OFF	DOWN	Spare
1	0	0	1	OFF	OFF	ON	DOWN	UP	OFF	ON	OFF	DOWN	Standard telephone operation
1	0	1	0	OFF	OFF	ON	DOWN	DOWN	OFF	ON	OFF	DOWN	Handset voice I/O
1	0	1	1	OFF	ON	ON	DOWN	DOWN	OFF	OFF	ON	DOWN	Telephone speaker/mic I/O
1	1	0	0	OFF	ON	ON	DOWN	UP	ON	OFF	OFF	DOWN	Telephone voice I/O with speaker monitoring
1	1	0	1	OFF	OFF	OFF	DOWN	UP	OFF	ON	OFF	DOWN	Standard telephone operation with "silent on hold"
1	1	1	0	OFF	OFF	ON	DOWN	UP	ON	OFF	OFF	DOWN	Telephone voice I/O without speaker monitorong
1	1	1	1	OFF	ON	ON	UP	DOWN	OFF	ON	ON	DOWN	Spare



NOTE:

All switches shown in the OFF position.

FIGURE 14

1.4 System Interrupt

The VPC sets an interrupt to the system to request action. Prior to setting the interrupt the VPC loads a byte into the READ DATA PORT to describe the type of action required. This is referred to as the REASON FOR INTERRUPT byte and may have one of the values listed below.

HEX VALUE	NAME	MEANING
00	AOK	Command has completed, no status.
1C	TERM	Command has completed, system must read Status Record.
1D	STAT	Command is not complete, system must read Status Record.
1E	READ	Command is not complete, system must read data.
1F	WRITE	Command is not complete, system must write data.

1.5 Starting a Function

A function is initiated by writing a value of 01 (Start Function) to the WRITE DATA PORT followed by a write to the INTERRUPT PORT. Unless the VPC has detected an error condition it responds by setting an interrupt to the system with a Reason for Interrupt of WRITE. The system must then transfer the Voice Control Record (or Telephone Control Record) to the board.

The various parameters contained in the VCR are further described in section 2 along with the actions caused by each individual function. The same information related to the TCR is described in detail in section 3.

1.6 Data Transfers

Once a function has been initiated the VPC may use the interrupt mechanism to request that the system either read or write data records to the VPC. In these cases the REASON FOR INTERRUPT would be either READ or WRITE respectively.

Whenever a data item is described as a word this indicates a two byte entity. When a word of data is transferred between the VPC and the system processor the least significant byte is the first one transferred.

1.7 Data Record Format

The format of blocks of data (records) written to or read from the VPC is always identical. The first two bytes of the record give the length (n) of the entire record (including the first two bytes). The remaining n-2 bytes are the voice, control or status data. No record can be longer than 512 bytes including the two byte record length.

The template data conforms to the general requirements listed above and has within the template record two 16 bit elements which must be modified by the application when a template is generated. The general format is given below.

OFFSET	SIZE	MEANING
00	word	Record length
02	word	Word number
04	word	Template number
06	byte	Voice data
.	.	.
.	.	.
.	.	.

The word number and template number (at offsets 2 and 4) must be assigned by the application. The word number is generally a constant for a given spoken word. The template number would normally be different for each of multiple templates for a given word.

A second record type associated with recognition operations is the Template Length Record. Whenever templates are loaded to the VPC either by a LOAD function or in the coarse of a RECOGNITION function this record is transferred prior to the templates. The record consists of the lengths (in bytes) of all of the templates in the same order in which the templates will be transferred to the VPC. The form is therefor:

OFFSET	SIZE	MEANING
00	word	Record length
02	word	First template length
04	word	Second template length
.	.	.
.	.	.
.	.	.
nn	word	Last template length

1.8 Status Record

The status record varies depending upon the function performed.

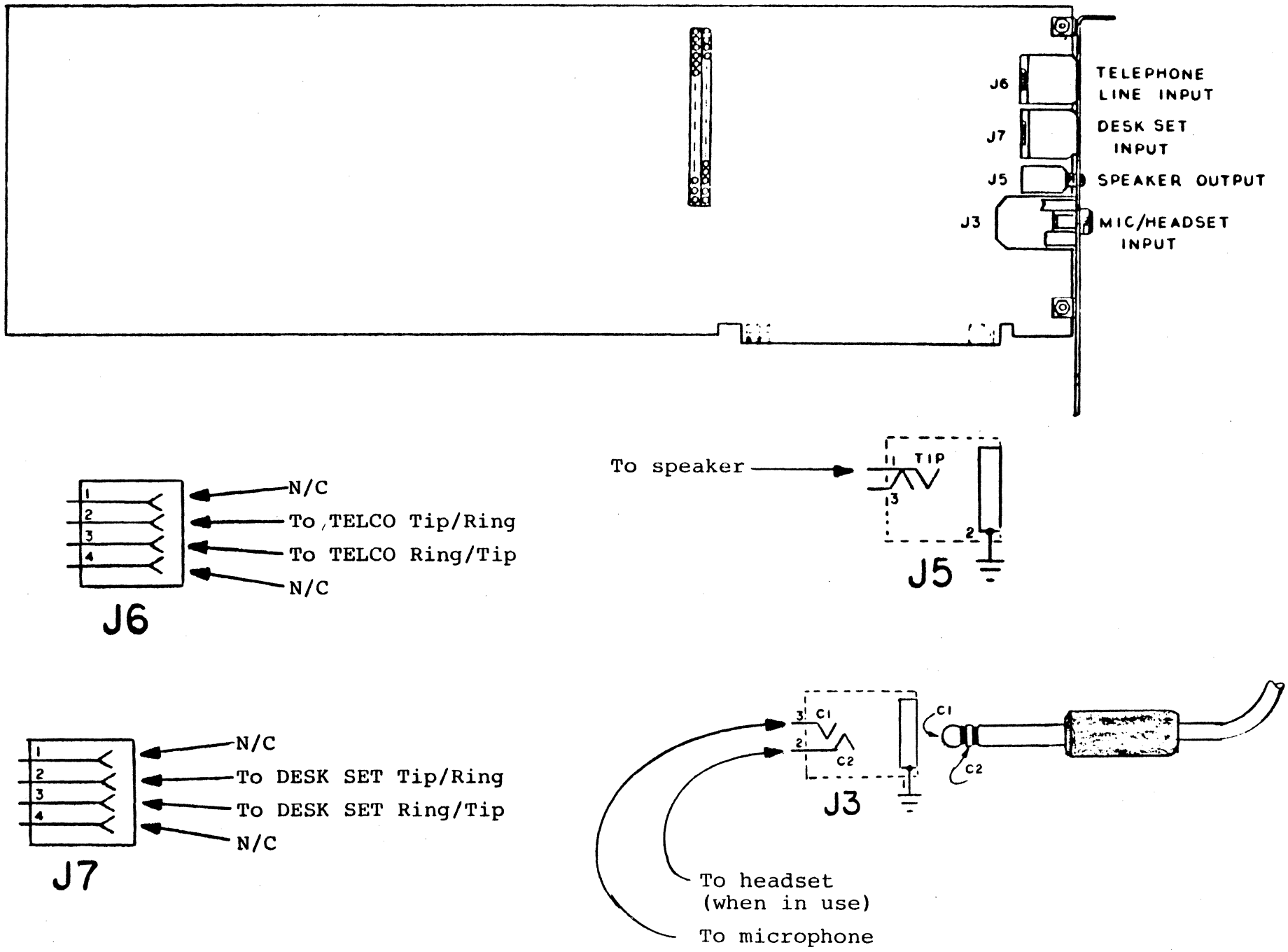

J6
TELEPHONE LINE INPUT
J7
DESK SET INPUT
J5
SPEAKER OUTPUT
J3
MIC/HEADSET INPUT
1
2
3
4
N/C
To TELCO Tip/Ring
To TELCO Ring/Tip
N/C
J6
To speaker
TIP
3
2
J5
1
2
3
4
N/C
To DESK SET Tip/Ring
To DESK SET Ring/Tip
N/C
J7
3
2
C1
C2
C1
C2
J3
To headset
(when in use)
To microphone

FIGURE 1.5

It is only read when the REASON FOR INTERRUPT is either TERM (1C) or STAT (1D). This status indicates that a function has been completed. The general form is given below.

OFFSET	SIZE	MEANING
00	word	Record Length
02	word	Ending Status
04	word	Data Return
.	.	.
.	.	.
.	.	.

In most cases the status record is only four bytes in length and contains only the record length and the ending status. The record may however be longer depending upon the function executed. The details of the form of the status for each available function is described further in section 2.1.

1.9 CSDR Recognition

CSDR (Continuous Speaker Dependent Recognition) function may return multiple recognition "events" for a single operation. To accomodate this an EVENT RECORD is provided. Several of these records generally are transferred from the VPC to the system during the course of the execution of a CSDR Recognize function. The form of the Event Record is as follows:

OFFSET	SIZE	MEANING
0	word	Record Length
2	byte	Event Number
3	byte	Template Number, Best Match
4	byte	Word Number, Best Match
5	byte	Distance, Best Match
6	byte	Template Number, Second Best Match
7	byte	Word Number, Second Best Match
8	byte	Distance, Second Best Match
9	byte	Word Start Index
10	byte	Word End Index
11	byte	Peak Amplitude

The EVENT NUMBER increments for each Event Record returned to the system and is used by the VPC for a REWIND function (see section 2.1.9). For extended CSDR recognition operation this Event Number will "roll over" from 255 to zero. The TEMPLATE NUMBER and WORD NUMBER in combination identify the particular template which was the best or second best match for a particular audio input. The WORD START and WORD END INDEX are used by the VPC for extracting embedded templates (see section

2.1.10). The peak amplitude gives a good indication of the audio input level and can be very useful for setting the input gain level. For good recognition results this amplitude should be in the range of B4 to E6 (hexadecimal).

1.10 Aborted Operations

When a command is aborted by interrupting the VPC with a value in the DATA WRITE PORT of 02 the VPC responds by setting an interrupt to the system. The REASON FOR INTERRUPT is usually TERM (1C) and the status record will be the four byte sequence 04h, 00h, 09h and 00h. This represents a record length of four and status representing NO VALID RESULTS. The status may be different depending upon the function in progress and the current state of the function when aborted.

Due to the asynchronous nature of aborting a function in progress the VPC may have been passing data to the system. Under these circumstances there may be one or more left-over bytes transferred prior to the System Interrupt and Status Record. This transfer of buffered information may be especially valuable during the ENCODE process as described in section 2.1.3. The NO VALID RESULTS status indicates that any data record which was being transferred when the function was aborted is incomplete and must be discarded.

II. Control Data Elements

The VPC2000 requires that very specific data records be exchanged between the board and the 8088 system processor. These records provide detail regarding command descriptions, status reporting to the system processor and voice data to be used during the various functions. The key data record associated with the function for voice operations is the Voice Control Record (VCR). A separate but similar data record provides function information for telephone operations and is referred to as the Telephone Control Record (TCR). The TCR is fully described in section 4.

2.0 Voice Control Record

The VCR consists of a sequence of bytes passed to the VPC2000 whenever a function is initiated. This data record contains the function definition, parameter setting information and (in some cases) a bit map describing the active vocabulary for a recognition function.

OFFEST DECIMAL)	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (bytes)
02	byte	Function code
03	byte	Flags
04	byte	Input Gain
05	byte	Output Gain
06	word	Byte Rate
08	word	Initial Timeout
10	word	Final Timeout (Word End Index/Acceptance Level)
12	.	Optional Bit Map
.	.	.
.	.	.

2.1 Function Code

This byte describes the basic function to be executed by the VPC. The basic function is subject to further modification by the flags and timeout values. The functions are listed below and described in more detail in the subsequent text.

Possible error status applicable to all functions and therefor not explicitly listed with each function are VPC Busy and Parameter Error. Any function is also subject to a status return indicating a Ring Detected if the appropriate flag bit

(Bit 6) was set when the function was initiated. See section 2.3 for a detailed description of ending status codes.

VALUE (HEXADECIMAL)	DESCRIPTION
00	Null Operation (NOP)
03	ISDR Train
05	Encode Message
06	Decode Message
07	Generate Tone
0B	ISDR Recognize
0E	Load Isolated Templates
10	CSDR Train
11	CSDR Recognize
12	CSDR Template Extract
13	Load Continuous Templates

2.1.1 Null Operation (NOP)

Returns the version and revision number of the board firmware and the model number of the board in the status record. No other effect.

Parameters used: None

Data Returned: Status Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word
04	byte	Control Processor Revision
05	byte	Control Processor Version
06	byte	DSP Revision
07	byte	DSP Version
08	word	Model Number

The designations "Control Processor" (CP) and "DSP" refer to the two processors on the VPC. Firmware is resident on the board for each of these processors. The VERSION and REVISION numbers are each interpreted as a binary number in the range of 00 - 99 rather than as two BCD digits. The model number is also returned as a binary equivalent of 2000, 2020 or 2040 in the 16 bit word designated above as Model Number.

2.1.2 ISDR Train

Causes the VPC to capture an ISDR (Isolated Speaker Dependent Recognition) utterance word from the incoming audio and transfer it to the system.

Parameters used: Flags: Bit 1 - Sample rate
Bit 6 - Enable State Change Int
Bit 7 - Enable DTMF Detect
Input Gain
Initial timeout

Data Returned: Template Record

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word
04	byte	Peak Amplitude
05	byte	Average Noise Level

Possible errors: Capture Buffer Overflow
Initial Timeout

2.1.3 Encode Message

Directs the VPC to encode (digitize and compress) the incoming audio and transfer the data to the 8088 processor. The BYTE RATE parameter provided to the VPC is the desired average data rate. The VPC keeps a moving average of the data generated by the compression process over a 240 ms. interval and adjusts the internal compression parameters accordingly to approximate as closely as possible the specified byte rate.

The operation may be terminated when the amount of silence specified by the FINAL TIMEOUT parameter is encountered. Under these circumstances all except 60 ms. of trailing silence is deleted from the data written. Likewise all leading silence greater than 60 ms. is deleted prior to the start of writing the data to system memory. If the FINAL TIMEOUT parameter is set at zero the VPC board will continue to encode audio indefinitely until the operation is terminated by a STOP command or by a PROGRAMMED RESET.

The ENCODE process may be terminated by the host processor at any time by issuing a STOP (ABORT) command to the board. Under these circumstances the board will complete the transfer of any partially transferred message records to the system prior to signalling completion with an interrupt. This means of terminating the ENCODE is recommended only when a transfer of data is not currently in process, i.e. it should be done prior to beginning a block read operation from the board.

Parameters used:
- Input Gain
- Initial Timeout
- Final Timeout
- Byte Rate
- Flags:
  - Bit 1 - Sample rate
  - Bit 6 - Enable State Change Int
  - Bit 7 - Enable DTMF Detect

Data returned: Compressed Speech Record(s)

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

Possible errors:
- DSP Overrun
- Initial Timeout

2.1.4 Decode Message

Directs the VPC to decode (decompress and convert to audio) the data supplied by the system. The operation can be terminated by the host processor at any time by issuing a STOP command to the VPC board.

Parameters used:
- Output Gain
- Flags:
  - Bit 1 - Sample rate
  - Bit 2 - Enable Voice Detect
  - Bit 6 - Enable State Change Int
  - Bit 7 - Enable DTMF Detect

Data supplied: Compressed Speech Record(s)

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

Possible errors: DSP Decode Error

2.1.5 Generate Tone

Causes the VPC to generate a 1 kHz tone with a duration of 0.125 second. This tone is useful for user prompting.

Parameters used: Output Gain

Data returned: None

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

2.1.6 ISDR Recognize

Captures an isolated word recognition input and matches it against previously trained recognition templates (see 2.1.2 for ISDR Train function). The recognition templates are either supplied to the VPC after the command is initiated or the command may use an already resident set of templates depending upon the state of Bit 5 of the FLAG byte in the VCR. See section 2.2.1 for a general explanation of this Flag bit option and section 2.2.7 for a description of the BIT MAP requirements.

Parameters used: Input Gain
Initial Timeout
Flags: Bit 1 - Sample rate
Bit 5 - Templates Loaded
Bit 6 - Enable State Change Int
Bit 7 - Enable DTMF Detect
Bit Map

Data supplied: Template Length Record (if Flag bit 5 = 0)
Template Records

Data returned: None

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=10)
02	word	Status Word
04	byte	Peak Amplitude
05	byte	Average Noise Level
06	byte	D1, Best Distance
07	byte	D2, Second Best Distance
08	word	W1, Best Word Number
10	word	W2, Second Best Word Number

Possible errors: Initial Timeout
Capture Buffer Overflow
Template Memory Overflow
Isolated Template Load Error

2.1.7 ISDR Load

This function loads templates for Isolated Speaker Dependent Recognition to the VPC. No voice function is performed.

Parameters used: None

Data Supplied: Template Lengths Record
Template Records

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

Possible errors: Template Memory Overflow
Isolated Template Load Error

2.1.8 CSDR Train

This function generates a template for a single word in the proper form for CSDR (Continuous Speaker Dependent Recognition) recognition operation. The template is transferred to the system processor. This operation is identical in form to the ISDR Train function, except that the template data placed in memory is different and is compatible with the CSDR Recog-

nize function.

Parameters used: Flags: Bit 1 - Sample rate
Bit 6 - Enable State Change Int
Bit 7 - Enable DTMF Detect
Input Gain
Initial timeout

Data returned: Template Record

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=6)
02	word	Status Word
04	byte	Peak Amplitude
05	byte	Average Noise Level

Possible errors: Capture Buffer Overflow
Initial Timeout

2.1.9 CSDR Recognize

This function is similar in many respects to the ISDR Capture and Match function. It initiates a Continuous Recognition Operation and either provides the templates to be used in the matching operation or not depending upon the state of Flag Bit 5. It varies significantly in that multiple recognition events may be returned to the system processor. See section 1.9 for a general description of the operation and event records returned.

Parameters used: Input Gain
Initial Timeout
Flags: Bit 0 - End on Pause
Bit 1 - Sample Rate
Bit 3 - Enable Rewind
Bit 5 - Templates Loaded
Bit 6 - Enable State Change Int
Bit 7 - Enable DTMF Detect

Data supplied: Template Length Record (if Flag bit 5 = 0)
Template Records

Data returned: Event Records

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

Possible errors: Template Memory Overflow
CSDR Template Load Error
Unresolved, Buffer Full
Initial Timeout

If the Least Significant Bit of the Flags byte is set to a logical one then the operation will automatically terminate after either 7.5 seconds of speech without a pause or after the end of a word when a "word end" time value is exceeded (End on Pause). This time is a constant value of 240 ms. and is not alterable by the system. This will generally result in several recognition events being passed to the system. These EVENT RECORDS then may subsequently be used to EXTRACT embedded templates from the original processed audio data which continues to reside on the board until another command is issued. See the description of the CSDR TEMPLATE EXTRACT command for a further description of this procedure.

When Bit 3 of the Flags byte is set to a logical one then the board will use the information available in the WORD END INDEX provided in the VCR (byte 10) to begin the current recognition operation using the buffered real-time audio data currently resident on the board. This allows the system to switch to a different portion of the same vocabulary or to a completely different set of vocabulary templates for subsequent recognition even in the midst of a continuous stream of speech input. This function is referred to as REWIND and can be thought of as being analogous to rewinding a tape recorder to the proper position to pick up audio input. In this particular case it is rewinding through a data buffer to accomplish the same function.

2.1.10 CSDR Template Extract

This function extracts a single embedded template from the buffered audio on the VPC. This command must only be used following a CSDR Match operation or another CSDR Template Extract operation. Any other preceding command will cause either an error to be flagged or erroneous results to be obtained with no error indication.

This command causes processed audio data in the on-board buffer of processed audio data to be transferred to system memory properly formatted as a CSDR Template Record. This data is identified by the EVENT RECORD passed to the board following the VCR transfer. A template of this sort can considerably improve recognition of words within an audio context of surrounding words. This is especially valuable for words which are both short in duration and apt to be said in strings such as the digits.

Parameters used: None

Data supplied: Event Record

Data returned: Template Record

Status Record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (=4)
02	word	Status Word

Possible errors: None

2.1.11 CSDR Template Load

This function performs the same function as ISDR Load except that CSDR templates are loaded.

2.2 Voice Function Parameters

Several parameters as described below control audio levels, timing and various other aspects of the voice functions. These parameters are sent in the VCR for each function initiated.

2.2.1 Flag Byte

This byte provides modifying data for the basic function being executed. The bits are assigned as follows:

BIT	MEANING (IF SET)
0	End on Pause
1	6kHz Audio Sample Rate (8kHz is default)
2	Enable Voice Detect
3	Enable Rewind Mode
4	Reserved (must be zero)

2.2.1 Flag Byte (con't)

BIT	MEANING (IF SET)
5	Templates Loaded
6	Enable State Change Interrupt
7	Enable DTMF Interrupt

The END ON PAUSE bit (bit 0) applies to the CSDR Recognize function. When set, it will cause the function to terminate when 240 millseconds of silence is detected after speech has been detected or after 7.5 seconds of speech without a pause. This mode is useful for creation of embedded templates (see section 2.1.10).

The sample rate selection bit (bit 1) is most useful for reducing the bit rate for intelligible compressed speech. The user can experiment with this control. The effect of a lower sample rate on the accuracy of recognition is currently unknown but the expectation would be for lower accuracy. The most recent version of firmware ignores this bit for all speech compression functions, flags an error if set to the lower rate for telephone operations and it's use is not recommended for recognition operations. As a result, future versions of code will ignore it entirely and automatically set the preferred sample rate for any given function. All software under development should therefore always set this bit to zero.

Bit 2 provides a Voice Detect option when set. This option is only applicable to a DECODE operation. When this option is enabled the VPC2000 looks at the input audio signal whenever silence is being decoded. Since normal speech patterns always have some silence within them this allows a user to "interrupt" a message being played by the board. This option is only available on boards with CP firmware of 3.00 or greater. See the description of the NULL command to determine how to read and interpret the firmware revisions.

The bit to enable REWIND for recognition (bit 3) allows a change of candidate recognition templates while the Digital Signal Processor (DSP) continues to process and buffer incoming audio. Once the candidate template set is restablished CSDR Recognition can resume at the precise time following a previously recognized word. The effect of this is that during a continuous stream of speech the candidate vocabulary can be changed without the user being being aware of this background activity.
Bit 4 is currently unused but is reserved for future expan-

sion. This bit must be set to a zero for compatability with any future enhancements utilizing this bit.

The TEMPLATES LOADED bit (bit 5) instructs the VPC to commence a recognition function using the templates already resident on the board. This eliminates the requirement to transfer a template set to the board and the board will make no requests for templates.

The ENABLE STATE CHANGE INTERRUPT bit (bit 6) causes the VPC to terminate any voice operation in progress and return a status value indicating a change of state on either the telephone line interface or the deskset interface. If a ringing signal is detected on the line interface then the RING DETECTED status is returned. If the handset switchook changes state then either the DESKSET OFF-HOOK or DESKSET ON-HOOK status is returned to indicate the current state of the deskset reciever. This option is applicable to all of the commands involving audio input or output, i.e. TRAIN, CAPTURE, ENCODE and DECODE operations. In addition it is applicable to the telephone command ANSWER. This option bit is only available on boards with CP firmware of 3.01 or greater. See the description of the NULL command to determine how to read and interpret the firmware revisions.

Bit 7 enables recognition of a DTMF tone during the performance of an audio input or output command. If a DTMF tone is recognized then the status returned is DTMF DETECTED. The code for the particular tone detected is returned in the low byte of an additional word added to a 6 byte status record. The higher order byte of the added word is set to zero. This option is only available on boards with CP firmware of 4.00 or greater. See the description of the NULL command to determine how to read and interpret the firmware revisions.

2.2.2 Input and Output Gain

Both the input and output gain are adjusted in 6db increments with parameter values 0 to 7. A parameter outside this range causes a parameter error to be generated in the status byte.

2.2.3 Byte Rate

The byte rate parameter adjusts the average quantity of data generated in the encoding process and is expressed in bytes per second. Byte rates are valid in the range of 500 to 2500 bytes per second.

2.2.4 Initial and Final Timeout

These are time values expressed in 0.01 second increments. Initial timeout refers to the time between the initiation of the function and detection of any significant audio input (i.e., not judged to be silence) by the VPC board. If the Initial Timeout value is set to a non-zero value, then the VPC terminates the operation and returns a status byte of Initial Timeout if the period of silence exceeds this amount of time. The Final Timeout refers to the maximum duration of silence that will be allowed once "non-silent" audio has been detected. When the Final Timeout value is non-zero then a timeout will occur if a silence period equal to or exceeding this period is detected. The operation is then terminated and successful completion status is returned.

The Initial Timeout parameter is applicable to the ISDR Train, Encode Message, ISDR Recognize, CSDR Train, and CSDR Recognize functions.

The Final Timeout parameter is applicable only to the Encode voice function. For this function all trailing silence in excess of 60 ms. is deleted from the compressed speech data transferred to system memory.

2.2.5 Acceptance Level

This byte indicates to the VPC the maximum acceptable DISTANCE measured during CSDR recognition operations. The DISTANCE is a measure of the closeness of fit between a stored template and the incoming audio. For discreet word recognition operations, the system must make all acceptance level determinations. This value must be placed in byte 11 of the VCR.

Any value of acceptance above 70 will cause increased response time for CSDR Recognize operations due to increased processing requirements for the VPC control processor.

Determination of recognition acceptance for ISDR must be made by the system processor. The VPC returns the distance values for this purpose. 2.2.6 Word End Index

This data item provides a reference value for the extraction of an embedded template from the compressed audio data buffered on the VPC following a CSDR Recognize command. This value is found in byte 10 of each event record and must be placed in byte 10 of the VCR to rewind to the end of this recognition event.

2.2.7 Recognition Bit Map

The transfer of recognition templates to the card may be accomplished by using either the ISDR or CSDR load commands or by transferring the necessary recognition templates to the VPC when the recognition command is issued. In either case the the application must take note of the order in which templates are transferred to the board in order to utilize the capability of identifying a subset of the total template resident on the board for the recognition operation. Using only a subset of the total vocabulary available provides faster response times and greater recognition accuracy in most applications.

The bits in the Bit Map are assigned according to the order in which the templates were transferred to the VPC. The convention is that a bit set in the least significant bit of the first byte of the map indicates that the first template transferred to the VPC is to be a candidate template for the recognition function. The order proceeds from least significant bit of the least significant byte (first Bit Map byte tranferred) to the most significant bit of the most significant byte (last Bit Map byte transferred). The Bit Map must be long enough to reference all of the templates transferred to the VPC. The length of the Bit Map is indicated by adding to the record length of the VCR so that it includes the Bit Map.

Since templates cannot be added to those already resident on the VPC a completely new set of templates must be loaded if any desired templates are not resident for a particular recognition operation.

A zero length Bit Map indicates that all resident templates are to be active for a recognition operation.

2.3 Status Byte

Following either voice or telephone functions a word byte of status is returned by the VPC in the Status Record. Only the least significant byte of the word reserved for an ending status code is actually used. The more significant byte of this word is always zero. The valid ending and intermediate status values are given below.

	VALUE	MEANING
	00	Successful Completion
	01	(Unused)
	02	(Unused)
	03	Parameter Error
	04	(Unused)
	05	Capture Buffer Overflow
	06	VPC Busy Error
	07	DSP Overrun
	08	Initial Timeout
	09	No Valid Results/Invalid Data
	0A	DSP Decode Data Error
	0B	Ring Detected
	0C	DP Memory Full
	0D	Template Load Error
	0E	CSDR Word Buffer Overrun
	0F	Unused
	10	Voice Detected
	11	DTMF Detected
	12	Deskset Off-Hook
	13	Unused
	14	Function Not Available
	15	Deskset On-Hook
*	20	Dial Tone Timeout
*	21	Ringback Timeout
*	22	Voice Detect Timeout
*	23	Busy Signal Detected
*	25	Ring Timeout
*	26	Control Sequence Error
*	27	Data String Error
*	28	Detone Timeout
*	29	Termination Tone Detected

* = Status for Telephone Function Only (see section 3)

2.3.1 Successful completion

The requested operation has been completed without any errors encountered. Any data required to be placed in the system

memory by the operation now resides in the data area(s) specified.

2.3.2 Parameter error

This error is caused by an invalid value in the VCR, TCR, or template data when an operation is started. The causes of this error are listed below.

VCR Parameter Errors

1. Illegal function code.
2. Illegal gain (>7).

TCR Errors

1. Illegal function code.
2. 6Khz sample rate specified for Dial, Entone or Detone commands.
3. Illegal gain.

Template Errors

1. Template record length not a multiple of 8 (ISDR) or 9 (CSDR).
2. Template record length less than 32 or more than 512.
3. Length of templates transferred doesn't match lengths specified in template parameter block.

2.3.3 Capture buffer overflow

This error is indicated when audio data into the VPC exceeds that which can be processed and buffered by the board when in isolated word recognition modes. This will occur if the audio level is such that no silence exists.

2.3.4 DSP overrun

In CSDR it means that the board has been unable to process the data fast enough to keep up with the audio input. This typically can happen if a large number of templates are being used and the spoken input continues for a relatively long period of time (more than twenty seconds) without a pause. This can also be caused by the same non-silent audio input as described for the Capture Buffer Overflow described above.

2.3.5 VPC Busy error

This error is a result of the host issuing a new command request to the VPC before the previously requested command is completed. It indicates a sequencing error from the host.

2.3.6 Initial timeout

Indicates that no audio input above a silence level was input to the system between the time a command was issued to the system and the time specified by the Initial Timeout interval was completed.

2.3.7 No valid results/Invalid data

NO VALID RESULTS is indicated following a STOP command to the board when a recognition or train command has not had time to complete. It signifies that no results from the recognition operation have been passed to the system processor.

INVALID DATA is indicated following an attempted load of template data to the board if the templates loaded are of the wrong type (CSDR templates for an ISDR command for example) or if the number of templates loaded do not match the number of templates implied by the previously loaded template parameter block.

2.3.8 DSP decode data error

Indicates that invalid data was detected by the VPC during decoding of compressed speech. Specifically this is the detection of invalid information embedded with the message data. This can be caused by invalid data present in the system memory, hardware difficulties which alter data while being read from memory and potentially a variety of other system causes.

2.3.9 Ring detected

Indicates that an incoming call on the loop interface has been detected. This feature must be enabled by setting the VCR flag byte, Bit 6 to enable an interrupt if an incoming call is detected. Any voice command that might be in progress is terminated. Any data which may result from an active voice command is invalid.

2.3.10 DP Full

This error indicates that either an attempt was made to either load over 150 recognition templates to board or that the templates required more than the 22k bytes of on-board memory space available for template storage.

2.3.11 Template Load Error

This error is returned when no templates have been loaded to the board and a recognize command is initiated with the TEMPLATES LOADED flag bit set.

2.3.12 CSDR Word Buffer Overrun

This status may occur in the CSDR Recognize function. If bit 0 of the Flag byte is set (End on Pause) it will occur if speech (or noise) of over 7.5 seconds occurs without a pause. If the above mentioned flag bit is not set, it occurs if for any reason the 7.5 second on-board speech buffer is exceeded. This can occur if the maximum number of templates are active during the recognition operation and a user speaks for a long period (generally more than 15 seconds) without pausing. It may also occur if a REWIND function is in progress with delays before restarting the recognition due to system operation, template loading, etc.

2.3.13 Voice Detected

This status indicates that voice input was detected during the execution of a DECODE command. See section 2.2.2 concerning the FLAG byte options for a further description of the conditions under which this status can be returned.

2.3.14 DTMF Detected

This status indicates that a DTMF Tone was detected during the execution of a voice command. See section 2.2.2 concerning the FLAG byte options for a further description of the conditions under which this status can be returned.

2.3.15 Deskset Off-Hook

Indicates that a change in switchhook status from on-hook to off-hook was detected for the deskset. See section 2.2.2 concerning the FLAG byte options for a further description of the conditions under which this status can be returned.

2.3.16 Function Not Available

This status is returned if a function is attempted on a VPC2020 or VPC2040 which is not installed on that board model. See section 1 for a description of the functions available for these models.

2.3.17 Deskset On-Hook

Indicates that a change in switchhook status from off-hook to on-hook was detected for the deskset. See section 2.2.2 concerning the FLAG byte options for a further description of the conditions under which this status can be returned.

III. Telephone Operations

The VPC may be equipped with an FCC approved telephone interface (TIP and RING). This allows attachment either directly to central office facilities or to a PBX. The VPC telephone command set allows it to both initiate and receive calls, and provides most functions available through PBX equipment that uses a "switch-hook flash". An auxiliary input is available which allows the attachment of a standard telephone deskset. This deskset can be used in its normal function as a standard telephone, can be used to monitor the line while VPC telephone operations are in progress or can be used for audio I/O to the VPC. These modes of operation are all under software control.

3.0 Telephone Control Record

The telephone functions are controlled by means of a telephone control record (TCR) which takes the place of the VCR for phone functions. The contents of the TCR are listed below.

OFFSET (DECIMAL)	ELEMENT SIZE	DESCRIPTION
00	word	Record Length (bytes)
02	byte	Function Code
03	byte	Flags
04	byte	Input Gain
05	byte	Output Gain
06	byte	Detone Timeout
07	byte	Ring Count (Answer)
08	byte	Answer Delay
09	byte	Dial Tone Timeout
10	byte	Ringback Timeout
11	byte	Voice Detect Timeout
12	byte	Ring Count (Dial)
13	byte	Detone Termination Character
14	byte	Detone String Length
15	byte	Not Used

3.1 Function Code

A single byte function code provides the basic command type to the VPC. These codes are listed in the table below and subsequently described in detail. The possible errors associated with each command is detailed in section 3.4.

BYTE VALUE (HEXADECIMAL)	DESCRIPTION
20	Onhook
21	Offhook
22	Dial
23	Answer
24	Entone
25	Detone
26	Set Parameters
27	Flash
28	Telephone Monitor

3.1.1. Onhook

This function closes the telephone line. This is the equivalent of "hanging up" a telephone (hence the terminology). The actual physical effect is the opening of the switch for the line. If the line is already on-hook this command has no effect.

Parameters used: None

Data Returned: Status Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

3.1.2 Offhook

This is the opposite of ONHOOK. It closes the switch for the line and is the equivalent to lifting the receiver on a telephone. If the line is already off-hook this command has no effect.

Parameters used: None

Data returned: None

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

3.1.3 Dial

This function initiates a call. The string specifying the number to be dialed is supplied by the system processor as a data record. It is inspected for validity and the line is then taken off-hook and the VPC waits to detect a dial tone. Once the dial tone is detected, the supplied data stream is used to dial a telephone number. In addition to digits representing a telephone number or extension, the string may contain special codes which alter the mode of dialing (either DTMF or PULSE dialing) as well as codes which specify pauses in the dialing sequence to either wait for another dial tone or for a specific time interval. The call progress is then monitored by detection of the familiar "call progress tones" such as BUSY and RINGBACK. If BUSY is detected the call is immediately terminated (the line is placed on-hook) and the BUSY status is set in the status byte. When the telephone is answered, it is assumed that a person will speak into their telephone ("HELLO" is most common). The VPC detects this audio and sets the successful completion status. Details regarding the requirements for the dialing string are covered in section 3.5.

Throughout the calling sequence, several timeout and count parameters are monitored. These parameters as well as additional parameters to tailor telephone operation to the needs of the particular operating environment are available through the parameters detailed below as well as the SET PARAMETERS command.

Parameters used: Input and Output Gain
Dial Tone Timeout
Ringback Timeout
Voice Detect Timeout
Ring Count (Dial)

Data supplied: Dial String Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: Dial Tone Timeout
Ringback Timeout
Voice Detect Timeout
Busy Signal Detected
Ring Timeout
Data String Error

3.1.4 Answer

The answer function monitors the ring indicator and closes the hook switch (goes off-hook) when the specified number of ringing cycles for an incoming call are detected. Following the specified answer delay the VPC signals completion with appropriate status and an interrupt. The answer function may be cancelled by the on-hook function. This function can be initiated with the ENABLE STATE CHANGE bit (Bit 6) set in the Flag byte. The response to the ring indicator will follow the normal answer delay parameter under this circumstance.

The answer function may also be invoked by a flag bit in the VCR When the ENABLE STATE CHANGE bit is set, the control processor will monitor the ring indicator as well as the deskset switchhook status while performing voice functions. If a ring occurs, the VPC interrupts the system processor with the ring detected status. The system processor can then respond appropriately.

Parameters used: Ring Count (Answer)
Answer Delay

Data returned: None

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

3.1.5 Entone

This command sends the supplied dialing string over the already opened (off-hook) telephone line. The string format is described in section 3.5. The ASCII representation of the 16 tones is given in section 3.6.

Parameters used: Output Gain

Data supplied: Tone String Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: Control Sequence Error
Data String Error

3.1.6 Detone

The DETONE function monitors the line, decodes the received DTMF codes into ASCII, and passes them to the system using the standard data record format. The detone function continues until one of the conditions listed below is met. The ASCII representation of the 16 codes is given in section 3.6 and the string format is described in section 3.5.

- The requested number of tones has been received.
- A tone matching a designated termination tone is received.
- The Detone Timeout is met.

Parameters used: Input Gain
Detone Termination Character
Detone String Length
Detone Timeout

Data returned: Tone String Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: Control Sequence Error
Detone Timeout
Termination Sequence Detected

3.1.7 Set Parameters

This function permits the user to change the time intervals for DTMF tone generation and switchhook flash to values other than the default ones supplied by the VPC. The desired values are assembled into a data record (described in section 3.7) and passed to the VPC upon request. The VPC will then use the data to make the appropriate changes in its internal parameters. Modification of the time interval values allows adjustment to be compatible with various telephone switching equipment. Since the default progress tone frequencies are those specified for the telephone system in the United States, this feature is primarily intended to readily adapt the system for use in other parts of the world.

Parameters used: None

Data supplied: Parameter Record

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

The default values are stored in the DSP data area and are re-established any time the telephone code set is invoked after a non-telephone function, or after a reset. The default values for these parameters are given in section 3.7.

3.1.8 Flash

This function performs a "switch hook flash" using the current parameter setting to determine the time duration of the flash. This function allows access to features found on many PABX systems.

Parameters used: None

Data supplied: None

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

3.1.9 Telephone Monitor

This function is provided to wait for a status change on the deskset or line interface. An interupt is generated if either a ring is detected on the line or the hook status of the deskset changes. There are no timeout parameters associated with this command but it may be terminated with a stop command. The status byte for a state change may be either Ring Detected, On-hook or Off-hook.

Parameters used: None

Data supplied: None

Status record:

OFFSET	ELEMENT SIZE	DESCRIPTION
00	word	Record Length
02	word	Status Word

Possible errors: None

3.2 Flags

Two flag bits are defined for the telephone functions, as follows:

Bit 0: Unused (may be any value)
Bit 1: 6 kHz Audio Sample Rate (8 kHz default)
Bit 2: Reserved (must be zero)
Bit 3: Unused (may be any value)
Bit 4: Reserved (must be zero)
Bit 5: Unused (may be any value)
Bit 6: Enable Pulse Dialing/Enable State Change Int
Bit 7: Disable Voice Detect Timeout

The 6 kHz AUDIO SAMPLE RATE option (bit 1) should never be excercised (should always be set to zero) for telephone

operations. It will cause a parameter error to be flagged by the VPC. This option bit will be deleted in future releases.

The PULSE DIAL ENABLE bit (bit 6) causes the board to use pulse (rotary) dialing mode of operation for originating calls rather than DTMF tones (the default mode). Note duplicate usage in paragraph below.

The DISABLE VOICE TIMEOUT bit (bit 6) disables the call termination if neither a ringback or voice is detected while dialing a call. This is primarily used for test purposes when setting up an application. This same bit is used for an entirely different effect if used in confunction with the ANSWER function. In this context it allows an interrupt to be set if a change in the deskset switchhook status is detected. If such a change is detected, the status returned from the board is either DESKSET ON-HOOK or DESKSET OFF-HOOK depending upon the state of deskset reciever. Note duplicate usage in paragraph above.

3.3 TCR Parameters

Several entries in the TCR are parameters for telephone operations as described below.

3.3.1 Input Gain

The input gain parameter is always set by the firmware to a minimum gain level of 4 which generally gives the best results for dialing operations. This value may increased by inserting an input gain value of greater than 4 in the TCR.

3.3.2 Output Gain

The output gain for generation of touchtones in either the DIAL or ENTONE commands is fixed by the firmware at a gain level of 2 and is not affected by the value in the TCR.

3.3.3 Detone Timeout

The detone timeout parameter specifies in tenths of seconds the length of silence required to end the detone function. If no tones are detected for this length of time, a Detone Timeout status is returned.

3.3.4 Ring Timeout

The ring timeout parameter specifies the numbers of rings allowed before going off-hook and completing the answer func-

tion. This feature allows implementation of features found on automatic telephone answering equipment. Specifically, it allows the local user to answer the call rather than have it automatically answered and also allows the user to avoid telephone charges if no messages exist when calling from a remote location.

3.3.5 Answer Delay

The answer delay specifies in tenths of seconds the length of time between the line being taken off-hook and the VPC interrupt to the system.

3.3.6 Dial Tone Timeout

The dial tone timeout specifies in tenths of seconds the waiting time for a dial tone either after the line is initially taken off-hook or after the 'W' parameter in the call string.

3.3.7 Ringback Timeout

The ringback timeout parameter specifies in tenths of seconds the maximum time between the completion of dialing a number and the detection of a ringback tone. If no ringback is detected, the call is terminated on-hook with the no ringback status.

3.3.8 Voice Detect Timeout

The voice detect timeout specifies in tenths of seconds the length of time the VPC waits for a voice response after end of ringback if voice detect timeout (Flag bit 7) is enabled.

3.3.9 Ring Count

The ring count parameter specifies the number of rings the VPC waits for the ring back to stop after detecting the ringback tone during the dial function.

3.3.10 Detone Term Character

This byte specifies a terminating character which can end the DETONE function. If during the course of executing the DETONE function the VPC detects a received code that matches the character in this byte it terminates the command and returns Termination Tone Detected status. This tone as well as those preceding it are transferred to the system in a Tone String

Record.

3.3.11 Detone String Length

Indicates the maximum number of tones to be detected before terminating a DETONE function.

3.4 Status Byte

Many of the states of the status byte are related only to telephone operations and the meaning of these states are described below.

3.4.1 Dial Tone Timeout

This error status indicates that no dial tone was detected within the specified dial tone timeout period following taking the line off-hook. The line is placed back on-hook.

3.4.2 Ringback Timeout

This error status indicates that no ringback signal was detected within the timeout period following the last dialed digit when an attempt was made to initiate a call. The line is placed on-hook.

3.4.3 Voice Detect Timeout

This error status indicates that neither a call progress tone nor voice has been detected for the specified timeout interval. For this timeout period to be active a ringback tone must previously have been detected. The line is placed on-hook. This timeout may be disabled by setting bit 7 in the Flag byte.

3.4.4 Busy Signal Detected

This status indicates that a busy signal was detected and the line was placed on-hook.

3.4.5 Ring Timeout

This status indicates that the number of rings specified in the ring count has occurred since the first detection of a ring signal. The line has been placed on-hook.

3.4.6 Control Sequence Error

This error indicates that a function was requested that was impossible to execute given the current state of the VPC. An example of this would be a request to detect DTMF tones while the telephone line is in the on-hook state.

3.4.7 Data String Error

This error indicates that the data string supplied to the board specifying the number to dial (as well as intermediate functions) was found to be in error. The string is checked prior to initiating any actions with the telephone line.

3.4.8 Detone Timeout

This status indicates that the time between either the initiation of the DETONE function or the last detected DTMF tone equals or exceeds the specified timeout value. The line is left off-hook.

3.4.9 Termination Tone Detected

A DTMF tone matching the specified termination tone was detected.

3.4.10 Deskset On/Off-Hook

Either of these status conditions may occur if the Telephone Monitor function is executed. See section 2.3 for a further description.

3.5 Data String Description

The Tone and Dial Strings used by the VPC are of two types depending upon the function being performed. In both cases the valid character set consists of ASCII characters. The character sequence is transferred to or from the board using the standard data record format beginning with a record length word to define the total number of characters. For the ENTONE and DETONE commands the valid character set consists a standard character set representing the 16 standard DTMF tones as given below.

"0123456789" "ABCD" "*#"

For the DIAL command the additional characters "P", "W", "R" and "T" are valid.

The "W" in the dialing character string indicates to the board that it should wait for a dial tone before proceeding. If no dial tone is detected within the interval specified in the Dial Tone Timeout parameter, the call is terminated and the Dial Tone Timeout status is returned.

The "P" in the dialing sequence causes the VPC to pause for the specified time interval prior to proceeding with the dialing sequence. This is a method of dealing with delays introduced by various types of telephone switching equipment. The time interval is defined by the next character in the sequence (which must be a digit) and indicates the delay in tenths of seconds.

The "T" causes subsequent digits in the string to be dialed using DTMF tones and the "R" causes digits to be dialed using rotary (pulse) dialing. This feature allows a mix of dialing modes used in a single DIAL command.

3.6 DTMF Tones

A DTMF tone includes one frequency from each group of high and low frequencies given below as ROW and COLUMN frequencies. The minimum duration for detection is 40 milliseconds minimum and 40 milliseconds minimum between tones. Frequency tolerance for all tones is 1.5% + 2 Hz. The tone frequencies and ASCII representations are given below.

ASCII CHARACTER	ROW	COLUMN
1	0	0
2	0	1
3	0	2
4	1	0
5	1	1
6	1	2
7	2	0
8	2	1
9	2	2
0	3	1
*	3	0
#	3	2
A	0	3
B	1	3
C	2	3
D	3	3

ROW	FREQUENCIES
0	= 697 Hz
1	= 770 Hz
2	= 852 Hz
3	= 941 Hz

COLUMN	FREQUENCIES
0	= 1209 Hz
1	= 1336 Hz
2	= 1447 Hz
3	= 1633 Hz

3.7 Parameter Record

The data array contains parametric information to tailor key time intervals of the VPC board to accommodate as much as possible the characteristics of telephone systems world wide. The default values are appropriate for the United States.

OFFSET	SIZE	MEANING
00	word	Record Length
00	word	Entone Time Interval (both on and off)
01	word	Switch Hook Flash Interval
02	word	Ring ON time, milliseconds
03	word	Ring OFF time, milliseconds

The default values for the telephone parameters are given below.

Entone Interval:	90 ms. (both on and off time)
Flash Interval:	500 ms.
Busy trunk:	480 & 620 Hz, 0.5 seconds on, 0.5 seconds off.
Ringback tone:	440 & 480 Hz, 2 seconds on, 4 seconds off.
Dial tone:	350 & 440 Hz, steady tone.

3.8 Internal Tone Buffer

Due to the asynchronous nature of touch tone input, the board buffers a single DTMF tone internally to avoid missed tones between commands. This buffer is automatically flushed at the completion of the DIAL and ENTONE commands and may be explicitly cleared by either a SOFT or HARD RESET.

```
                                page    60,132

                        ;********* S A M P L E   C O D E   F O R   Votan  V P C 2 0 0 0 *************
                        ;                                                                           *
                        ;  This sample of some assembly code and comments demonstrates some of the    *
                        ;    programming requirements for the Votan VPC2000 voice board.            *
                        ;    This sample is intended only to illustrate the usage of the VPC2000    *
                        ;    and is not a complete I/O package for the VPC2000                      *
                        ;                                                                           *
                        ;*******************************************************************************

                        ;******* some variables typically needed by VPC2000 software ******************
0000                    data    segment

0000  ????              iobase  dw      ?               ; base I/O address of VPC2000
0002  ??                rfi     db      ?               ; reason-for-interrupt byte
0003  ????              intlvl  dw      ?               ; interrupt level of VPC2000
0005  ??                creg1   db      ?               ; control register 1 mask
0006  ??                creg2   db      ?               ; control register 2 mask
0007    80 [            thevcr  db      128 dup (?)     ; buffer reserved for a VCR/TCR
           ??
               ]

0087                    data    ends

                        ;****** various equates for program constants ********************************

                        ;----- VPC2000 register offsets ----------------------------------------------
= 0000                  RD   equ  00h       ; read-data register offset for VPC2000
= 0001                  RS   equ  01h       ; read-status register offset for VPC2000
= 0002                  WD   equ  02h       ; write-data register offset for VPC2000
= 0003                  CT1  equ  03h       ; control register 1 offset
= 0004                  RST  equ  04h       ; reset register offset
= 0005                  INT  equ  05h       ; interrupt register offset
= 0006                  IAK  equ  06h       ; interrupt acknowledge register offset for VPC2000
= 000B                  CT2  equ  0Bh       ; control register 2 (audio paths) offset
                        ;----- 8259 I/O addresses and codes ------------------------------------------
= 0020                  PICa equ  20h       ; 8259 EOI I/O address in IBM PC
= 0021                  PICb equ  21h       ; 8259 mask I/O address in IBM PC
= 0020                  NEOI equ  20h       ; non-specific EOI code for 8259
                        ;----- VPC2000 RFIs (Reason-For-Interrupt) codes ------------------------------
= 0000                  ADK  equ  00h       ; terminating, no status follows
= 001C                  TERM equ  1Ch       ; terminating, status follows
= 001D                  STAT equ  1Dh       ; not terminating, status follows
= 001E                  READ equ  1Eh       ; data needs to be read from VPC2000
= 001F                  WRIT equ  1Fh       ; data needs to be written to VPC2000
                        ;----- VPC2000 Status register bit masks --------------------------------------
= 0040                  RRDY equ  40h       ; read-ready bit in status register
= 0080                  WBSY equ  80h       ; write-busy bit in status register
```

```
                        page
0000                    code    segment
                        assume  CS:code,DS:data,ES:data,SS:code


                        ;************* VPC2000 Interrupt Handler ***************************************
                        ;                                                                               *
                        ; This sample interrupt handler is shown as a simplied mixture of              *
                        ;   assembly code, comments, and dummy procedures that attempt to show         *
                        ;   the essence of the VPC2000 operation. It is assumed that other             *
                        ;   functions external to interrupt handler are monitoring its progress        *
                        ;   through certain mutually addressible state variables.                      *
                        ;                                                                               *
                        ; assume offset & segment address of this routine has been loaded into:        *
                        ;   0:28 (level 2), 0:2C (level 3), 0:30 (level 4) and                         *
                        ;   the 8259 (PIC) has been enabled for this interrupt level                    *
                        ;   (see initialization code in next segment                                    *
                        ;                                                                               *
                        ;*******************************************************************************

0000                    VPCINT  proc            ; entry point for VPC2000 interrupts

                        ;*** push all register used by any portions of this routine
                        ;*** establish addressibility to all variables used by this routine

                        ;*** read the reason-for-interrupt byte
0000  8B 16 0000 R              mov  dx,[iobase]    ; get the base I/O address of VPC
0004  83 C2 00                  add  dx,RD          ;  & add the Read-Data register offset
0007  EC                        in   al,dx          ; read the RFI byte
0008  A2 0002 R                mov  [rfi],al       ;  & save it for later use

                        ;*** write interrupt acknowledge to VPC
000B  8B 16 0000 R              mov  dx,[iobase]    ; get the base I/O address of VPC
000F  83 C2 06                  add  dx,IAK         ;  & add the Interrupt-Ack register offset
0012  32 C0                     xor  al,al          ; use a zero to write to VPC
0014  EE                        out  dx,al          ; write to port to acknowledge interrupt
                          ;NOTE: The VPC could interrupt again at this point

                        ;*** write non-specific EOI to PIC to re-enable it for new interrupt
                          ;NOTE: Details of PIC operation should be followed from the
                          ;      appropriate system technical reference manuals and 8259 data sheets
0015  BA 0020                   mov  dx,PICa        ; get the I/O address of PIC EOI register
0018  B0 20                     mov  al,NEOI        ; set byte to non-specific EOI
001A  EE                        out  dx,al          ; acknowledge EOI to PIC
                          ;NOTE: Another interrupt could occur now and this routine could be
                          ;      re-entered prior to completing this interrupt. The interrupt
                          ;      handler should either be disabled or fully re-entrant.
```

```
                                page
                        ;*** now analyze and handle the interrupt

001B  A0 0002 R                 mov  al,[rfi]       ; fetch the RFI

001E  3C 00                     cmp  al,AOK         ; RFI = completion without further I/O ?
0020  75 03                     jnz  flih1          ; if not, try another
                        ;*** set some variables to indicate completion so other routines know
0022  EB 31 90                  jmp  VPCINTx        ; go through exit process

0025  3C 1E             flih1:  cmp  al,READ        ; RFI = read data from VPC
0027  75 06                     jnz  flih2          ; if not, try another
0029  E8 0065 R                 call PREAD          ; read the data
                        ;*** set appropriate variables to indicate data is available for
                        ;    additional processing
002C  EB 27 90                  jmp  VPCINTx        ; go through exit process

002F  3C 1F             flih2:  cmp  al,WRIT        ; RFI = write data to VPC
0031  75 0E                     jnz  flih3          ; if not, try another
                        ;       cmp  [stv],Wvcr     ; test some state variable to see if we
                                                    ;  need to write VCR in response to START
                                                    ;  or whether it is really data output
0033  75 06                     jnz  flih2w         ; if not, do data output
0035  E8 00B5 R                 call PVCR           ; write the vcr
0038  EB 1B 90                  jmp  VPCINTx        ; go through exit process
003B  E8 00B9 R         flih2w: call PWRITE         ; write the data
003E  EB 15 90                  jmp  VPCINTx        ; go through exit process

0041  3C 1D             flih3:  cmp  al,STAT        ; RFI = read status without termination
0043  75 06                     jnz  flih4          ; if not, try another
0045  E8 005E R                 call PSTAT          ; read the status
                        ;*** set appropriate variables to indicate status is available for
                        ;    additional processing
0048  EB 0B 90                  jmp  VPCINTx        ; go through exit process

004B  3C 1C             flih4:  cmp  al,TERM        ; RFI = read status and termination
004D  75 06                     jnz  flih5          ; if not, try another
004F  E8 0057 R                 call PTERM          ; read a termination status
                        ;*** set appropriate variables to indicate status is available for
                        ;    additional processing and VPC2000 activity has terminated
0052  EB 01 90                  jmp  VPCINTx        ; go through exit process

0055                    flih5:  ;*** there are no other RFIs defined, so this won't happen


0055                    VPCINTx: ;*** exit code for interrupt handler

                        ;*** set any variables to let other code know what happened

                        ;*** pop all registers pushed at the beginning

0055  FB                        sti                 ; re-enable interrupts (for sure)
0056  CF                        iret                ; return to whatever was interrupted
0057                    VPCINT  endp
```

```
                                page
0057                    ptera   proc            ; read termination status

0057  E8 006C R                 call vpcrlen    ; read length of status data
                                                ;  CX = length
                        ;*** find appropriate buffer (only one needed) to place status,
                        ;    set ES:DI -> buffer, retain length in CX

005A  E8 008F R                 call vpcread    ; use read routine to get the data

                        ;*** set any additional variables

005D  C3                        ret
005E                    ptera   endp


005E                    pstat   proc            ; read a status record

005E  E8 006C R                 call vpcrlen    ; read length of status data
                                                ;  CX = length
                        ;*** dynamically allocate a buffer to place status,
                        ;    set ES:DI -> buffer, retain length in CX

0061  E8 008F R                 call vpcread    ; use read routine to get the data

                        ;*** set any additional variables

0064  C3                        ret
0065                    pstat   endp


0065                    pread   proc            ; read a data record

0065  E8 006C R                 call vpcrlen    ; read length of status data
                                                ;  CX = length
                        ;*** dynamically allocate a buffer to place data,
                        ;    set ES:DI -> buffer, retain length in CX

0068  E8 008F R                 call vpcread    ; use read routine to get the data

                        ;*** set any additional variables

006B  C3                        ret
006C                    pread   endp
```

```
                                page
006C                    vpcrlen proc                ; read length for remaining information

006C  8B 16 0000 R              mov  dx,[iobase]    ; get the base I/O address of VPC
0070  8B DA                     mov  bx,dx          ; make a copy in BX
0072  83 C2 01                  add  dx,RS          ; add the Read-Status register offset
0075  83 C3 00                  add  bx,RD          ; add the Read-Data register offset
0078  EC                rrdy1:  in   al,dx          ; read status byte
0079  24 40                     and  al,RRDY        ; test the read-ready bit
007B  74 FB                     jz   rrdy1          ; if not on, repeat until it is
007D  87 DA                     xchg bx,dx          ; put RD I/O address in DX
007F  EC                        in   al,dx          ; get Least-Significant-Byte of length
0080  8A C8                     mov  cl,al          ; copy to LSB of CX
0082  87 DA                     xchg bx,dx          ; put RS I/O address in DX
0084  EC                rrdy2:  in   al,dx          ; read status byte
0085  24 40                     and  al,RRDY        ; test the read-ready bit
0087  74 FB                     jz   rrdy2          ; if not on, repeat until it is
0089  87 DA                     xchg bx,dx          ; put RD I/O address in DX
008B  EC                        in   al,dx          ; get Least-Significant-Byte of length
008C  8A E8                     mov  ch,al          ; copy to LSB of CX
008E  C3                        ret                 ; CX now = length of remaining data
008F                    vpcrlen endp


008F                    vpcread proc                ; read data, ES:DI->buffer, CX = length

008F  8B C1                     mov  ax,cx          ; copy the length
0091  FC                        cld                 ; set forward direction for auto-inc
0092  AB                       stosw               ; save the length in buffer
0093  83 E9 02                  sub  cx,2           ; subtract length itself from length
0096  0B C9                     or   cx,cx          ; test for only two byte length
0098  75 01                     jnz  vpcr1          ; if >, continue
009A  C3                        ret                 ; done, go back
009B  8B 16 0000 R      vpcr1:  mov  dx,[iobase]    ; get the base I/O address of VPC
009F  8B DA                     mov  bx,dx          ; make a copy in BX
00A1  83 C2 01                  add  dx,RS          ; add the Read-Status register offset
00A4  83 C3 00                  add  bx,RD          ; add the Read-Data register offset
00A7  EC                vpcrr:  in   al,dx          ; read status byte
00A8  24 40                     and  al,RRDY        ; test the read-ready bit
00AA  74 FB                     jz   vpcrr          ; if not on, repeat until it is
00AC  87 DA                     xchg bx,dx          ; switch RD I/O address into DX
00AE  EC                        in   al,dx          ; get data byte
00AF  AA                        stosb               ; save the data byte
00B0  87 DA                     xchg bx,dx          ; switch RS I/O address into DX
00B2  E2 F3                     loop vpcrr          ; repeat until all bytes in
00B4  C3                        ret                 ; all done
00B5                    vpcread endp
```

```
                                page
00B5                    pvcr    proc                    ; write some VCR to VPC

                        ;### locate the VCR and set DS:SI->length preceeding VCR
00B5  E8 00BD R                 call vpcwrit            ; write the vcr to board
00B8  C3                        ret                     ; all done
00B9                    pvcr    endp

00B9                    pwrite  proc                    ; write some data to VPC

                        ;### locate the next data record to write
                        ;     and set DS:SI -> length preceeding data
00B9  E8 00BD R                 call vpcwrit            ; write the data to board
00BC  C3                        ret                     ; all done
00BD                    pwrite  endp


00BD                    vpcwrit proc                    ; write data record/vcr to board
                                                        ;  DS:SI -> length + data

00BD  FC                        cld                     ; set forward direction for auto-inc
00BE  8B 0C                     mov  cx,[si]            ; get the length word
00C0  8B 16 0000 R              mov  dx,[iobase]        ; get the base I/O address of VPC
00C4  8B DA                     mov  bx,dx              ;  and make a copy
00C6  83 C2 01                  add  dx,RS              ; add in status register offset
00C9  83 C3 02                  add  bx,WD              ; add in write-data register offset
00CC  EC                vpcwr:  in   al,dx              ; read the status byte
00CD  24 80                     and  al,WBSY            ; test if VPC still busy with last byte written
00CF  75 FB                     jnz  vpcwr              ; if so, keep on testing until last byte done
00D1  87 D3                     xchg dx,bx              ; put the WD address in DX
00D3  AC                        lodsb                   ; get next data byte
00D4  EE                        out  dx,al              ; write the data byte
00D5  87 D3                     xchg dx,bx              ; put the RS address in DX
00D7  E2 F3                     loop vpcwr              ; continue for the rest of the data
00D9  C3                        ret                     ; all done

00DA                    vpcwrit endp


00DA                    code    ends
```

```
                        page
0000                    codem   segment
                        assume  CS:codem,DS:data,ES:data,SS:codem


                        ;******* VPC2000 Mainline ***************************************************
                        ;                                                                           *
                        ; This sample mainline program initializes the VPC2000 and installs its    *
                        ;  interrupt handler into the PC. It then shows some assembly code and      *
                        ;  dummy procedure to start and stop operations on the VPC.                 *
                        ;                                                                           *
                        ;*****************************************************************************

                        ;*** establish all register and addressibility

                        ;*** hard reset the VPC2000
0000  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of VPC
0004  83 C2 02                  add     dx,WD           ;  & add the Write-Data register offset
0007  B0 00                     mov     al,0            ; hard reset code
0009  EE                        out     dx,al           ; write hard reset code to VPC
000A  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of VPC
000E  83 C2 04                  add     dx,RST          ;  & add the Reset register offset
0011  B0 00                     mov     al,0            ; dummy byte for this port
0013  EE                        out     dx,al           ; cause the reset to occur
                        ;*** now delay while RESET is happening
                        ;    100 clocks/loop x 500 loops = 50,000 clocks = 10ms
0014  B9 01F4                   mov     cx,500          ; # of loops to delay
0017  D4 0A                     pdelay: aam             ; slow instruction (83 clocks)
0019  E2 FC                     loop    pdelay          ; repeat it       (17 clocks)

                        ;*** now setup the control register
001B  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of VPC
001F  83 C2 03                  add     dx,CT1          ;  & add the Control-Register 1 offset
0022  A0 0005 R                 mov     al,[creg1]      ; get the control register 1 mask
0025  EE                        out     dx,al           ; write the mask to control register
0026  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of VPC
002A  83 C2 0B                  add     dx,CT2          ;  & add the Control-Register 2 offset
002D  A0 0006 R                 mov     al,[creg2]      ; get the control register 2 mask
0030  EE                        out     dx,al           ; write the mask to control register
```

```
                        page
                    ;*** install the interrupt vector
                      ;NOTE: the interrupt level is a binary value of 2, 3, or 4. We have to
                      ;      add 8 to that value since the PC offsets the hardware interrupts
                      ;      handled by the PIC by 8 levels. Then we multiply by 4 (two words
                      ;      per interrupt level) to find the offset in low memory
0031  8B 1E 0003 R           mov     bx,[intlvl]     ; get the interrupt level
0035  83 C3 08               add     bx,8            ; add the offset of the PIC levels
0038  03 DB                  add     bx,bx           ; x2
003A  03 DB                  add     bx,bx           ;  & x2 to get double word offset in lowmem
003C  33 C0                  xor     ax,ax           ; get a zero
003E  8E C0                  mov     es,ax           ;  and set ES to low memory. ES:BX-> IV
0040  8D 06 0000 R           lea     ax,VPCINT       ; get offset of our interrupt handler
0044  26: 89 07              mov     ES:[bx],ax      ;  and save that in the IV offset
0047  83 C3 02               add     bx,2            ; advance ptr to segment
004A  8C C8                  mov     ax,cs           ; get our code segment
004C  26: 89 07              mov     ES:[bx],ax      ;  and save that in the IV segment

                    ;*** now enable the PIC to accept this interrupt
                      ;NOTE: see the appropriate hardware technical reference manuals to get
                      ;      explanations of masking interrupt levels in the PIC
004F  8B 0E 0003 R           mov     cx,[intlvl]     ; get the interrupt level
0053  B8 0001                mov     ax,1            ; get a 1 for shifting
0056  D3 E0                  shl     ax,cl           ; left shift that bit by interrupt level
0058  F7 D0                  not     ax              ; complement mask to unmask (enable) that level
005A  8A C8                  mov     cl,al           ; make a copy
005C  BA 0021                mov     dx,PICb         ; get I/O address of PIC mask register
005F  EC                     in      al,dx           ; get current PIC mask
0060  22 C1                  and     al,cl           ; unmask our interrupt level, other lvls unchanged
0062  EE                     out     dx,al           ; write new mask to PIC
                      ;NOTE: everything is now enabled to let interrupts from VPC2000 occur
```

```
                                page
                        ;*** now issue BEEP function to VPC2000
                        ;   build a VCR
0063  B8 0006                   mov     ax,6                    ; length of BEEP VCR
0066  A3 0007 R                 mov     word ptr [thevcr],ax    ; save that length
0069  C6 06 0009 R 07           mov     byte ptr [thevcr+2],07h ; BEEP function
006E  C6 06 000A R 00           mov     byte ptr [thevcr+3],00h ; flags
0073  C6 06 000B R 03           mov     byte ptr [thevcr+4],03h ; set gain in = 3
0078  C6 06 000C R 03           mov     byte ptr [thevcr+5],03h ; set gain out = 3
                        ;   start the command
007D  E8 0087 R                 call    pstart                  ; start the operation
                        ;NOTE: assume that VPCINT is disabled so we will poll it until it is
                        ;     finished by seeing the rfi go to AOK
0080  80 3E 0002 R 00   bdelay: cmp     [rfi],AOK               ; see if we have AOK yet
0085  75 F9                     jnz     bdelay                  ; keep looping until we do
```

```
                                page

0087                        pstart  proc                    ; issue a start command to VPC2000

0087  C6 06 0002 R 77               mov     [rfi],77h       ; set an invalid RFI for poll loops
008C  8B 16 0000 R                  mov     dx,[iobase]     ; get the base I/O address of the VPC
0090  83 C2 02                      add     dx,WD           ;  & add the write-data register offset
0093  B0 01                         mov     al,01           ; start command code
0095  EE                            out     dx,al           ; write start command to board
0096  8B 16 0000 R                  mov     dx,[iobase]     ; get the base I/O address of the VPC
009A  83 C2 05                      add     dx,INT          ;  & add the interrupt register offset
009D  B0 00                         mov     al,00           ; data byte is don't care
009F  EE                            out     dx,al           ; cause an interrupt in the VPC
00A0  C3                            ret                     ; all done, VPCINT takes over from here

                                ;NOTE. After the VPC sees the interrupt caused by the write to its
                                ;      interrupt register, it will read its data byte and recognize
                                ;      the start command. It will then interrupt the system processor
                                ;      and supply a WRIT RFI, indicating it wants the VCR.
                                ;NOTE. In the case of BEEP, a short VCR can be sent to the board and
                                ;      upon completion of the BEEP the VPC will interrupt the system
                                ;      processor with an AOK RFI
                                ;NOTE. This sample routine sets the RFI to a software-defined "busy"
                                ;      status so an external routine can poll the RFI to see the
                                ;      interrupt handler processing the interrupt and determine when
                                ;      the function is complete

00A1                        pstart  endp

00A1                        pstop   proc                    ; issue a stop command to VPC2000

00A1  C6 06 0002 R FF               mov     [rfi],0FFh      ; set an invalid RFI for poll loops
00A6  8B 16 0000 R                  mov     dx,[iobase]     ; get the base I/O address of the VPC
00AA  83 C2 02                      add     dx,WD           ;  & add the write-data register offset
00AD  B0 02                         mov     al,02           ; stop command code
00AF  EE                            out     dx,al           ; write start command to board
00B0  8B 16 0000 R                  mov     dx,[iobase]     ; get the base I/O address of the VPC
00B4  83 C2 05                      add     dx,INT          ;  & add the interrupt register offset
00B7  B0 00                         mov     al,00           ; data byte is don't care
00B9  EE                            out     dx,al           ; cause an interrupt in the VPC
00BA  C3                            ret                     ; all done, VPCINT takes over from here

                                ;NOTE: After the VPC sees the interrupt caused by the write to its
                                ;      interrupt register, it will read its data byte and recognize
                                ;      the stop command. Typically it will then interrupt the system
                                ;      processor and supply a TERM RFI, and then a status record.

00BB                        pstop   endp

00BB                        codem   ends
                                    end
```

```
                                page

0087                    pstart  proc                    ; issue a start command to VPC2000

0087  C6 06 0002 R 77           mov     [rfi],77h       ; set an invalid RFI for poll loops
008C  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of the VPC
0090  83 C2 02                  add     dx,WD           ;  & add the write-data register offset
0093  B0 01                     mov     al,01           ; start command code
0095  EE                        out     dx,al           ; write start command to board
0096  8B 16 0000 R              mov     dx,[iobase]     ; get the base I/O address of the VPC
009A  83 C2 05                  add     dx,INT          ;  & add the interrupt register offset
009D  B0 00                     mov     al,00           ; data byte is don't care
009F  EE                        out     dx,al           ; cause an interrupt in the VPC
00A0  C3                        ret                     ; all done, VPCINT takes over from here

                                ;NOTE. After the VPC sees the interrupt caused by the write to its
                                ;      interrupt register, it will read its data byte and recognize
                                ;      the start command. It will then interrupt the system processor
                                ;      and supply a WRIT RFI, indicating it wants the VCR.
                                ;NOTE. In the case of BEEP, a short VCR can be sent to the board and
                                ;      upon completion of the BEEP the VPC will interrupt the system
                                ;      processor with an AOK RFI
                                ;NOTE. This sample routine sets the RFI to a software-defined "busy"
                                ;      status so an external routine can poll the RFI to see the
                                ;      interrupt handler processing the interrupt and determine when
                                ;      the function is complete

00A1                    pstart  endp

00A1                    codes   ends
                                end
```

COMPRESSED SPEECH DATA FORMAT

RECORD FRAMING

The compressed speech data supplied by the speech board is provided as a continuous stream of data with an internal format comprised of speech RECORDS. Each record consists of 256 bytes of which the first two bytes contain the length of the record and the last byte contains a checksum character. The record length is the length of the entire record including the two length bytes and the checksum byte. The checksum byte is the two's complement of a data sum arrived at by adding together each of the remaining 253 bytes of the data into an eight bit sum with carry wrap-around. The exact count of 256 bytes is not a necessary constant. It is only a requirement that the record conform to a length indicated by the count, that the count not exceed 512 and that the checksum be provided for the data content.

DATA FRAMING

The data removed from the three framing bytes described above can be regarded as a continuous stream of data with additional framing of time significance embedded within it. This framing is recognizable in the form of two unique byte values, FEh (Hex) and FDh (Hex).

The FEh byte is a SYNC byte used to recover synchronization if the data stream is corrupted by errors. The FDh byte is a SILENCE accumulation flag byte which allows a very compact representation of silence at synchronization boundaries.

This split between silence and sync formats has an exception. If the timeout value (ending timeout) is set to zero (infinite timeout value) then silence is never accumulated along with the FDh silence flag. The observed data pattern for all silence in this unique case is alternating sync bytes and all zeros bytes. This 00h FEh pattern represent seven frames of silence at a bit level representation, a padding bit and a sync byte.

The interval between two sync bytes when a silence flag is not found will be 126 milliseconds which corresponds to seven audio intervals of 18 milliseconds each. When an FDh silence flag is found to precede a sync flag with either one or three

intervening bytes the time value of the indicated accumulated silence can be calculated as follows. If there were less than 16 accumulated silent intervals then there will be a single byte between the silent flag and the sync flag in the form 8Xh where "X" is the count of the number of eighteen (18) millisecond intervals of silence. If there are three bytes between the silence flag and the sync flag then the first of these bytes will be 80h and the remaining two bytes will be a 16 bit count of the number of silent frames represented by this sequence.

EDITING OPERATIONS

System software may edit the speech data as long as the resultant edited data be consistent with the requirements of both the record framing and the synchronization framing as described above. The data may not be edited other than at synchronization of the identifiable silence accumulation boundaries. Silence accumulation may be changed according to the silence accumulation format rules described above to reduce the pauses often found in speech. "Pause","rewind" and "fast forward" functions can be accomplished by starting playback at a synchronization boundary. If edited speech is spliced together at sync boundaries there will likely be an audible discontinuity at the boundary. If the splice is made at a silence accumulation there should be no noticeable effect.

If the contents of data is changed within the basic record framing, the checksum and possibly the record length must be recalculated by software to assure correct playback.

If editing silence accumulation to reduce silence and speed playback, the software must take care not to convert an 8Xh byte to an 80h byte as can be seen from the accumulated silence format description.

If implementing a fast forward or rewind function, the software can calculated the position in a speech file with about 1/8th second resolution by identifying the sync and accumulated silence flags.