P-ESI CARD

programmable sound speech and time

LPS Systems, inc.

A Software Wizardry Product.

P-SST Card
Programmable Sound Speech and Time
User's Guide
Manual Version Number 84.10.26

TRADEMARKS

ATARI is a trademark of Atari Corporation
Z-BASIC and Z-DOS are trademarks of Zenith Data Systems Corp
MS and Microsoft are trademarks of Microsoft Corporation
StarIndex is a trademark of MicroPro International
IBM is a trademark of International Business Machines Corp

COPYRIGHT NOTICES

ISBN 0-927092-13-1

DISCLAIMER

LP Systems, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of the merchantability or fitness for any particular purpose. Further, LP Systems, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of LP Systems, Inc. to notify any person or organization of such revision or changes.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the patience, help and cooperation of some valuable people to this project:

Tom Jorgenson of Software Wizardry, Inc.
Dale Wilson of Codewright
John Stetson of Capitol Heath Users' Group
Don Collier of Bendix Test Systems
Dick Lockerby, wherever he may now find himself

Two software packages used in the compilation of this manual deserve special mention: StarIndex (from MicroPro International) and Palette (from Software Wizardry, Inc.). StarIndex was used to generate the table of contents and the index at the end. Palette was used to generate the enclosed drawings.

Not to be forgotten, thanks to my wife Regine and children Christopher, Mark and Leslie for giving me the time and support to complete this project.

Leon Prewitt, LP Systems, Inc.

TABLE OF CONTENTS

L I S T O F F I G U R E S

L I S T O F T A B L E S

1. HOW TO USE THIS MANUAL

This USER'S GUIDE provides information that will enable you to quickly install, and easily learn to work with, the multiple functions that the P-SST Card adds to your computer. The following paragraphs describe the intent and contents of the remainder of the manual.

1.1. Functions and Features of the P-SST Card

Section Two (2) briefly describes the major functions and features of your P-SST Card. The specific programmable features of each major function of the P-SST are documented in sections 5 through 8.

1.2. Hardware Installation

Section Three (3) of this guide provides important information and instructions that should be read and understood before you attempt to install the P-SST Card in your computer.

1.3. Software Description

Section Four (4) of this guide contains the documentation for the comprehensive utilities package delivered with your P-SST. This section contains important information and instructions that should be read and understood before installing the software utility package.

1.4. Programming Information

Sections 5 through 8 of this document present information that will enable both beginning and experienced programmers to write software for the P-SST Card. The following paragraphs describe the format and approach taken in presenting the data in these sections.

1.5. Format

As indicated in the table of contents, each major function of the P-SST Card is presented separately. This approach has been adopted for clarity of presentation, at the expense of redundancy in some instances.

Each function is described in the following sequence:

- Overview
- Functional Description
- Programming Example

Block Diagrams

The block diagrams used to illustrate Overviews and Functional Descriptions do not necessarily represent the hardware configuration of the P-SST Card or the integrated circuits contained on the card. Instead, these diagrams serve as a visual presentation of the operations involved with each function. For specific hardware information, refer to the schematic in section nine (9) of this guide.

Overviews

The overview portions of sections 5 through 8 provide summaries of the programmable features presented in those sections.

Functional Description

The data presented under this heading describes specific requirements of each programmable feature summarized under the overview heading. Details presented under this heading include: function selection, programming data, and command sequence.

Programmable Example

The examples presented in these sections are written in Z-BASIC. Programmers who prefer to program in some other language will have no difficulty in converting these examples.

2. FUNCTIONS AND FEATURES OF THE P-SST CARD

Your Programmable Sound/Speech/Time (P-SST) circuit card is designed for installation in Heath/Zenith Data Systems Z-100 series computers. However, the P-SST Card may be installed in any S-100 buss computer with an available expansion slot.

Although the P-SST card was designed to IEEE-696 specifications, LP Systems, Inc. cannot make any guarantees that the P-SST card will operate in any S-100 system that claims to be IEEE-696 compatible.

Your P-SST Card provides the following functions and features:

o An IEEE-696 S-100 buss compatible interface.

o User-selectable port addressing.

o A real-time clock/calender function featuring battery back-up power, "alarm" mode interrupt, "heartbeat" mode interrupt, and an external "standby" interrupt that can be programmed to activate even when computer power is turned off.

o A three channel sound synthesizer function for generating both music and complex sound effects.

o Two general purpose parallel I/O ports, ideally suited for joystick interface. Note that these ports are not Centronics parallel ports. However, they can be used to drive a Centronics compatible printer with user-provided software and cable hookup.

o A phoneme speech synthesizer with unlimited vocabulary and adjustable voice pitch.

o An audio amplifier with adjustable volume control and output connector for external speaker connection.

o An unamplified audio output for connection to external audio equipment.

3. HARDWARE INSTALLATION

3.1. Initial Inspection

Prior to installing the P-SST Card into your computer, inspect the card for obvious signs of damage that may have occured during shipment. If your P-SST Card was damaged during shipment, this fact should be brought to the immediate attention of the shipper and/or the dealer from whom you purchased the card.

Next, you should fill out, and return, your registration card. This card is not used for warranty information, but is used instead for notification to the user of any newly identified problem areas, updates, and for product support. We must have this card on file prior to responding to any questions from the purchaser.

Be sure to retain your original sales invoice for the P-SST card. Should any problems manifest themselves during the warranty period, you will be required to produce a copy of this invoice as proof of your purchase date.

** IMPORTANT NOTICE **

Do not install a damaged circuit card into your computer. Doing so may result in damage to your system.

Figure 3-1 illustrates the location of components on the P-SST Card.

3.2. Adjustments

There are three adjustable components on your P-SST Card: R25, R26 and C20. R25 and R26 are user adjustments. C20, the clock speed trimmer, should not require adjustment. It is preset prior to shipment, and is extremely difficult to set properly. Should your clock fail to keep accurate time, however, you should refer to Appendix E for the C20 adjustment procedure.

Note that printed circuit calibration components are easily damaged by careless adjustment, or improper tools. Perform all calibration with care and use only tools of correct size and nature. The manufacturer and its distributors will not be responsible for damage resulting from improper use of adjustment tools, or carelessness on the part of the end-user.

R25 is the volume control for the audio amplifier. This potentiometer is 1/2 of a turn from volume off (fully CW) to full volume (fully CCW). Adjust R25 to mid-scale initially. You may readjust it later to suit your own needs.

Distortion from the on-board amplifier will result if the volume setting (R25) is set too high. This may be most noticeable with the speech synthesizer. Should you suspect this problem to be present, merely reduce the volume level by adjusting R25 fully CW, then readjusting for a comfortable level in which distortion does not occur.

R26 is the coarse voice inflection (pitch) adjustment for the speech synthesizer. This potentiometer is 1/2 of a turn from lowest pitch (fully CCW) to highest pitch (fully CW). Adjust R26 to fully CCW initially. You may readjust it later to suit your own preference.

** NOTE **

Occasionally, after powering on your system with the P-SST card installed, you may find that a phoneme is being sounded from the speech synthesizer. This anomaly is due to characteristics of the speech IC beyond our control.

This can be virtually eliminated by replacing R3 (10K ohm) with a 13k-15k resistor and maintaining R26 in the CCW half of its rotation. This will significantly slow down the speech rate and lower the voice pitch. The pitch and rate then can be increased under program control by sending new inflection data (data=2 or 3) to relative port 3. We felt that the present resistor value of 10K ohm for R3 represented an agreeable compromise between intelligible speech and performance.

C20 is preset by the manufacturer (LP Systems, Inc.) to provide a precise reference to the real-time clock. Improper adjustment of C20 will cause the clock to run fast or slow, obviously degrading system performance. Care sould be taken in adjusting C20. Before adjusting, be sure to mark the current setting so that later C20 may be returned to this original position. The adjustment procedure for C20 is located in Appendix E.

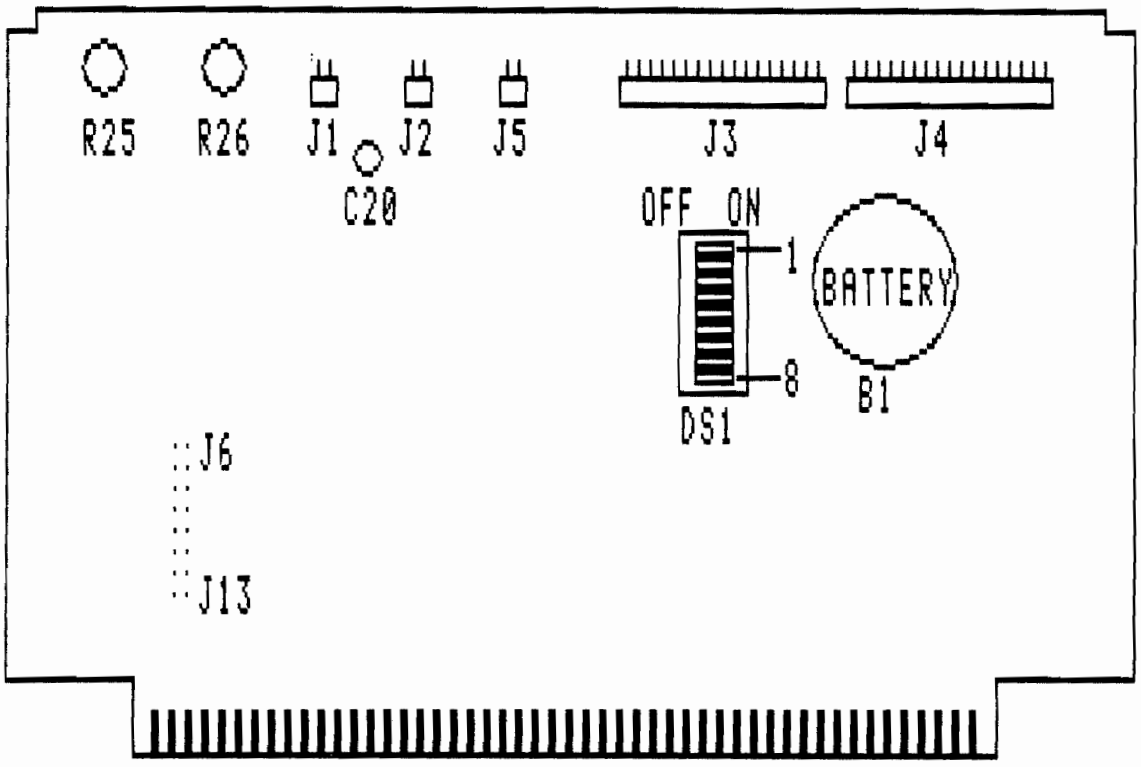
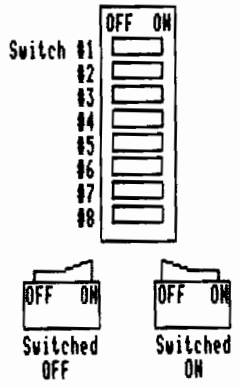


Fig. 3-1: Component Location

3.3. Port Address Selection

The port address of the P-SST Card is selected by switches 1 through 4 of DS1, as shown in figure 3-2. The manufacturer has pre-set your P-SST port address at 0. Insure that switches 0 through 3 of DS1 are set on (to the right.)

If I/O port addresses 0 through 7 is already assigned to some other function within your computer, then set DS1 to an unassigned I/O port address. Remember, the P-SST card occupies the first 8 I/O addresses (0 through 7), starting with the base address selected. Each I/O address port on the P-SST card is referenced to the base address setting. We will speak of the addresses (0 through 7) on the P-SST card as "relative ports" because they are referenced to the base address setting of switch DS1. (Refer to figure 3-2 for switch setting definitions).



SWITCH	DEFINITION
1	BIT 7 Base Port Address (MSB)
2	BIT 6 Base Port Address
3	BIT 5 Base Port Address
4	BIT 4 Base Port Address

** NOTE **

When bits 4 to 7 of the I/O port address match the switch as shown above, bits 0 to 3 select relative ports on the P-SST card.

Fig. 3-2: Port Address selection (Dipswitch DS1)

3.4. Configuration Switches

The configuration data from switches 5 through 8 of DS1 can be examined under program control by reading relative port 7 on the P-SST card. Figure 3-3 defines the configuration options assigned to each segment of the DS1 switches. If you are connecting a joystick or some other external device to J3 (Port 1), then set switch 5 on DS1 to the ON position. Set switch 6 to ON if a joystick or other external device is to be added to J4. Refer to Figure 3-1 for the location of DS1.

SWITCH	PORT 7 BIT	DEFINITION
5	BIT 0	RIGHT means J3 is used
6	BIT 1	RIGHT means J4 is used
7	BIT 2	Not Defined
8	BIT 3	Not Defined

** NOTE **

Bits 4 through 7 of relative port 7 are always zero.

Fig. 3-3: Configuration Option Selection (Dipswitch DS1)

3.5. Battery Installation

Included with your P-SST Card is a lithium battery which is used to power the clock/calendar function of the P-SST Card when your computer is turned off. The nominal life of this battery is nine (9) months. Before installing the P-SST Card into your computer, insure that the lithium battery supplied with the card is installed in the battery holder on the card. Figure 3-4 illustrates proper orientation of the battery. Insure that the positive (+) polarity symbol marked on the battery is oriented as shown and is facing up after it is installed.

If the battery contacts do not make firm contact with the battery, the clock on the P-SST may stop running or even lose the time altogether when the computer is turned off. If this situation occurs, bend the bottom (negative) contact up until it presses against the top contact--then reinsert the battery.

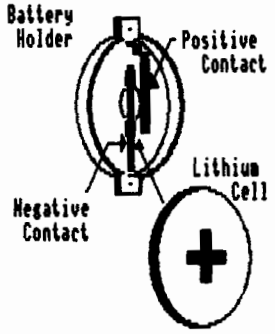


Fig. 3-4: Clock/Calendar Battery Installation

3.6. Circuit Card Installation

Your P-SST Card is designed for installation in the S-100 card cage in the rear of Heath/Zenith Z-100 series computers. Figure 3-5 illustrates P-SST Card orientation with regard to the Z-100 expansion card cage. Insure that the P-SST card is facing forward (component side toward the front panel of the computer) in the buss slot. The removal of the Z-100 top cover should be performed according to the instructions which are contained in the documentation supplied with your Heath/Zenith computer.

Prior to inserting the P-SST card in your machine, insure that DIP switch DS1 is setup according to where in your system I/O address space you wish the card functions addressed. Refer to Figure 3-2 for an explanation of the I/O port selection through DS1.

While inserting the P-SST card into the S-100 buss slot, apply firm pressure evenly to the top of the card. Do not try to force the card into the slot. Occasionally one slot will require slightly more insertion force than another. You may wish to try another slot if you feel that the insertion force is excessive.

Keep in mind that buss slots which have never been used will require firm pressure in order to fit a card in that slot. Insert the P-SST card into an available buss slot at the right rear of your Z-100. MAKE SURE THAT THE COMPONENT SIDE OF THE P-SST IS FACING TOWARD THE FRONT.

** NOTE **

You may need to momentarily disconnect the drive cables which are connected from your disk controller card to the rear panel of the Z-100 in order to gain free access to the S-100 slots. Check carefully to make sure that the P-SST card is completely seated into the slot before proceeding.

o Power on your Z-100. If you do not hear the familiar BEEP or your Operating System fails to Boot then power off and recheck that the card is seated properly and is oriented correctly. Next check that DS1 is set to a non-conflicting base address port. WE SUGGEST THAT YOU SET THE DS1 SWITCHES TO ALL ON (PORT ZERO). This will insure that the P-SST demo and checkout programs will operate correctly. Connect an audio cable to the P-SST.

o Insert the P-SST Distribution Diskette (after making a copy) and run the program called PSSTQA. This program will allow you to check all the functions of the P-SST card to verify operation. For a further explanation of the software, consult the Software Description section 4.

o After installation, reconnect the drive cables (if disconnected previously) and reinstall the Z-100 top cover.

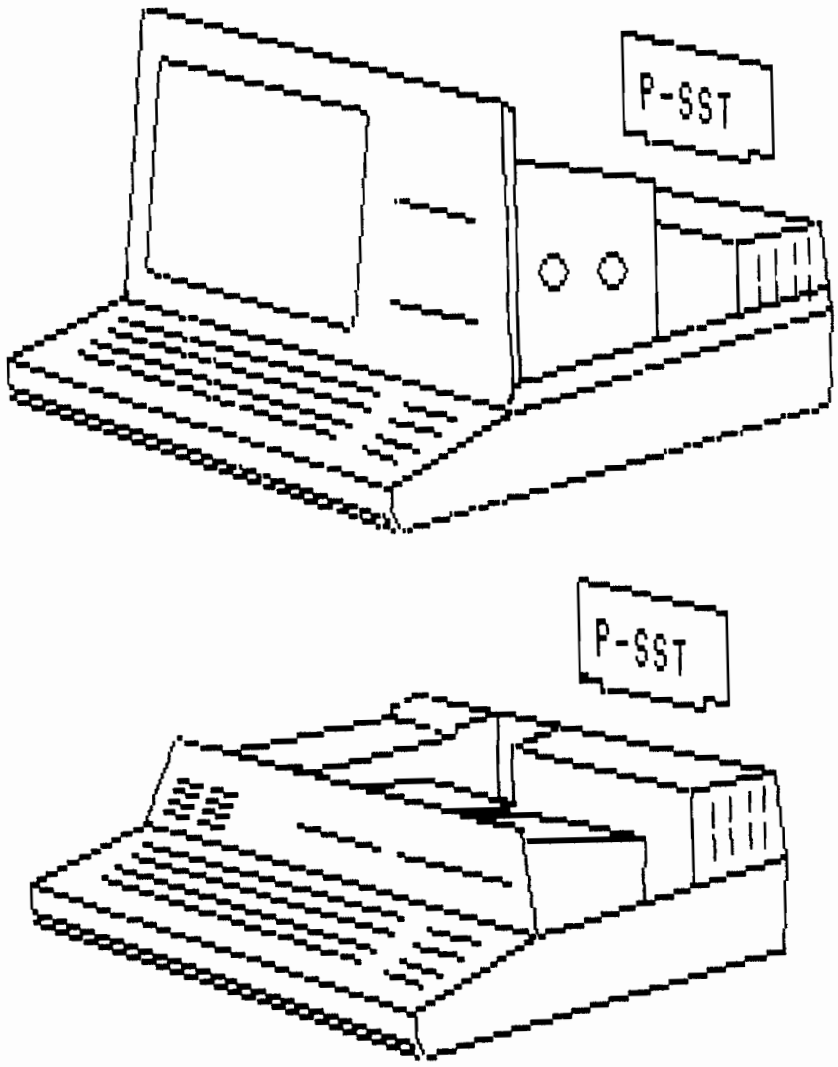


Fig. 3-5: Circuit Card Installation

3.7. Cable Installation

There are five cable connectors at the top edge of your P-SST Card: J1 through J5. Connector J1 provides for interface of the unamplified audio output of your P-SST Card to external audio equipment. Connector J2 provides for direct connection of the amplified audio output of your P-SST Card to external speakers. Connectors J3 and J4 provide for interface of parallel I/O ports A and B respectively. Connector J5 provides for interface of the P-SST Card standby interrupt. Figure 3-6 defines the pinouts for connectors J1 through J5. You may want to refer back to figure 3-1 to find these connectors.

3.8. J1, J2 and J5 Cable Requirements

A mating connector for J1, J2 or J5 is provided with your P-SST Card. The equipment with which you intend to interface will dictate cable construction and second connector selection. If you wish to use the speaker in a ZVM135 monitor, ask your P-SST dealer for part number LPS-PSST-CBL-1.

3.9. J3 and J4 Cable Requirements

Connectors J3 and J4 can be interfaced with the "outside world" with the LPS-PSST-CBL-3, available as a separate product from your P-SST dealer. You may also wish to construct your own cable or to modify Heath/Zenith cables (Part Number Z204-2) for this purpose. Refer to Appendix F for pinout and the details of modifying this cable. Two cables can be used: one for each connector. These connectors are mated to the rear panel of the Z-100 through the connector knockouts. External devices, such as joysticks may require additional adapters to mate to these rear panel connecters. See figures 3-6 and 3-7 for details.

J1 THROUGH J5 CONNECTOR PINOUT DEFINITION

J3 CONNECTOR PORT 1

PIN	SIGNAL	JOYSTICK(PIN)
1	SPARE	
2	BIT D0	RIGHT (4)
3	BIT D1	LEFT (3)
4	BIT D2	DOWN (2)
5	BIT D3	UP (1)
6	BIT D4	FIRE (6)
7	BIT D5	
8	BIT D6	
9	BIT D7	
10	SPARE	
11	SPARE	
12	SIGNAL GROUND	
13-15	SPARES	

J4 CONNECTOR PORT 2

PIN	SIGNAL	JOYSTICK(PIN)
1	SPARE	
2	BIT D0	RIGHT (4)
3	BIT D1	LEFT (3)
4	BIT D2	DOWN (2)
5	BIT D3	UP (1)
6	BIT D4	FIRE (6)
7	BIT D5	
8	BIT D6	
9	BIT D7	
10	SPARE	
11	SPARE	
12	SIGNAL GROUND	
13-15	SPARES	

PIN	J1 SIGNAL	PIN	J2 SIGNAL	PIN	J3 SIGNAL
2	TO EXT. AMP	2	TO SPEAKER	1	STANDBY INT OUT
1	SIGNAL GROUND	1	SIGNAL GROUND	2	SIGNAL GROUND

Fig. 3-6: P-SST Interface Connector Pinout Definition

3.10. Joystick Interface

The P-SST Card provides for interfacing one or two Atari switch type joysticks to your computer. This can be accomplished by first attaching interface cables, LPS-CBL-3, as described in the previous section, between the P-SST card J3 or J4 connector to the Z-100 back panel. Then, joystick adapters can be attached to the back panel connectors of the J3 & J4 interface cables. These adapters are available from you dealer as part number LPS-PSST-CBL-5. If you wish to construct these adapters yourself, the pinouts are shown below:

	Z-100 BACK PANEL DB25P		JOYSTICK END DB9P	
DATA BIT D0	15	<---------->	4	RIGHT
DATA BIT D1	2	<---------->	3	LEFT
DATA BIT D2	17	<---------->	2	DOWN
DATA BIT D3	3	<---------->	1	UP
DATA BIT D4	10	<---------->	6	FIRE
DATA BIT D5	NC		NC	
DATA BIT D6	NC		NC	
DATA BIT D7	NC		NC	
SIGNAL GROUND	12	<---------->	8	GROUND

NC = no connection

Fig. 3-7: P-SST Card Joystick Interface Adapter Wiring

3.11. Vectored Interrupt Buss Interface

The clock/calendar function of your P-SST Card provides a dual function output to the Vectored Interrupt portion of your computer's S-100 buss. The interrupt priority (VI0 Through VI7) is determined by the position of a shorting plug installed in connector strips J6 through J13 respectively. No shorting plugs are necessary for the operation of the P-SST card. Your application will dictate the positioning of this shorting block on one of the available 8 positions. Refer to the Z-100 Technical Manual for further information on the vectored interrupt circuitry. See Figure 3-8 for a definition of the vectored interrupt pins.

VECTORED INTERRUPTS

J6	--	VI7 / S-100 PIN 11
J7	--	VI6 / S-100 PIN 10
J8	--	VI5 / S-100 PIN 9
J9	--	VI4 / S-100 PIN 8
J10	--	VI3 / S-100 PIN 7
J11	--	VI2 / S-100 PIN 6
J12	--	VI1 / S-100 PIN 5
J13	--	VI0 / S-100 PIN 4

Fig. 3-8: Vectored Interrupt Interface Definition

4. SOFTWARE DESCRIPTION

4.1. P-SST Distribution Software Overview

The files on the P-SST software distribution disk are listed below. All of the files on this disk are copyrighted with the exception of the .SCR music score files and the .TXT speech text files which are in the public domain.

Those programs which are copyrighted may not be redistributed in either original or modified form without the express written permission of LP Systems, Inc. However, this permission may easily obtained in most cases via a request sent through Software Wizardry, Inc. describing the intent of this distribution and the nature of the product. Even with such permission, however, credit must be given to the original source and copyright notices must be included intact.

We would encourage those of you who develop software products utilizing the P-SST card to notify us, since we hope to maintain a list of software sources for our customers. Providing the use of the product is non-competitive (with the P-SST card) in nature, and well thought out, you will find us eager to encourage use of your product, and eager to make a success of it.

Z-DOS Batch Command Files

DEMO	.BAT	- Batch file for automatic demonstration
PSST	.BAT	- Batch file invoked by DEMO.BAT

Demonstration Programs

STARWARS	.EXE	- Demonstrates speech, music & graphics
MUSICA	.EXE	- Musical Spiral Art, music & graphics
MUSICK	.EXE	- Musical Kaleidoscope, music & graphics
SPEECH	.EXE	- Demonstrates speech synthesis
SART	.EXE	- String Art, sound & graphics
PSGPONG	.EXE	- Arcade game, sound & graphics

Utility Programs

PSSTQA	.EXE	- P-SST Quality Assurance program
MUSISR	.COM	- Background music interrupt service program
SD	.COM	- Z-DOS Sorted Directory program
CLEAR	.COM	- Program to reset the P-SST card & console
CLOCK	.COM	- CHRONOLOGIC support program for P-SST clock

Music and Speech Data Files

STARWARS.SCR	- Music score, "StarWars"
WMTELL .SCR	- Music score, "The William Tell Overture"
SOLFE .SCR	- Music score, "Solfeggietto"
MOON2 .SCR	- Music score, "The Moonlight Sonata" (2nd m.)
MOON3 .SCR	- Music score, "The Moonlight Sonata" (3rd m.)
BACH1 .SCR	- Music score, "Praedludium I"
BACH2 .SCR	- Music score, "Praedludium II"
BACH3 .SCR	- Music score, "Minuet"
BACH4 .SCR	- Music score, "Minuet in D minor"
BLOWIN .SCR	- Music score, "Blowing in the Wind"
JAMAICA .SCR	- Music score, "Jamaican Farewell"
DIXIE .SCR	- Music score, "Dixie"
EDELWEIS.SCR	- Music score, "Edelweiss"
CABARET .SCR	- Music score, "Cabaret"
DOWNTOWN.SCR	- Music score, "Downtown"
SILENCE .SCR	- Music score, "The Sounds of Silence"
TAKEFIVE.SCR	- Music score, "Take Five"
WINDMILL.SCR	- Music score, "The Windmills of Your Mind"
STARWARS.TXT	- Speech text, "StarWars"
JABWOCKY.TXT	- Speech text, "The Jabberwocky"
ADVENTUR.TXT	- Speech text, "Adventure"

Z-BASIC Language Utilities & Sample Programs

P-SST .BAS	- Z-BASIC interpreter interface program
VOTRAX .BAS	- Votrax interface sample program
PSG .BAS	- PSG interface sample program
CLOCK .BAS	- Clock interface sample program
SPACEWAR.BAS	- Joystick interface sample program
SPEAK .BLD	- Votrax Speech subroutine in BLOAD format
MUSIC .BLD	- PSG Music Synthesis subroutine in BLOAD format
SOUND .BLD	- PSG Sound Effects subroutine in BLOAD format
TONE .BLD	- PSG Tone Generation subroutine in BLOAD format
JOYSTK .BLD	- Joystick subroutine in BLOAD format

4.2. Hardware and Software Requirements

The following is a list of the required and optional hardware and software necessary to effectively utilize the P-SST card with the software provided on this disk:

Hardware

o A Heath/Zenith H/Z-100 series microcomputer with 128k bytes or more of main memory is required (note that non S-100 models in this series are not supported by the P-SST card).

o Color video RAM chips and a color monitor are recommended, but are not required.

o A P-SST card installed in one of the S-100 slots in the computer, with the base port address set to 0, is required.

o An external speaker and/or audio amplifier connected to the P-SST card is required for the speech, sound, and music. A standard 8 ohm speaker may be attached to the appropriate two pin connector (J2) on the P-SST card which is connected to the on-board audio amplifier. Alternatively, an external audio amplifier and speaker may be connected to the two pin connector on the P-SST card which provides the preamplifier output (J1). An external amplifier is recommended to minimize any turn-on "thump" from the speaker, and to permit easy adjustments of the sound volume.

o One or two Atari style joysticks are required for programs which use the joystick ports on the P-SST card. These joysticks have 4 internal switches which permit 8 different directions to be indicated by the user. They also have one or more external switches which are referred to as "fire" buttons.

o Foreground timing calculations are done using the on-board P-SST Real Time Clock so that music, sound and speech will be independent of the system CPU speed (e.g. 5 or 8 MHz). Timing calculations for background music (played by the resident MUSIC Interrupt Service Routine) are dependent on the resolution of the Z-DOS 8253 timer interrupt (100 msec. for Z-DOS BIOS V1.0 and 10 msec. for Z-DOS BIOS V1.1 and MS-DOS V2.x). Use of the original version of Z-DOS (BIOS V1.0) is not recommend, due to the coarse timer resolution.

Software

o The Z-DOS or MS-DOS 2.x Operating System is required. Z-DOS Release 1.01, Version 1.25, BIOS Release 1.00, Version 1.10 or higher is recommended.

o The Z-BASIC interpreter is required in order to list and execute the .BAS files on the distribution disk. These files will primarily be of interest to those who wish to program the P-SST card using the Z-BASIC interpreter.

o MTR-100 Version 1.2 or higher is required for proper operation of some of the programs on the P-SST distribution disk. Some early Z-100's have Version 1.1 of the MTR-100 Monitor ROM. Press Shift-Reset, followed by DELETE, to enter the MTR-100 ROM code. At this point typing V will display the monitor version.

4.3. Z-DOS Batch Command Files

The DEMO.BAT file is designed to provide an automatic demonstration of the speech, music, and graphics capabilities of the P-SST card and the Z-100 microcomputer. Perform the following steps to start the automatic demonstration on a Z-100

with 2 floppy disk drives:

1) Install the P-SST card in one of the S-100 slots. Make sure that the base port address is set to zero by the DIP switches on DS1.

2) Connect an external speaker and/or amplifier to the P-SST card.

3) Make a backup copy of the P-SST distribution disk using DSKCOPY.

4) Place the copy of the P-SST distribution disk in drive B:.

5) Place a bootable Z-DOS disk in drive A: and boot Z-DOS.

6) Enter the following command and press Return: B:DEMO B:

The demonstration takes about 50 minutes to finish. In order to avoid prompts to insert a disk containing the file COMMAND.COM, it is desirable to make sure that the currently logged-in drive contains this file before starting the demonstration. The DEMO.BAT batch file will accept an optional command line parameter which specifies the name of the drive containing the distribution disk. This allows the demonstration to be executed from any valid drive.

You may press any key on the keyboard to prematurely terminate any of the demonstration programs. When the demonstration reaches the end (String Art program), it can be restarted by pressing any key.

The PSST.BAT file is executed internally by the DEMO.BAT file so that the MUSISR.COM program is only executed the first time the demonstration is run.

4.4. Demonstration Programs

There are a variety of demonstration programs included on the P-SST software distribution disk. These programs all use the graphics capabilities of the Z-100 computer, and require a color monitor for optimum performance. These programs are designed to illustrate the simultaneous use of the Z-100's graphics capabilities and the sound (music) and speech capabilities of the P-SST S-100 card.

Although these programs are intended for use on Z-100's equipped with color capability, most will perform reasonably on monochrome display systems. All of the programs will run on Z-100 systems having 128K or more of main memory. All of these programs require that the P-SST card be addressed (via DIP switch DS1) at base port 0.

The MUSICA, MUSICK, SPEECH and SART programs were written using the Microsoft MS-PASCAL compiler and linked with the PSST.LIB and the GRAPHICS.LIB P-SST support libraries, which are implemented for most of the Microsoft languages, and are available as a separate product.

The STARWARS and PSGPONG programs were written using the Microsoft Z-BASIC compiler and linked with the PSST.LIB support library. These programs require that the Z-100 in use have MTR-100 V1.2 or higher to operate correctly.

The MUSICA (Musical Spiral Art) and MUSICK (Musical KaleidoScope) programs demonstrate simultaneous background music playing and color graphics. They read a user specified ASCII music score file and play the musical notes and rests (pauses) indicated in score. The score file name is specified on the Z-DOS command line:

```
B:MUSICK B:WMTELL.SCR
```

The .SCR files on the distribution disk are provided as examples of how to prepare input for the MUSIC subroutine and the MUSIC programs. The notation used is similar to that supported by the IBM-PC BASICA "PLAY" command; these scores are available from a variety of sources and can be easily formatted for playing by the MUSIC programs. Scores can also be transcribed from sheet music. See the MUSIC subroutine documentation below for additional details. The MUSISR program must be executed prior to running the MUSICA and MUSICK programs. See the MUSISR documentation below for details.

SPEECH is a speech synthesis program which uses the SPEAK routine to both speak and display the words in a user specified ASCII text file. The text file name is specified on the Z-DOS command line:

```
B:SPEECH B:ADVENTUR.TXT
```

The .TXT files on the P-SST distribution disk are provided as examples of how to prepare input for the SPEECH program and SPEAK subroutine with a minimum of pronunciation errors. See the SPEAK subroutine documentation below for additional details. There is no memory limitation on the size of the file which can be spoken.

The SPEECH program and SPEAK routine have an accuracy rate of about 75%; the text to speech algorithm they use will usually cause about 3 in 4 words to be spoken in an acceptable manner, with one or more mispronounciations in the remaining word. This is unavoidable in an algorithm of this kind, since it represents a compromise between accuracy versus size and speed. If the text

to be spoken is not known in advance, the default text to speech algorithm must be used with the attendant pronunciation errors. A more typical case is that the words to be spoken are known by the programmer at the time the program which calls SPEAK is written. In this case, corrective action can easily be taken by intentionally misspelling the words passed to the SPEAK subroutine, so that they are spoken correctly.

The most typical types of pronunciation errors occur for vowels, and can usually be corrected by substituting another vowel or vowels in place of the original. A little experimentation with the Speech Test option of the PSSTQA utility program will enable you to easily correct the pronunciation errors which are unavoidably introduced by the default text to speech algorithm. Simply type in the text to be spoken, and try different spellings for any words which are not pronounced in an acceptable manner. Record these for later reference and use them when preparing input text for SPEAK.

4.5. Utility Programs

Several utility programs are also included on the P-SST software distribution disk. These include PSSTQA, MUSISR, SD, CLEAR and CLOCK. PSSTQA was written using the Microsoft MS-PASCAL compiler, and the others were written in 8088 assembly language.

PSSTQA is a quality assurance program which serves two main purposes. First, it allows you to check out each function of the P-SST card separately, to verify proper operation. Second, it acquaints you with the various capabilities of the P-SST card and includes several demonstrations for this purpose. You may wish to run this program when you initially install the P-SST card in your computer, to make sure that the DIP switches, external speaker (or amplifier), and joysticks (if available), are correctly installed. The program is menu driven with the following options in the primary menu:

P-SST Configuration Tests
Programmable Sound Generator Tests
Joystick Tests
Votrax Speech Tests
Real Time Clock Tests
Demonstrations (Music, Sound & Speech)

MUSISR is a utility program which installs a resident interrupt service routine which is required to play "background" music. This allows the 8088 CPU to perform another function (typically graphics or console I/O) while the music is playing. MUSISR should be run exactly once after booting Z-DOS before the MUSICA or MUSICK programs are used. Internal checks in these programs detect attempts to circumvent this sequence. You may find it convenient to place the MUSISR command in an AUTOEXEC.BAT file.

MUSISR contains about 1k of code and a 16k data buffer which is used for storing information which enables it to play musical notes. This memory is not available for use by the operating system, and the only way to reclaim it is to reboot the system. When not playing a music score, the overhead caused by MUSISR is negligible and should not interfere with normal operating system functions.

When playing background music with MUSICA and MUSICK, Z-DOS version 1.1 or higher should be used. The original version of Z-DOS (1.0) had a relatively long interval between clock timer "ticks" (100 msec) which will noticeably degrade the quality of the music playing. Floppy disk I/O interrupts which occur while background music is being played will result in an increase in the time between timer intercepts by MUSISR, and will cause correspondingly lengthened notes. Execution of other timer interrupt interception programs such as spoolers may cause unpredictable results due to contention. MUSISR requires 4 bytes to represent each note and 2 bytes to represent each pause in the resident data buffer. This allows approximately 2730 notes to be stored in the buffer, which might represent up to about 10 minutes of music, depending on note length and tempo.

SD is a sorted directory utility program which provides a convenient method of displaying the files and their attributes on a Z-DOS disk. Type "SD/X" for a summary of the command syntax and operands. This program is based upon a public-domain utility of the same name, and is included strictly as a convenience to the operator.

CLEAR is a handy utility program which resets the Z-100 console terminal to its default configuration state and kills all sound from the sound and speech circuits on the P-SST card. This is useful when debugging programs which may leave sound or speech enabled.

CLOCK provides support for the P-SST real time clock. The primary use of this utility is to set the Z-DOS system time and date at boot time from the real time clock on the P-SST card. This is accomplished by executing the command "CLOCK READ", which should be the first line in a file named "AUTOEXEC.BAT" on your boot disk. This will automatically be executed when the operating system is booted up. Clock will also reset the speech synthesizer to eliminate any startup noise from that portion of the P-SST card.

Additionally, CLOCK.COM can be used as a console command. For example, typing:

```
CLOCK
```

will display the date and time of day as kept by Z-DOS.

Typing:

CLOCK mm/dd/yy,hh:mm:ss

will set the date and time of day in Z-DOS and in the P-SST clock. This command may be abbreviated by omitting portions of the date and time. For example:

CLOCK /dd,hh:mm

will set the day of the month to dd and set the time to hh:mm, but it will not change the month or the year.

The CLOCK command is actually part of the CHRONOLOGIC package available from Software Wizardry, Inc. This package includes support for several hardware clocks and the source program for the CLOCK command to allow it to be customized for other needs.

4.6. Music and Speech Data Files

A variety of music score (.SCR) and speech text (.TXT) files are provided to both demonstrate the capabilities of the MUSIC and SPEECH programs, and to illustrate the notation used for their input. These files are prepared using any standard text editor and their format is described in detail below in the MUSIC and SPEAK routine documentation. These files are intended to be in the public domain and to be freely exchanged between owners of P-SST cards.

4.7. Z-BASIC Language Utilities & Sample Programs

Although use of the PSST.LIB and GRAPHICS.LIB linking libraries with one of the MS-DOS language compilers is recommended to take full advantage of the capabilities of the P-SST card, support is provided for the Z-BASIC interpreter, which should be adequate for small to medium size applications. Versions of five of the PSST.LIB routines are included on the distribution disk: SPEAK, MUSIC, SOUND, TONE, and JOYSTK. These files are in a special "BLOAD" format for the Z-BASIC interpreter and cannot be used with the Z-BASIC compiler. These routines are described individually later on in this document. All of the .BAS files on the P-SST distribution disk are saved in Z-BASIC binary (tokenized) format, and must be loaded inside Z-BASIC in order to list them.

The PSST.BAS program illustrates the use of these routines with the Z-BASIC interpreter. Using several assembly language subroutines with the interpreter requires a bit of "housekeeping" and has other disadvantages:

o You must remember to invoke the Z-BASIC interpreter in such a way that it reserves the upper portion of its data segment for the subroutines which will subsequently be loaded. This is accomplished using the command: ZBASIC/M:32768, to reserve the upper 32k of the Z-BASIC data segment.

o The BLOAD segment address is dependent on the amount of main memory installed in the Z-100 computer; see the PSST.BAS program source code and the Z-BASIC interpreter manual for further details.

o The subroutines must be loaded into memory from separate disk files each time the BASIC program is run; this is inefficient and requires that the files be on a known disk at run time.

o Variables must be used as the subroutine arguments. Numeric constants (4) or string literals ("abc") cannot be used.

o The P-SST card base port address is assumed to be zero.

Another implementation restriction is that separate segments must be used for the SPEAK, MUSIC, and PSG subroutines. This requires that a DEF SEG statement be used each time a switch is made from the SPEAK routine to the MUSIC routine, to any of the other routines and is due to the need for separate local data areas by the routines.

None of the above restrictions applies to the Z-BASIC compiler if the PSST.LIB linking library is used.

Another point to remember is that all of the parameters passed to the subroutines are either varying length character strings, or 16 bit integers. Be sure to either use a DEFINT statement or the % notation when defining integer parameters in a BASIC program.

See the PSST.BAS source code for additional details.

4.7.1. The SPEAK subroutine

Function

Speak the specified text string

Usage

```
CALL SPEAK(text,length,echo,delay)
```

Input parameters

text

An ASCII character string containing the text to be spoken enclosed in quotation marks (BASIC language). All upper and lower case ASCII letters and digits are translated into phonemes using a text-to-speech algorithm and the selected phonemes are then spoken by the Votrax SC01 chip. All other ASCII characters except those below act as interword delimiters and will not generate speech. This parameter is passed by reference (address).

The following characters are treated specially:

[

If echo=1, indicates that any characters which follow (until the next | character) are to be echoed to the console but not spoken.

|

If echo=1, indicates that any characters which follow (until the next] character) are to be spoken but not echoed to the console.

]

If echo=1, indicates that any characters which follow are to be both echoed and spoken.

$

If length=0, indicates the end of the text string. If echo=1, the echoed text is not followed by CR/LF.

*

If length=0, indicates the end of the text string. If echo=1, the echoed text is followed by CR/LF.

>

Increases the pitch of the speech inflection.

<

Decreases the pitch of the speech inflection.

,;:

Cause a pause of 16 * delay milliseconds.

.!?

Cause a pause of 32 * delay milliseconds.

length

The length of the text string in characters from 0 to 255. A zero value indicates that a special character in the text of the string indicates the string length. If this method is used, it eliminates the need for the caller to calculate the exact number of characters in the text string, which may be much less convenient. In the BASIC language version, the text length will be obtained from the String Descriptor created by BASIC, if length is set to zero, which is recommended.

echo

A flag indicating whether or not the input text should be written to the system console before it is spoken. If echo=(0/1), text (will not/will) be echoed. The ...[...|...]... sequence in the text string can be used to improve pronunciation by misspelling without affecting the spelling of the words echoed to the console. Since Z-BASIC normally controls the console screen, care must be taken to avoid overlaying text displayed by Z-BASIC.

delay

Time delay to wait between speaking phonemes measured in milliseconds from 1 to 32767. If delay=0, then a default value of 20 milliseconds is used. The smaller the delay, the more rapid the speech will be.

Output parameters

None

Example of use

```
ZBASIC/M:32768
100 DEFINT A-Z
110 DEF SEG=&H2F00 : SPEAK=0 ' for Z-100 with 192k+ memory
110 DEF SEG=&H1F00 : SPEAK=0 ' for Z-100 with 128k memory
120 BLOAD "B:SPEAK.BLD",SPEAK
130 TEXT$ = "sli[t|]hy to[ve|ov]s"
140 LENGTH = 0 : ECHO = 1 : DELAY = 0
150 CALL SPEAK(TEXT$,LENGTH,ECHO,DELAY)
```

Description

The text-to-speech algorithm employed represents a trade-off between accuracy and size/speed, etc. The more exception rules that are employed in an algorithm of this kind, the greater the accuracy in the pronunciation of the spoken text will be. If the text of a string to be spoken is known in advance, better results can sometimes be obtained by intentionally misspelling the words to circumvent the default which would otherwise be used by the built in text-to-speech algorithm. Of course, the Votrax chip can be directly programmed at the phoneme level if maximum accuracy is needed. There are four different speech inflections

which may be selected using the <> symbols. The default inflection has the lowest pitch.

4.7.2. The MUSIC subroutine

Function

Play the specified musical score

Only one musical voice is supported by this routine. The music is played in the "foreground", which means that the calling program will not receive control back until after the musical score has been played.

Usage

CALL MUSIC(score,length,init)

Input parameters

score

An ASCII character string containing the musical score to play enclosed in quotation marks (BASIC Language). The score is composed using a "Music Definition Language" which is described in detail below.

length

The length of the score string in characters from 0 to 255. A zero value indicates that a dollar sign ($) in the text of the string indicates the string length. If this method is used, it eliminates the need for the caller to calculate the exact number of characters in the score string, which may be much less convenient. In the BASIC Language version, the score length will be obtained from the String Descriptor created by BASIC, if length is set to zero, which is recommended.

init

A flag indicating whether or not this is the first call to the MUSIC routine for the current score. A score which is longer than 255 characters must be broken up into smaller segments, each of which is passed to the MUSIC routine during successive calls. The caller must set init=1 before the first call, and to any other 16 bit integer value before successive calls to continue playing the rest of the musical score.

Output parameters

None, however, various messages are issued if syntax errors are detected in the score; i.e. "Bad octave".

Example of use

```
ZBASIC/M:32768
100 DEFINT A-Z
110 DEF SEG=&H2F00 : MUSIC=0 ' for Z-100 with 192k+ memory
110 DEF SEG=&H1F00 : MUSIC=0 ' for Z-100 with 128k memory
120 BLOAD "B:MUSIC.BLD",MUSIC
130 ' Play "Mary had a little lamb"
140 SCORE$ = "t100 o3 l8 gfe-fggg p8 fff4"
150 LENGTH = 0 : INIT = 1
160 CALL MUSIC(SCORE$,LENGTH,INIT)
170 SCORE$ = "gb-b-4 gfe-fggg gffgfe-."
180 INIT = 0
190 CALL MUSIC(SCORE$,LENGTH,INIT)
```

Description

Music Definition Language

This notation for musical scores (which may be transcribed from sheet music), provides a concise way of specifying the desired tone frequencies (octave and note), pauses (rests), note volume, note length, and the inter-note delay time. The tempo, or rate of playing, may also be specified.

A score is composed of a series of orders and actions which may be optionally separated by either spaces or semicolons. The letters in the score may be either upper or lower case. The maximum length of the score string is 255 characters, which may optionally be terminated by a dollar sign ($). The text of a score line following an apostrophe (') is not processed; it is considered to be a comment and not an error.

Orders remain in effect until they are respecified. Actions are executed immediately to produce notes and pauses (rests).

Orders

The <M>ode order provides compatibility with the IBM-PC BASICA "Play" command. The available modes are:

```
MF = Music Foreground - the default
MB = Music Background - ignored
MS = Music Staccato   - same as the S4 order
MN = Music Normal     - same as the S8 order
ML = Music Legato     - same as the S0 order
```

The <T>empo order sets the number of quarter notes in a minute, and may range from 32 (slowest) to 255 (fastest). The default is 120.

The most common tempos and their interpretations are:

Tempo	Beats/Minute	Speed
Largo	40-60	very slow
Larghetto	60-66	very slow
Adagio	66-76	slow
Andante	76-108	slow-medium
Moderato	108-120	medium
Allegro	120-168	fast
Presto	168-208	very fast

The <V>olume order specifies the intensity with which the notes are played, from 0 (least) to 15 (most). By varying the volume from note to note, musical "dynamics" can be achieved. The default volume level is 15.

The <O>ctave order specifies the musical octave from 0 to 7 for the following notes. The octave definitions are given in the Note Frequency Table, below. The default octave is 4. The > symbol increases the current octave number by one and the < symbol decreases the current octave number by one.

The <L>ength order specifies the length of the following notes from 1 to 64. The note length is computed as 1/L; i.e. L1 is a whole note, L2 is a half note, ..., L64 is a sixty-fourth note. The length of an individual note may be changed from the current L value by specifying the length after the note: A16 is equivalent to L16 A. The default length is 4.

Odd note lengths are supported, but "triplets", "quintuplets" and other musical effects must be implemented by temporarily changing the tempo, as necessary. For example, the score:

```
t120 l4 c d e t180 f g a t120 b > c d
```

plays notes f, g, a as a "quater note triplet". The three notes are played in the time it would normally take to play two quarter notes.

The <S>ize order specifies the inter-note delay time in terms of the length of a note being played (Delay = Length/Size). The effective length of a note is computed as follows:

Note Length = (Length * (Size-1))/Size, if Size > 0.

S0 = slurred (no delay between notes), S1 = pause (no tone), S2-4 = staccato (short), S8 = normal, S16-64 = legato (long). The default size value is 8.

Actions

A pause or rest is a period of time during which no notes are being played. The <P>ause action specifies the length of a pause from 1 to 64. The length of the pause is computed as 1/P; i.e. P1 is a pause equivalent in length to a whole note, P2 is equivalent in length to a half note, ..., P64 is equivalent in length to a sixty-fourth note. If the pause symbol is followed by 1 dot, the length is multiplied by 3/2. If it is followed by 2 dots, the length is multiplied by 9/4.

Notes are specified by the letters C,D,E,F,G,A,B. This is the order that the notes occur within each octave. The frequency of a note is determined by the octave and the letter of the note. The frequency may be modified by following the note symbol by (#) or (+) to denote a sharp, or by (-) to denote a flat (C+ is equivalent to D-). Not all of the notes have sharps and flats, due to the two half-steps in each octave: There is no B# or E# and no C- or F-. A note symbol may be followed by a note length specification from 1-64, mentioned above, or by a single or double dot (.), which lengthens the note duration, as described for <P>ause, above.

Grammar

```
<score>   ::= <command> {< >|<;> <command>} ... {$}
<command> ::= <order> | <action> | <comment>
<order>   ::= <mode> | <tempo> | <volume> |
              <octave> | <length> | <size>
<action>  ::= <pause> | <note>
<mode>    ::= MF | MB | MS | MN | ML
<tempo>   ::= T 32-255     (beats/minute, default=120)
<volume>  ::= V 0-15       (default=15)
<octave>  ::= O 0-7        (default=4, < = down, > = up)
<length>  ::= L 1-64       (usually 2 ^ n, default=4)
<size>    ::= S 0-64       (usually 2 ^ n, default=8)
<pause>   ::= P 1-64 {..}  (usually 2 ^ n)
<note>    ::= A-G {<sharp>|<flat>} {1-64} {..}
<sharp>   ::= <#>|<+>      (not after note B or E)
<flat>    ::= <->          (not after note C or F)
<comment> ::= <'><any text>
```

Conversion of IBM-PC BASIC scores

The IBM-PC Note command (N 0-84) is redundant and rarely used, and is therefore not supported.

The IBM-PC X command which is used to include a substring into the score in BASIC, is not implemented.

Note Frequency Table

The following table provides a summary of the scheme used to number the notes used in a score. The table columns are: octave number, note, frequency of note in Hz, and the values of the

Programmable Sound Generator Tone Period registers (both Coarse and Fine) which correspond to the note frequency:

Oct	Note	Frequency	TC	TF	Oct	Note	Frequency	TC	TF
0	C	32.7032	0E	EE	4	C	523.251	00	EF
0	C#	34.6478	0E	18	4	C#	554.365	00	E1
0	D	36.7081	0D	4D	4	D	587.329	00	D5
0	D#	38.8909	0C	8E	4	D#	622.254	00	C9
0	E	41.2034	0B	DA	4	E	659.255	00	BE
0	F	43.6535	0B	2F	4	F	698.456	00	B3
0	F#	46.2493	0A	8F	4	F#	739.988	00	A9
0	G	48.9994	09	F7	4	G	783.990	00	9F
0	G#	51.9131	09	68	4	G#	830.609	00	96
0	A	55.0000	08	E1	4	A	880.000	00	8E
0	A#	58.2705	08	61	4	A#	932.327	00	86
0	B	61.7354	07	E9	4	B	987.766	00	7F
1	C	65.4064	07	77	5	C	1046.50	00	77
1	C#	69.2956	07	0C	5	C#	1108.73	00	71
1	D	73.4162	06	A7	5	D	1174.66	00	6A
1	D#	77.7817	06	47	5	D#	1244.51	00	64
1	E	82.4069	05	ED	5	E	1318.51	00	5F
1	F	87.3070	05	98	5	F	1396.91	00	59
1	F#	92.4986	05	47	5	F#	1479.98	00	54
1	G	97.9988	04	FC	5	G	1567.98	00	50
1	G#	103.826	04	B4	5	G#	1661.22	00	4B
1	A	110.000	04	70	5	A	1760.00	00	47
1	A#	116.541	04	31	5	A#	1864.65	00	43
1	B	123.471	03	F4	5	B	1975.53	00	3F
2	C	130.813	03	BC	6	C	2093.00	00	3C
2	C#	138.591	03	86	6	C#	2217.46	00	38
2	D	146.832	03	53	6	D	2349.32	00	35
2	D#	155.563	03	24	6	D#	2489.01	00	32
2	E	164.814	02	F6	6	E	2637.02	00	2F
2	F	174.614	02	CC	6	F	2793.82	00	2D
2	F#	184.997	02	A4	6	F#	2959.95	00	2A
2	G	195.998	02	7E	6	G	3135.96	00	28
2	G#	207.652	02	5A	6	G#	3322.44	00	26
2	A	220.000	02	38	6	A	3520.00	00	24
2	A#	233.082	02	18	6	A#	3729.31	00	22
2	B	246.942	01	FA	6	B	3951.06	00	20
3	C	261.625	01	DE	7	C	4186.00	00	1E
3	C#	277.182	01	C3	7	C#	4434.92	00	1C
3	D	293.665	01	AA	7	D	4698.63	00	1B
3	D#	311.127	01	92	7	D#	4978.03	00	19
3	E	329.627	01	7B	7	E	5274.04	00	18
3	F	349.228	01	66	7	F	5587.65	00	16
3	F#	369.994	01	52	7	F#	5919.90	00	15
3	G	391.995	01	3F	7	G	6271.92	00	14
3	G#	415.305	01	2D	7	G#	6644.87	00	13
3	A	440.000	01	1C	7	A	7040.00	00	12
3	A#	466.164	01	0C	7	A#	7458.61	00	11
3	B	493.883	00	FD	7	B	7902.12	00	10

Table 4-1: Note Frequency Table

The C note in octave 3, in the table, is middle C on the piano keyboard. The A note in octave 3 is the tuning note. The C note in octave 7 is the highest note on the piano.

4.7.3. The SOUND subroutine

Function

Generate various sound effects

Usage

CALL SOUND(effect)

Input parameters

effect

The number of the desired sound effect from 1 to 12:

01 - Siren	07 - Race Car
02 - Gun Shot	08 - Chime
03 - Explosion	09 - Ray Gun
04 - Laser	10 - Ocean Waves
05 - Whistling Bomb	11 - Cricket
06 - Wolf Whistle	12 - Random Tones

Output parameters

None

Description

This is a mid-level routine which generates one of several "pre-recorded" sound effects. A given sound effect may be repeated by calling the routine repeatedly, possibly waiting for a given time period between calls, as desired.

4.7.4. The TONE subroutine

Function

Generate a tone for the specified time

Usage

CALL TONE(channel,frequency,volume,msecs,delay)

Input parameters

channel

The desired PSG channel number (1=A, 2=B, 3=C).

frequency

The desired tone frequency for the specified channel from 31 Hz to 31250 Hz.

volume

The desired tone volume from 0 to 15.

msecs

The length of time to generate the tone specified in milliseconds from 1 to 32767.

delay

The length of time to wait after the tone specified in milliseconds from 0 to 32767.

frequency calculations:

tone period = 125000 / tone frequency (2MHz clock)
31 <= tone frequency <= 31250
4 <= tone period <= 4032

Output parameters

None

Description

This is a mid-level routine which provides a convenient way of generating a tone of the desired frequency for a specified time duration. Control over the sound pitch is not quite as precise when specifying the tone frequency as it is when specifying the tone period. This routine sets the values of the PSG tone period registers (both coarse & fine) which correspond to the specified channel.

4.7.5. The JOYSTK subroutine

Function

Read data from a joystick

Usage

CALL JOYSTK(joystick,data)

Input parameters

joystick

The number of the desired joystick (1 or 2). Joystick 1 is attached to the connector at J3 and joystick 2 is attached to the connector at J4. The PSG I/O port corresponding to the specified joystick is automatically enabled for input.

Output parameters

data

The data value read from the specified joystick. The data value is the combination of the joystick input switch settings which are defined as follows:

RIGHT = 1, LEFT = 2, DOWN = 4, UP = 8, FIRE = 16.

Note that RIGHT/LEFT and DOWN/UP are mutually exclusive inputs (can't occur at the same time).

Description

This routine provides a convenient way to read the PSG I/O port data registers.

4.8. Low-Level Programming Considerations

An alternative to using the five assembly language subroutines is to program the P-SST card at the lowest level, using port level I/O. Examples of this technique are provided by the VOTRAX.BAS, PSG.BAS, CLOCK.BAS, and SPACEWAR.BAS programs. Keep the following points in mind when programming the P-SST card in this manner:

Votrax Speech Chip

o Wait until the SC01 chip is not busy after sending it a phoneme; test the status bit provided for this purpose.

o Provide a time delay of about 20 milliseconds after each phoneme is spoken; this provides a normal rate of speech.

Programmable Sound Generator

o All programming is done by first selecting the desired PSG register, and then reading or writing the data to or from the PSG data port.

o The clock frequency used by the P-SST card is 2 x 10**6 Hz. This value must be used when calculating the tone, noise, and envelope period values (see the PSG data sheet and below).

o The PSG tone frequency = 125000 / tone period. Since the tone period ranges from 1 to 4095, the tone frequency ranges from 30.5 Hz to 125 KHz.

o The PSG noise frequency = 125000 / noise period. Since the noise period ranges from 1 to 31, the noise frequency ranges from 4032 Hz to 125 KHz.

o The PSG envelope frequency = 7812.5 / envelope period. Since the envelope period ranges from 1 to 65535, the envelope frequency ranges from 0.12 Hz (8.33 sec) to 7812.5 Hz (1.28 x 10**-4 sec).

o The PSG I/O port and noise/tone mixer enable register bit values are complemented (active low).

o Unless the amplitude is variable (under control of the envelope control registers), you must pause the execution of your program for a specific time interval in order to generate a sustained tone or noise. Even if you use the envelope control method, you must still disable tone and/or noise at the appropriate time.

Real Time Clock

o All programming is done by first selecting the desired clock register, and then reading or writing the data to or from the clock data port.

o When writing data to the clock, keep in mind that the clock expects to receive data in a hexadecimal format. You can use the following conversion formula to convert base 10 data to base 16 for the clock: data16 = 16 * int(data10/10) + mod(data10,10)

o When reading data from the clock, keep in mind that the clock sends out data in a hexadecimal format. You can use the following conversion formula to convert base 16 clock data to base 10: data10 = 10 * int(data16/16) + mod(data16,16)

Joysticks

o You must be sure that the appropriate PSG I/O port (for the desired joystick) is enabled for input, prior to reading the port value.

o Data read from the joystick is complemented (active low), so you will need to take the one's complement of the input data.

4.9. PSST.LIB & GRAPHICS.LIB linking libraries

Due to the inherent limitations of the Z-BASIC interpreter when medium to large scale program development is being done, it is recommended that the PSST.LIB and GRAPHICS.LIB linking libraries be used with a high level compiled language. There are versions of these libraries for most of the popular language compilers which operate under the Z-DOS operating system, including BASIC, PASCAL, FORTRAN, and C. Over 50 subroutines and functions are included in these libraries, which enable the application programmer to take full advantage of the capabilities of the P-SST card and the Z-100 graphics capabilities using the language of choice. Source code for most of the demonstration and utility programs is included with the P-SST support libraries to illustrate the use of the libraries.

THESE LIBRARIES CAN BE PURCHASED FROM THE DEALER WHERE YOU PURCHASED YOUR P-SST CIRCUIT CARD. PLEASE CONTACT THEM FOR ANY INFORMATION.

4.10. PSSTDEFS.ASM Assembly Language File

The following file is designed to be included into a user-written assembly language program which accesses the P-SST card at the port level. The manufacturers data sheets for the PSG, Votrax SC01 chip, and the Real Time Clock should be consulted for additional details on low level programming.

```
;       Programmable Sound Speech and Time (P-SST) Definitions

;       Gen'l Instruments AY-3-8910 Programmable Sound Generator
;       Votrax SC01 Phoneme Speech Synthesizer
;       National Semiconductor MM58167 Real Time Clock

;       Version 1.6 (c) by John Stetson 01-Dec-83

;       P-SST Port Definitions (relative to base port)

PSGREG  EQU     0       ;PSG Register Select Port (0-15)
PSGDAT  EQU     1       ;PSG Data and Control Port
VOTCTL  EQU     2       ;Votrax Control and Status Port
VOTENF  EQU     3       ;Votrax Inflection Port (0-3)
CLKREG  EQU     4       ;Clock Register Select Port (0-23)
CLKDAT  EQU     5       ;Clock Data and Control Port
VOTFHO  EQU     6       ;Votrax Phoneme Data Port (0-63)
CONFIG  EQU     7       ;P-SST Configuration Status Port
```

```
;       PSG Register Definitions

PSGATF  EQU     0       ;Channel A Tone Period (Fine - 8 bits)
PSGATC  EQU     1       ;Channel A Tone Period (Coarse - 4 bits)
PSGBTF  EQU     2       ;Channel B Tone Period (Fine - 8 bits)
PSGBTC  EQU     3       ;Channel B Tone Period (Coarse - 4 bits)
PSGCTF  EQU     4       ;Channel C Tone Period (Fine - 8 bits)
PSGCTC  EQU     5       ;Channel C Tone Period (Coarse - 4 bits)
PSGNP   EQU     6       ;Noise Period Control (5 bits)
PSGEN   EQU     7       ;I/O Port and Noise/Tone Mixer Enable
PSGAA   EQU     8       ;Channel A Amplitude Control (5 bits)
PSGBA   EQU     9       ;Channel B Amplitude Control (5 bits)
PSGCA   EQU     10      ;Channel C Amplitude Control (5 bits)
PSGEPF  EQU     11      ;Envelope Period (Fine - 8 bits)
PSGEPC  EQU     12      ;Envelope Period (Coarse - 8 bits)
PSGESC  EQU     13      ;Envelope Shape/Cycle Control (4 bits)
PSGIOA  EQU     14      ;I/O Port A Data (5 bits for joystick)
PSGIOB  EQU     15      ;I/O Port B Data (5 bits for joystick)

;       PSG Enable Register Bit Definitions

PSGEIB  EQU     80H     ;NOT Enable I/O Port B Input
PSGEIA  EQU     40H     ;NOT Enable I/O Port A Input
PSGENC  EQU     20H     ;NOT Enable Channel C Noise
PSGENB  EQU     10H     ;NOT Enable Channel B Noise
PSGENA  EQU     08H     ;NOT Enable Channel A Noise
PSGETC  EQU     04H     ;NOT Enable Channel C Tone
PSGETB  EQU     02H     ;NOT Enable Channel B Tone
PSGETA  EQU     01H     ;NOT Enable Channel A Tone

;       PSG Amplitude Control Register Bit Definitions

PSGAMD  EQU     10H     ;Amplitude Mode (0/1 = fixed/variable)

;       PSG Envelope Control Register Bit Definitions

PSGCNT  EQU     08H     ;Continue  Envelope Cycle Pattern
PSGATT  EQU     04H     ;Attack    Envelope Cycle Pattern
PSGALT  EQU     02H     ;Alternate Envelope Cycle Pattern
PSGHLD  EQU     01H     ;Hold      Envelope Cycle Pattern

;       PSG I/O Port Register Bit Definitions (for joysticks)

JOYRGT  EQU     01H     ;Joystick Right
JOYLFT  EQU     02H     ;Joystick Left
JOYDOW  EQU     04H     ;Joystick Down
JOYUP   EQU     08H     ;Joystick Up
JOYFIR  EQU     10H     ;Joystick Fire
```

```
;       Speech Synthesizer Status Port Bit Definitions

VOTBSY  EQU     80H     ;NOT Votrax Busy

;       Speech Synthesizer Phoneme Data Port Bit Definitions

VOTPA0  EQU     03H     ;No sound for  47 msec
VOTPA1  EQU     3EH     ;No sound for 185 msec
VOTSTO  EQU     3FH     ;No sound for  47 msec

;       Real Time Clock Register Definitions

CLKCTH  EQU     0       ;Counter - Thousandths of Seconds
CLKCHT  EQU     1       ;Counter - 100th's & 10th's of Seconds
CLKCSC  EQU     2       ;Counter - Seconds
CLKCMN  EQU     3       ;Counter - Minutes
CLKCHR  EQU     4       ;Counter - Hours
CLKCDW  EQU     5       ;Counter - Day of the Week
CLKCDM  EQU     6       ;Counter - Day of the Month
CLKCMO  EQU     7       ;Counter - Months
CLKLTH  EQU     8       ;Latches - Thousandths of Seconds
CLKLHT  EQU     9       ;Latches - 100th's and 10th's of Seconds
CLKLSC  EQU     10      ;Latches - Seconds
CLKLMN  EQU     11      ;Latches - Minutes
CLKLHR  EQU     12      ;Latches - Hours
CLKLDW  EQU     13      ;Latches - Day of the Week
CLKLDM  EQU     14      ;Latches - Day of the Month
CLKLMO  EQU     15      ;Latches - Months
CLKIS   EQU     16      ;Interrupt Status
CLKIC   EQU     17      ;Interrupt Control
CLKCR   EQU     18      ;Counter Reset
CLKLR   EQU     19      ;Latch Reset
CLKST   EQU     20      ;Status
CLKGO   EQU     21      ;GO Command
CLKSI   EQU     22      ;Standby Interrupt
CLKTST  EQU     23      ;Test Mode

;       P-SST Configuration Port Bit Definitions

JOY1    EQU     01H     ;Joystick 1 installed
JOY2    EQU     02H     ;Joystick 2 installed
UNDEF1  EQU     04H     ;Not defined
UNDEF2  EQU     08H     ;Not defined

;       End of P-SST Definitions
```

4.11. Recommended Software Add-Ons

The following software products are currently under development by Software Wizardry, Inc. and LP Systems, Inc. - and should be available from your P-SST dealer in the very near future:

CP/M P-SST Distribution Disk
P-SST Libraries (CP/M)
P-SST Libraries (Z-DOS)
P-SST Three Voice Music Editor (Z-DOS)

Additionally, the following existing products are now undergoing modification for P-SST support, as described, and will be available from your dealer in revised form once completed. Registered owners of the current versions of these products will be notified automatically once these are available via postcard, and will be able to purchase the updates at a substantially reduced price:

CHRONOLOGIC (Clock/Calendar Support)
PALETTE (Joystick Support)
REACTOR-100 (Sound/Voice Support)
ZLYNK/II (Clock/Calendar Support)

5. CLOCK/CALENDAR

5.1. Overview

The P-SST time function provides your computer with a four year real-time clock/calendar with battery back-up power. Additional features provided by the P-SST time function include an "alarm" mode interrupt, a "heartbeat" interrupt and a "standby" interrupt which can be programmed to activate even when your computer is turned off. The following paragraphs describe the capabilities for these features of your P-SST time function.

Real-Time Clock/Calendar

The real-time clock/calendar is time generated by a ripple counter capable of tracking time in increments from ten thousandths of a second to months. This counter may be set and initialized to a precise present time and date. Once set and initialized, the counter will continuously track the precise time and date, even when power is turned off, for the duration of the back-up power battery's life. As stated in the installation section of this guide, the life of the battery is nominally 9 months but experience has shown that typically a battery should last well over a year.

The counter circuitry in the real-time clock is not configured to compensate for the extra day (February 29) in leap years. This inconvenience, along with the inconveniences posed by the normal spring and fall shifts between standard and daylight savings time may be corrected by including modules in your application programs for handling these exceptions.

The real-time clock/calendar counter may also be used in elapsed time applications. This is accomplished by setting and initializing the counter to zero. The mechanics for accomplishing this will be shown later in the functional description section of the real time clock.

"Alarm" Mode Interrupt

In the "alarm" mode, the P-SST time function utilizes a comparison latch. This latch may be set to some exact time in the future. When the value of the real-time clock/calendar and the value stored in the comparison latch are equal, an interrupt is applied to your computer's S-100 vectored interrupt buss (if a shorting plug is positioned on one of the dual-post J6 through J13 connectors). There are eight (8) user selectable interrupt priorities provided on your P-SST Card. As described in the installation section of this guide, the installation position of the shorting block interrupt interface determines the generated interrupt priority. Figure 3-8 defines the vectored interrupt selected for each position of the shorting block. Refer to the documentation provided with your computer for

software implementation of the vectored interrupts.

"Heartbeat" Interrupt

The "heartbeat" interrupt provides your computer with seven (7) programmable clock tick interval interrupts as follows:

- o Once each 1/10 of a second
- o Once each second
- o Once each minute
- o Once each hour
- o Once each day (at midnight)
- o Once each week (at midnight Saturday)
- o Once each month (last day at midnight)

This interrupt is also applied to your computer's S-100 vectored interrupt buss through the same shorting block interface described for "alarm" mode above.

"Standby" interrupt

The "standby" interrupt utilizes the comparison latch in the same manner as described for the "alarm" mode interrupt. However, unlike the "alarm" mode and "heartbeat" interrupts, the "standby" interrupt is an external interrupt. That is, the "standby" interrupt is sent to the "outside world" rather than to the computer's S-100 vectored interrupt buss. This feature provides your computer with the capability of activating other equipment, under the control of your programs, when this interrupt is interfaced through appropriate circuitry. Properly interfaced, this interrupt will enable your computer to turn itself on at a time preprogrammed by you.

5.2. Functional Description

The Real-Time Clock consists of four major sections:

- o Buss Control/Buffer
- o Clock Register Address Latch/Selector
- o Counters/Registers
- o Clock Address Decoder

The Buss Control/Buffer allows data to be transferred into or out of the selected register any time the Clock Address is selected and the clock is not in the "Power Down" mode. The Clock Register Address Latch/Selector holds the address code for the function that will be read from or written to. The Counters/Registers are the timing and storage components of the real time clock. The Clock Address Decoder examines the S-100 buss to determine when an interchange of data should take place for the clock. It then

selects the clock as the active function on the P-SST card.

Function Select and Buss Control

The clock/calendar function is selected (in Z-BASIC) by an INP or OUT command followed by the address port or data port designation of P-SST clock/calendar function. These port designations are:

```
ADDRESS PORT = "P-SST BASE ADDRESS" + 4
DATA PORT = "P-SST BASE ADDRESS" + 5
```

When the address port is selected, the P-SST Buss Controller / Decoder is conditioned to interpret the information on the data buss as a register select instruction. There are 24 registers associated with the P-SST clock/calendar function. These registers are defined in Figure 5-1 and are also given in Figure 5-5. The register number is the register select data which is written to relative port 4 to enable a particular clock register for input or output.

When the data port is selected, the P-SST Buss Controller/Decoder connects the data buss to the previously selected register for the read/write operation.

REGISTER NUMBER	REGISTER FUNCTION
	Clock Counter Registers
0	Thousandths of a Second.
1	Hundredths and Tenths of a Second.
2	Seconds.
3	Minutes.
4	Hours.
5	Day of week.
6	Day of Month.
7	Months.
	Interrupt Comparison Latch Registers
8	Thousandths of a Second.
9	Hundredths and Tenths of a Second.
10	Seconds.
11	Minutes.
12	Hours.
13	Day of Week.
14	Day of Month.
15	Months.
	"Heartbeat" Interrupt Registers
16	Interrupt Status.
17	Interrupt Control.
	Counter/Latch Reset Registers
18	Counter Reset
19	Latch Reset
	Special Purpose Registers
20	Status Bit
21	"GO" Command
22	Standby Interrupt
23	Test Mode

Fig. 5-1: Clock Register Definitions

To get the time and day information from the clock, the data in the counter registers must be read and interpreted. There are a total of eight counters as part of the clock, ranging from ten thousandths of seconds to a month. The format for these counters is as shown in Figure 5-4. Note this information is in binary coded decimal(BCD). As an example, we will set the clock for 16 December 1984 at 3:20 p.m. The number 12, representing December, will need to be converted to its BCD equivalent and stored in register 7 of the clock. We must first enable register 7 of the clock to receive the month data we are about to write to it. For this, we write the number 7 to address port 4 of the P-SST. Assuming that the P-SST relative address port selector is set to zero by DIP switch DS1. Then the data, binary number of 00010010(BCD encoded 12), will be written to address port 5 of the P-SST. Similarly, the day of the month, 16, will be placed in register 6 of the clock. First, write the number 6 to address port 4. This enables register 6 of the clock to receive the data that will be written next to address port 5. In this case, 000010110 (binary) will be written to port 5. The remaining clock registers, 0 through 5, are written following the same procedure as just described.

The Z-BASIC instructions given below will do the entire setup:

```
OUT 4,7   'enable reg 7
OUT 5,18  'write BCD 12 to reg 7 as month
OUT 4,6   'enable reg 6
OUT 5,22  'write BCD 16 to reg 5 as day of month
OUT 4,5   'enable reg 5
OUT 5,1   'write Sunday as first day of week
OUT 4,4   'enable reg 4
OUT 5,21  'write BCD 15 for 15th hour of 24 hour clock
OUT 4,3   'enable reg 3
OUT 5,32  'write BCD 20 to reg 3 as minutes
OUT 4,2   'enable reg 2
OUT 5,0   'write 0 to reg 2 for seconds
OUT 4,1   'enable reg 1
OUT 5,0   'write 0 to reg 1 for hundredths/tenths sec
OUT 4,0   'enable reg 0
OUT 5,0   'write 0 to reg 0 for ten thousandths of sec
```

Note that for 3 p.m. the hours register is set to 15 for the 24 hour clock.

To read the information from the clock a similar sequence of instructions is needed. Only this time, input instructions are used to read the data from previously enabled clock registers.

Z-BASIC instructions to read the months register might look like the following:

```
OUT 4,7     'enable register 7
D = INP(5) 'read data from previously selected register
           'data will be stored in variable D
```

Invalid data will be read from any of the real time counters if the counters are read while the clock rolls over. Therefore, address port 20 should be read to determine if the previous counter read is valid. The status bit (D0) will be set if an erroneous reading took place. If set, then a reread of that counter will be necessary.

As indicated in figure 5-5, most registers contain unused bits. When reading these registers the unused bits will be logical zero. A write to any unused bit will be ignored.

The time may be precisely set to the second using the GO command built into the clock. Specifically, each register (except for seconds, tenths and hundredths of second and ten thousandths of seconds) should be loaded with the desired time to the nearest upcoming minute then a GO command given. This will reset the seconds, tenths and hundredths of seconds and then thousandths of seconds. Thus, initializing the clock to a specific time. The GO command is given by writing to register 21. The data accompanying this instruction has no significance. Therefore any data written to register 21 will be discarded. A GO command in Z-BASIC may be written:

```
OUT 4,21  'select register 21 to be written to
OUT 5,0   'the zero data has no significance
```

During a GO command, if the seconds counter is greater than 40 the minutes register will be incremented, otherwise the minutes register is not affected. Then, for setting the clock to a precise time with the GO command insure that the seconds counter is less than 40. See Figure 5-2 for a pictorial representation of the "GO" logic.

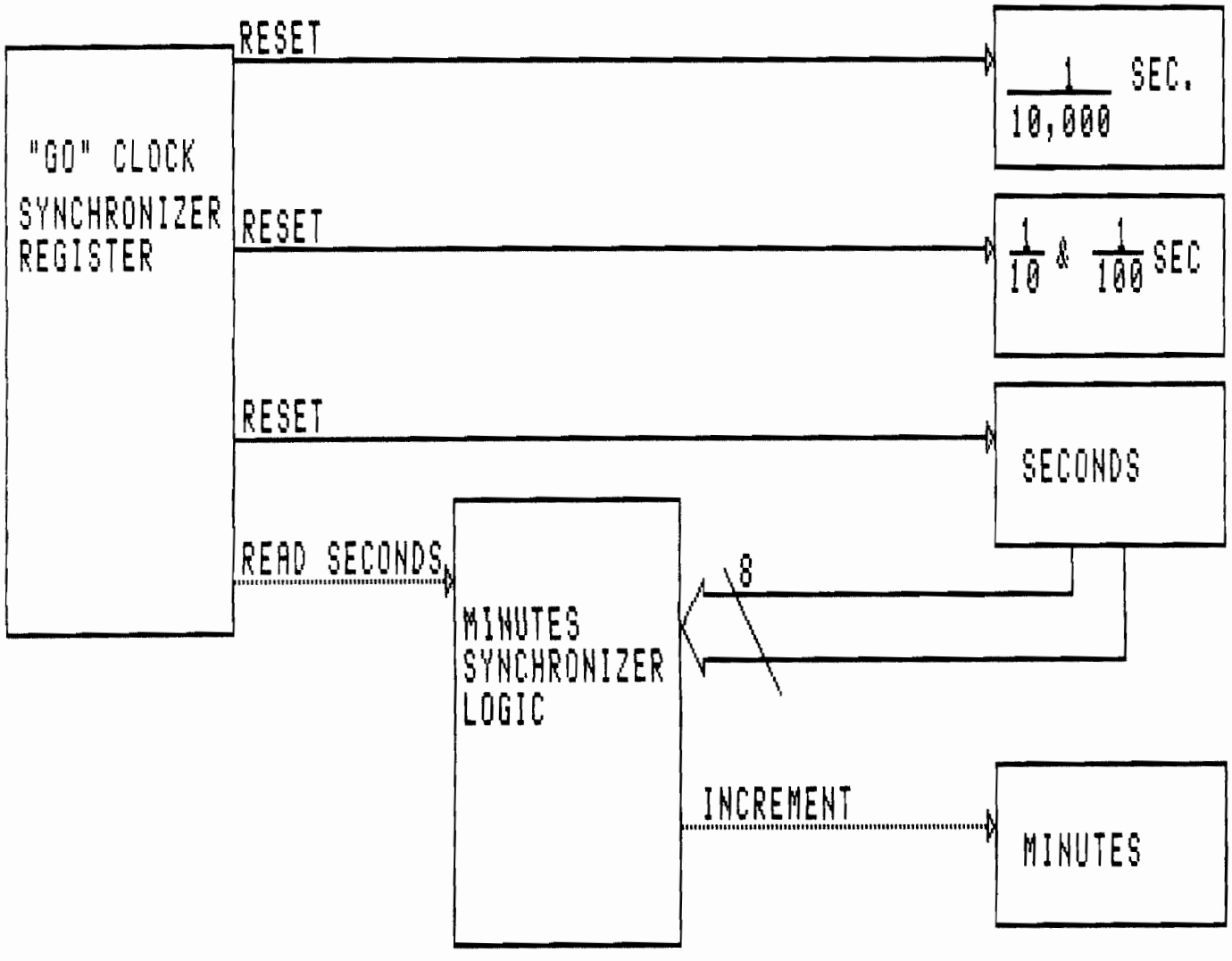


Fig. 5-2: GO Logic and Counter Synchronization

5.3. Resetting the Clock/Calender Counter

Registers 0 through 7, the counter registers, may be reset to zero by writing 255 decimal (all ones) to register 18(decimal) of the clock. As mentioned in the overview, this may be a convenient method for initializing the counters in elapsed time applications. Configured this way, the clock could be used to time events up to 1 year in duration. The counter (registers 0 through 7) may be reset to zero in its entirety or selected registers of the counter may be zeroed. The following command will reset all counter registers to zero:

```
OUT 4,18 :OUT 5,&HFF
```

The first part of the command (OUT 4,18) selects register 18 (counter reset register) as the register to receive the data given in the second half of the command (OUT 5,&HFF). The OUT 5,&HFF sets bits 0 through 7 of register 18 to a logic 1, thus resetting counter registers 0 through 7. Note that the above command assumes that the P-SST BASE ADDRESS is set to 0 as suggested in section 3 of this guide. Figure 5-3 represents the logic required for resetting individual registers of the clock counters. The following table shows the bit assignment in Register 12 :

Bit D7 -- Months
Bit D6 -- Days of the Month
Bit D5 -- Days of the Week
Bit D4 -- Hours
Bit D3 -- Minutes
Bit D2 -- Seconds
Bit D1 -- Tenths and Hundredths of Seconds
Bit D0 -- Thousanths and Ten Thousandths of Seconds

Register 18 is the Counter Reset Register
Register 19 is the Latch Reset Register

Table 5-1: Latch and Counter Reset Bit Assignments

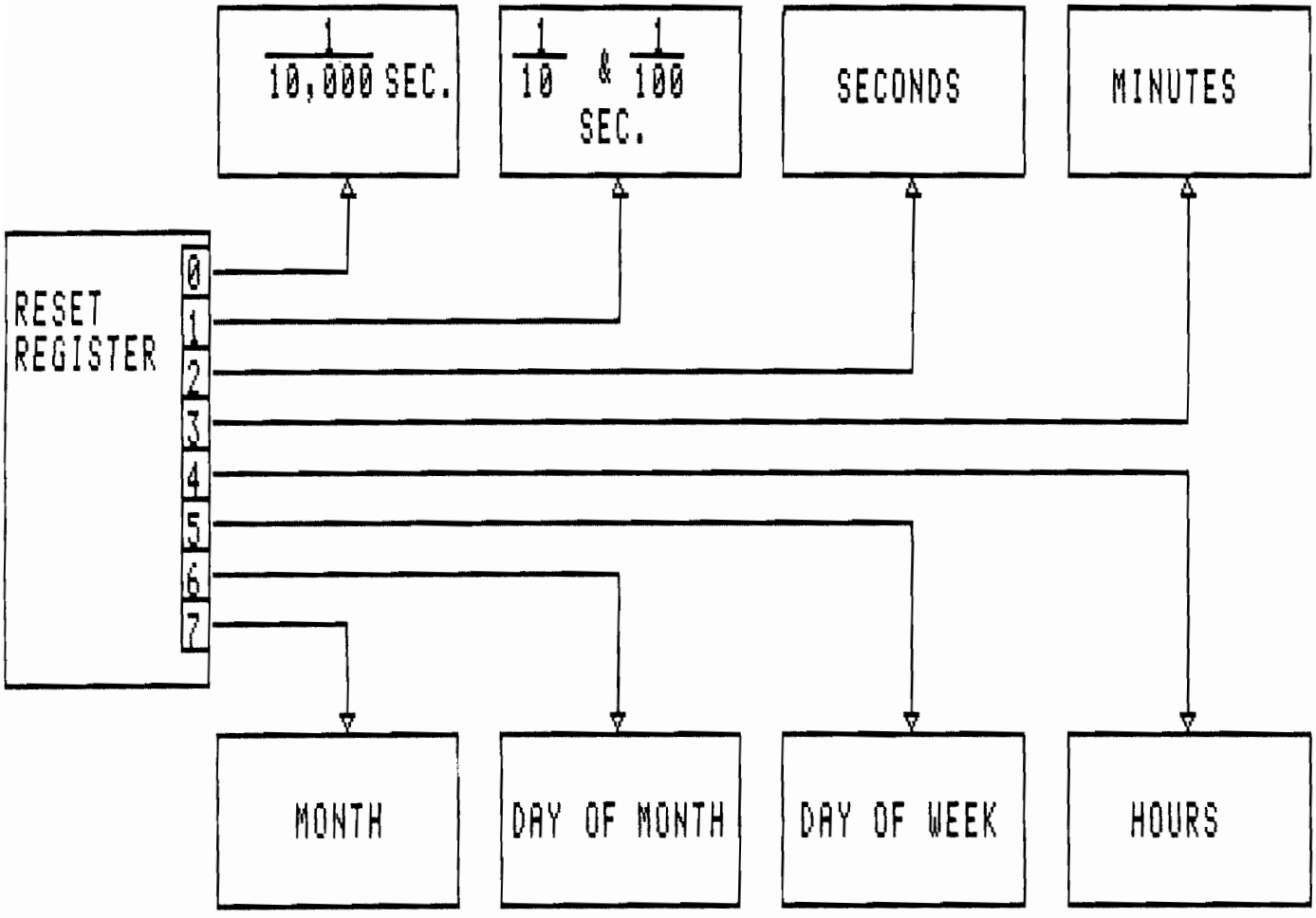


Fig. 5-3: Counter/Comparator Reset

5.4. Resetting The Latch Comparator

All of the latch comparator registers (8 through 15) may be reset to zero or selected registers of the comparator may be zeroed. The following command will reset all registers of the latch comparator:

```
OUT 4,19 :OUT 5,&HFF
```

The first part of the command (OUT 4,19) selects register 19 (latch reset register) as the register to receive the data given in the second half of the command (OUT 5,&HFF). The OUT 5,&HFF sets bits 0 through 7 of register 19 to logic 1. This resets comparator registers 8 through 15. Note that the above command assumes that the P-SST BASE ADDRESS is set to 0 as is suggested in section 3 of this guide. Page 5-10 shows the bits required for resetting individual registers of the clock comparator.

To reset the minutes and hours in the counter register the

following commands could be used:

```
OUT 4,18    'enables counter reset register to get data
OUT 5,&H14  'sets bits D4 and D3 in counter reset reg.
```

5.5. Clock Memory/RAM

There are eight registers (8 through 15) which may be used to hold information. Normally these registers are used to hold alarm data but may be used for data storage. They are occasionally referred to as RAM registers. Due to the battery backup, they will retain their contents while your system is powered off.

5.6. Alarm Mode

The main purpose for the RAM registers is to hold alarm set information for comparison purposes when in the alarm mode. In the alarm mode the clock has the capability of issuing an interrupt to the S-100 buss via one of the vectored interrupts at J6 through J13. During power down this alarm can be detected at the Standby Interrupt connector J5, provided that the Standby Interrupt is enabled. The procedure for setting the alarm registers is similar to that of setting the time in the counter registers. Once the alarm is set the "compare for alarm" can be activated by writing a data value of 1 to register 17 (Interrupt Control Register) of the clock. This activates the comparison logic which sets bit 1 in register 16 (Interrupt Status Register) when a match is found in the corresponding counter registers 0 through 7. Note that the unused bits in the counters will only compare with zeros in the RAM registers. The Interrupt Status Register is a read only register. The ten thousandths of seconds and days of the week are not compared because they are not used in their respective counter registers.

Each four bit BCD digit in the RAM registers can be forced to compare with its corresponding counter register digit by setting the two most significant bits of that digit in the applicable RAM register to ones. For example, if an alarm was needed every "Monday" at a specific time, the "Month" and "Day of the Month" would need to compare all the time. Therefore, bits D2, D3, D6 and D7 would be written to clock registers 6 and 7 (Month and Day of the Month).

	D7	D6	D5	D4	D3	D2	D1	D0
Day of the Month register 6	1	1	0	0	1	1	0	0
Month register 7	1	1	0	0	1	1	0	0

In other words, 204 decimal would be written to registers 6 and 7. By using this feature, practically any interval of time can be

used to generate an alarm.

5.7. Heartbeat Interrupt

The clock function also has the ability to generate a heartbeat interrupt (a periodic interrupt). That is, the frequency of an interrupt can be established by setting the appropriate bit in Register 17, the Interrupt Control Register. The bit definition for this register is shown below.

Interrupt Control Bits	Definitions
D7	Compare/not heartbeat
D6	1/10 second
D5	1 second
D4	1 minute
D3	1 hour
D2	1 day
D1	1 week
D0	1 month

Once a bit has been set in the Interrupt Control Register, the rollover of the counter, corresponding to the frequency selected, will cause a logic 1 interrupt indication on the appropriate bit in the Interrupt Status register. This interrupt can then be transferred to the S-100 buss via one of the vectored interrupts at J6 through J13 (VIO through VI7) if a shorting block is in position. The interrupt output to the S-100 buss is a logical ORing of all the bits in the Interrupt Status Register. If any bit is set the interrupt output will be active. The contents of the Interrupt Status Register may be read to determine the particular interrupt being generated. The Interrupt Status register is reset automatically after each time it is read.

5.8. Standby Interrupt

The Standby Interrupt is available to the outside world via connector J5 on the P-SST. It is triggered by a comparison between the RAM registers and their corresponding clock real-time counter registers. Generation of the Standby Interrupt can be done only during the "power down" mode and when the Standby Interrupt is enabled. The sequence of events would be to set the alarm for the proper time you wish to activate the external Standby Interrupt, then enable the Standby Interrupt. Then when powered off, the alarm can be detected externally through J5.

To enable the Standby Interrupt, a 1 is written to register 22. A 0 is written to register 22 to disable the Standby Interrupt.

Putting it another way, to enable the Standby Interrupt:

```
OUT 4,22  'enable the Standby Interrupt register to get data
OUT 5,01  'set the enable bit in the register
```

	TENS				UNITS			
COUNTER	D7	D6	D5	D4	D3	D2	D1	D0
Ten Thousandth Sec	1	1	1	1	-	-	-	-
Tenths/Hundredths Sec	1	1	1	1	1	1	1	1
Seconds	-	1	1	1	1	1	1	1
Minutes	-	1	1	1	1	1	1	1
Hours	-	-	1	1	1	1	1	1
Day of the Week	-	-	-	-	-	1	1	1
Day of the Month	-	-	1	1	1	1	1	1
Month	-	-	-	1	1	1	1	1

NOTE: the symbol (-) indicates not used, (1) indicates valid bit

Fig. 5-4: Counter Format Definition

D4	D3	D2	D1	D0	FUNCTION
0	0	0	0	0	Counter-Thousandths of Seconds
0	0	0	0	1	Counter-Hundredths and Tenths of Seconds
0	0	0	1	0	Counter-Seconds
0	0	0	1	1	Counter-Minutes
0	0	1	0	0	Counter-Hours
0	0	1	0	1	Counter-Day of the Week
0	0	1	1	0	Counter-Day of the Month
0	0	1	1	1	Counter-Months
0	1	0	0	0	Latches-Thousandths of Seconds
0	1	0	0	1	Latches-Hundredths and Tenths of Seconds
0	1	0	1	0	Latches-Seconds
0	1	0	1	1	Latches-Minutes
0	1	1	0	0	Latches-Hours
0	1	1	0	1	Latches-Day of the Week
0	1	1	1	0	Latches-Day of the Month
0	1	1	1	1	Latches-Months
1	0	0	0	0	Interrupt Status Register
1	0	0	0	1	Interrupt Control Register
1	0	0	1	0	Counter Reset
1	0	0	1	1	Latch Reset
1	0	1	0	0	Status Bit
1	0	1	0	1	"GO" Command
1	0	1	1	0	Standby Interrupt

Fig. 5-5: Register Selection Codes and Function

6. SOUND SYNTHESIZER

6.1. Overview

The P-SST sound function provides your computer with a three channel sound synthesizer capable of producing both music and complex sound effects. The programmable features of the P-SST sound function are summarized below and described in the following paragraphs:

- 14 Read/Write Control Registers
- Three Sound Generators
- One White Noise Generator
- Three Sound/Noise Mixers
- Three Amplitude Controllers
- One Envelope Generator

Read/Write Control Registers

An array of 14 read/write control registers (Figure 6-1) is used for programming all features of the P-SST Card sound function. When a register is selected (addressed), that register is latched (remains the selected register) until another register is selected. This feature eliminates the need for continuously readdressing the same register for successive write operations. In addition to latching the selected register address, the sound synthesizer also latches the data written to the selected register. This feature allows the synthesizer to maintain its output while your computer's central processor unit (CPU) performs other tasks. Definition of this register array and register addressing requirements are presented in the Functional Description portion of this section.

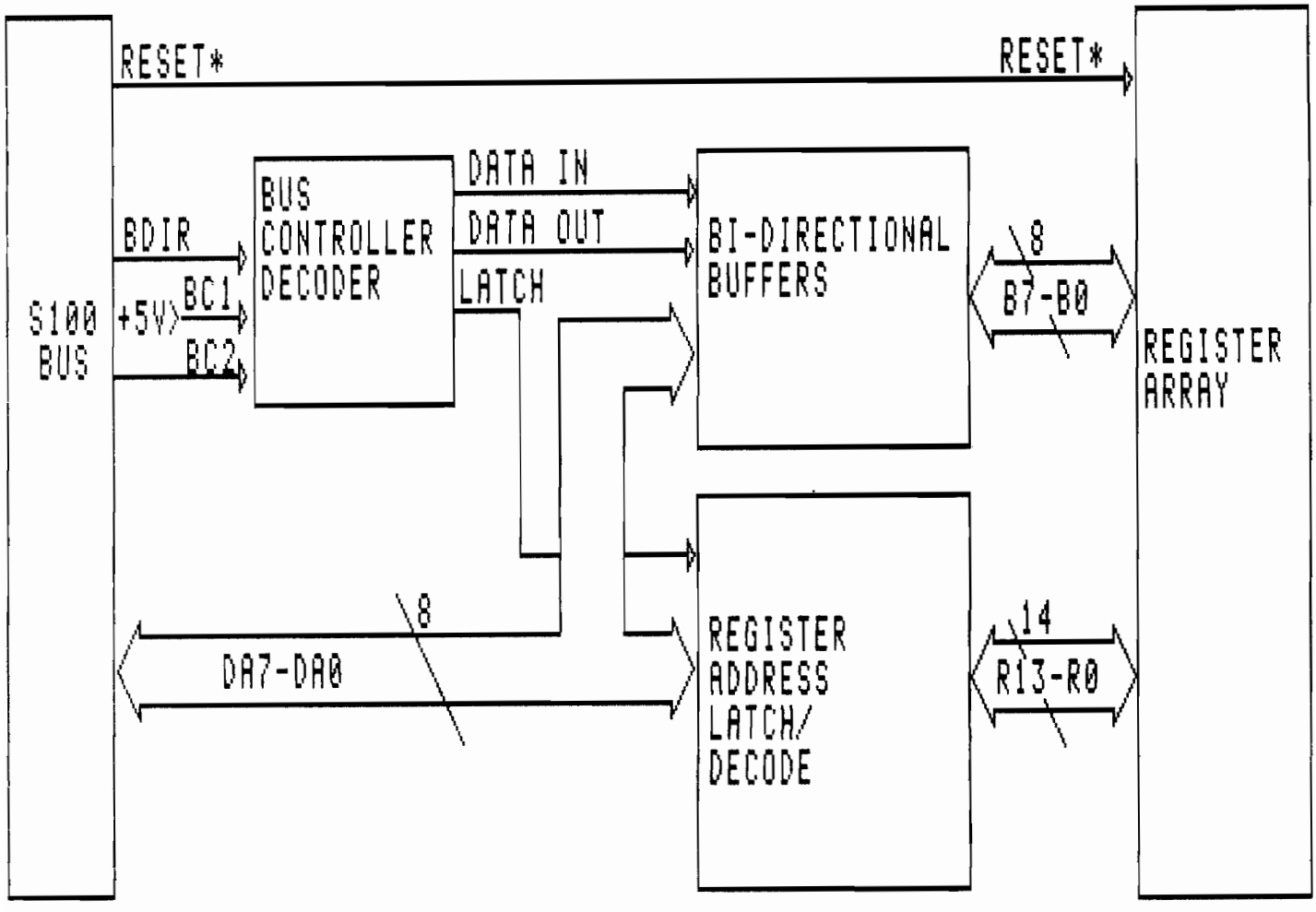


Fig. 6-1: Sound Synthesizer Register Array Interface

Sound Generators

Each of the three output channels (channel A,B & C) of the sound synthesizer contains a square wave sound generator. These sound generators are individually programmed, through the register array, to produce frequencies of from 30.5 Hz to 125 kHz kilohertz. This is accomplished by writing a tone period value into the control register associated with the desired sound generator. The tone period values required to produce eight (8) octaves of the musical scale are provided in the Functional Description portion of this section.

White Noise Generator

The noise generator produces a programmable frequency modulated pseudo-random pulse-width square-wave output. The frequency of the modulation is programmable from 4 kHz to 125 kHz. The output of the noise generator may be mixed with the output of any or all sound generator outputs.

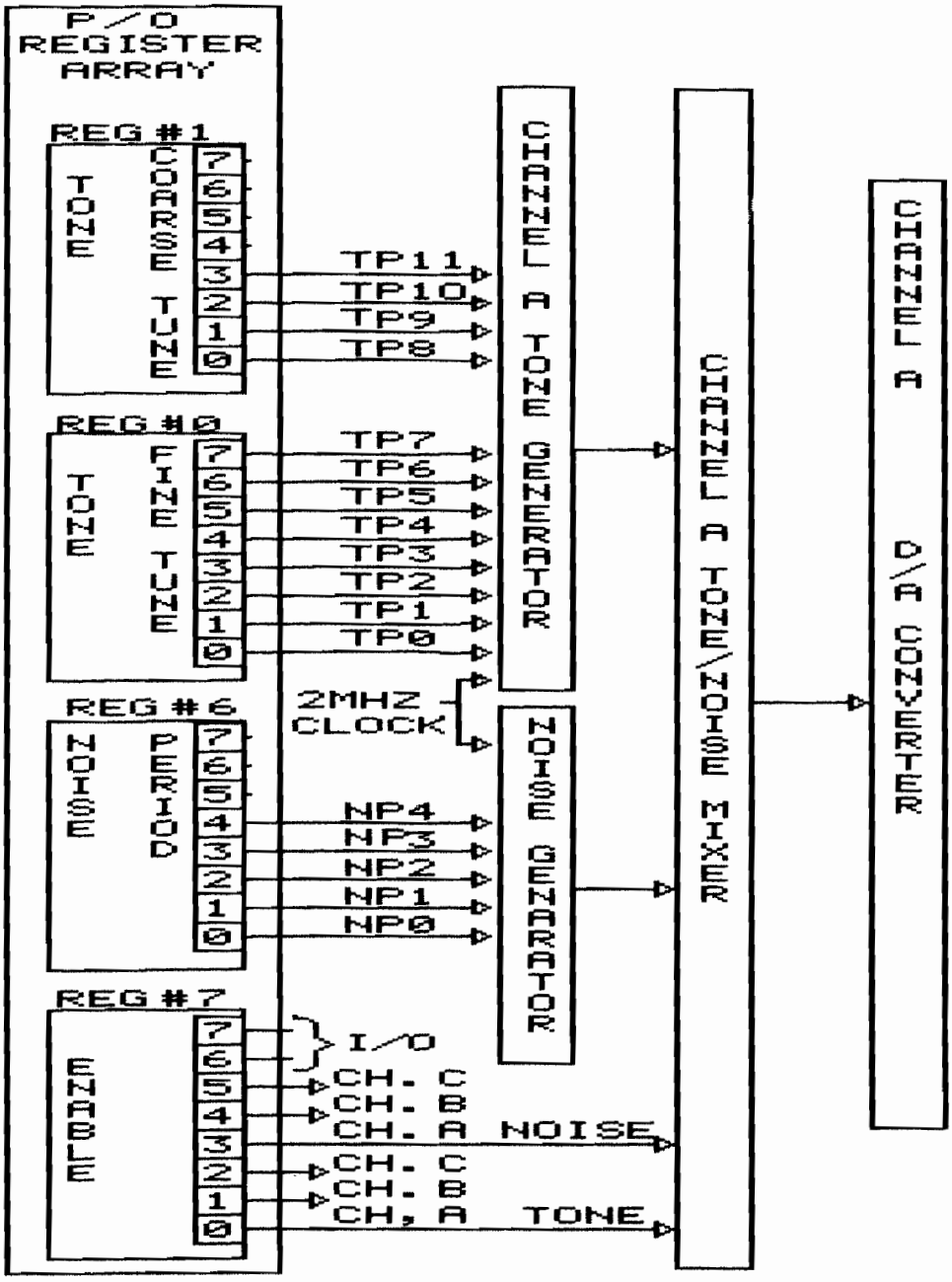


Fig. 6-2: Channel A (Typical) Tone & Noise Generators

Mixers

The mixers of the synthesizer determine the output produced by each channel. The possible selections for each channel are:

- Neither Sound nor Noise
- Either Sound or Noise
- Both Sound and Noise

With regard to the first programmable selection (Neither/Nor), you should be aware that selecting this programming option does not turn off the selected channel. A channel is turned off only when the amplitude control register for that channel is programmed for disable.

Amplitude Controllers

The amplitude controllers of channels A, B and C determine the source of amplitude control for the digital-to-analog (D/A) converters of their respective channels. The programmable selections are:

- Channel Disabled
- Channel Under Direct Program Control
- Channel Controlled By Envelope Generator

Under direct program control, the D/A converters may be programmed to any one of 16 fixed levels of amplitude. Under control of the envelope generator, the D/A converters are programmed to 16 varying amplitude levels of varying duration, with variable relative shape and cycle pattern. It is this combination of user programmable variables that enables the P-SST sound synthesizer to produce complex sound patterns.

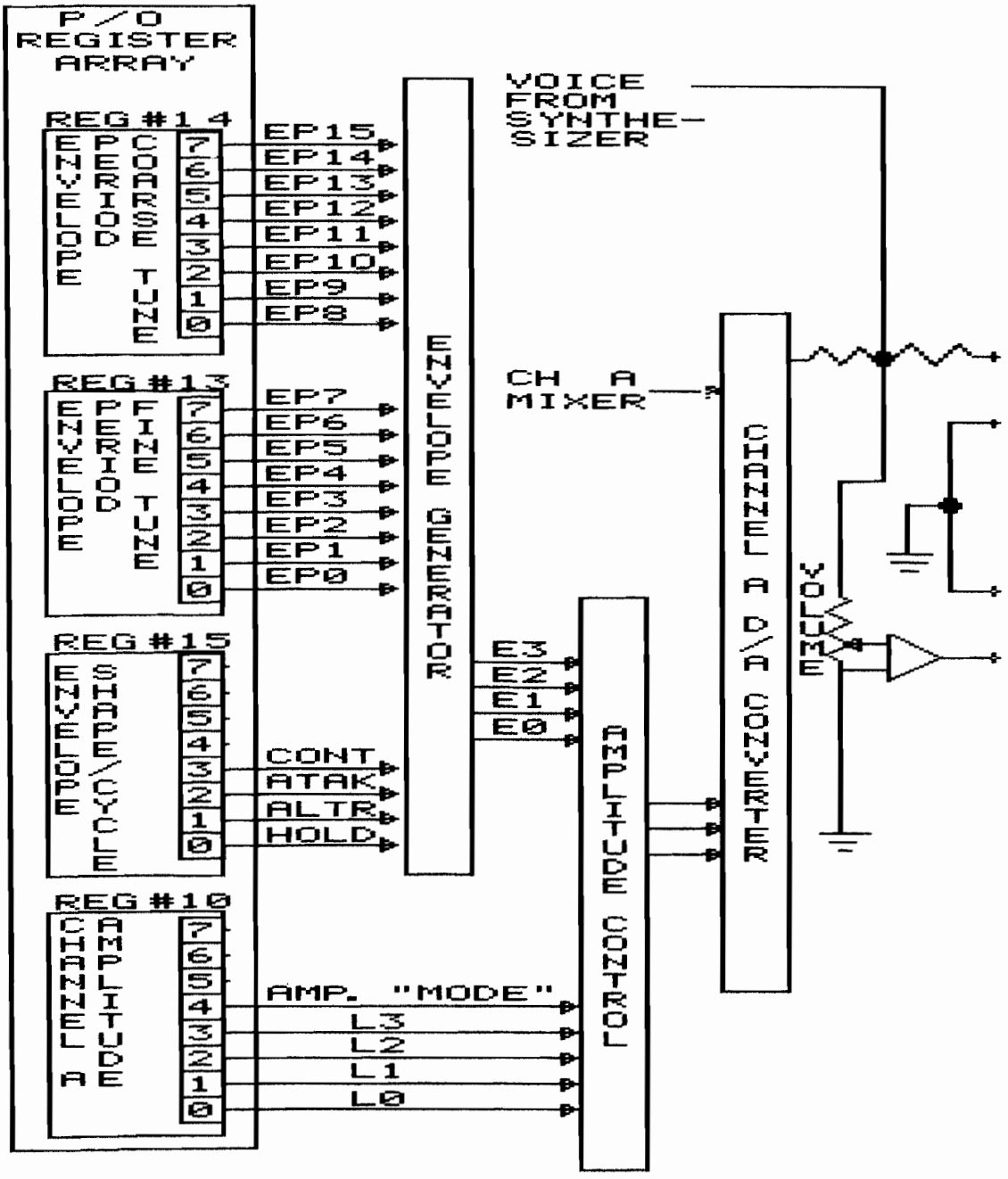


Fig. 6-3: Channel A (Typical) Amplitude Control

Envelope Duration

The duration of the envelope is determined by the envelope period. Longer envelope periods create more mellow sounds, whereas shorter envelope periods create more piercing sounds.

Envelope Shape

For any given envelope period, the shape of the envelope is either an attacking or a decaying shape or a combination of the two. An attacking envelope will start at minimum volume (which is no output) and build to maximum volume over the duration of that envelope period. A decaying envelope will start at maximum volume and drop to minimum volume over the duration of the envelope period. Using a combination of the attack and decay adds to the versatility of the synthesizer.

Envelope Cycle Pattern

The P-SST sound synthesizer is capable of generating a variety of envelope cycle patterns from the basic envelope shapes described above. These programmable envelope patterns include:

- o Single decay envelope period from maximum to minimum volume.
- o Single attack envelope period from minimum to maximum volume.
- o Single envelope decay period, return to sustained max volume.
- o Single envelope period of attack with return to min volume.
- o Continuously repeating envelope periods of decay.
- o Continuously repeating envelope periods of attack.
- o Alternating envelope periods of decay and attack.
- o Alternating envelope periods of attack and decay.

6.2. Functional Description

Relative ports 0 and 1 are used to program sound using the P-SST sound synthesizer. Remember, relative ports are based upon the current base address setting on DS1. Port 0 is used to select one of the fifteen available registers to accept data and port 1 is used to write or read data to or from the previously selected register. To produce sound or to communicate with the two ports, programming must be done in two steps. First select one of the fifteen available registers and then write data to that register. The format may look something like this:

```
OUT 0, register number to select
OUT 1, data to be sent to the register
```

The functional registers used in the programming of the sound synthesizer can be listed as follows:

- o Tone Generation
- o Noise Generation
- o Mixer Control
- o Amplitude Control
- o Envelope Control

6.2.1. Tone Generation - Registers 0,1,2,3,4,5

There are three tone channels (A,B,C) which can be mixed into the audio output. Two registers are used for each channel. One is for coarse tuning and the other is for fine tuning.

	4 BIT Coarse Tune	8 BIT Fine Tune
Channel A	Reg 1	Reg 0
Channel B	Reg 3	Reg 2
Channel C	Reg 5	Reg 4

A total of 12 bits make up the programming word for each of the tone generators. The tone frequency can be found as a function of the 12 bit tone generator word according to the following:

$$Ftone = Fclock / (16 * tp)$$

Where Fclock is the frequency of the master sound clock. In the P-SST this is 2 megahertz (2000000 cycles per second). The term "tp" is the decimal equivalent of the 12 bit value stored in the coarse and fine registers for that channel. The weighting of the registers is:

$$"tp" = (256 * coarse) + fine$$

For example, if register 0 contains 130 (decimal) and register 1 contains 10 (decimal) the calculation for the corresponding frequency on channel A is:

```
register 0 =   1 0 0 0 0 0 1 0  (binary)
register 1 =   0 0 0 0 1 0 1 0  (binary)
```

```
then,          tp = (256 * 10) + 130  =   2690
therefore,     tone frequency = Fclock / (16 * tp)
                              = 2000000 / (16 * 2690)
                              = 46 Hertz
```

Likewise, given a frequency, the values for the coarse and fine registers can be found. Example:

To find corresponding register values to produce a 1000 Hertz tone. First we must find the value for "tp", the 12 bit tone register value.

```
tp = Fclock / (16 * tone frequency)
   = 2000000 / (16 * 1000)
   = 125
```

The value of "tp", 125, is then the combination of the values of coarse and fine registers according to their weighting. That is,

```
tp = 125 = (256 * coarse reg) + fine reg
```

First we find the coarse register value. The number 256 will not divide into 125. Therefore, to satisfy the given equation, the coarse register value must be zero. This leaves the fine register value to be 125.

```
register 0 = 125 (decimal) =   01111101 (binary)
register 1 =   0           =   00000000 (binary)
```

6.2.2. Noise Generation - Register 6

The noise generation register can be used in the production of various waveforms and sounds which have random characteristics such as explosions, train sounds, motor sounds, gunshots, water running and so on. The noise generator produces a frequency modulated random pulse width square wave. This can be mixed with the tone generators to produce a wide variety of sound effects. The noise frequency can be found by using the following equation:

```
noise frequency = Fclock / (16 * "np")
```

Where "np" is the value contained in the 5 bit noise register and "Fclock" is the 2 Megahertz system clock. Derivation of the noise frequency given the system clock frequency and the noise period register value "np" is similar to the previous example involving the tone frequency. Note that the three most significant bits of the noise register are not used.

Noise Period Register 6

D7 D6 D5	D4 D3 D2 D1 D0
not used	5 bit noise value

6.2.3. Mixer Control - Register 7

The mixer control is part of register 7. It enables the noise generator on any or all of the channels. It also enables tone on any or all of the channels. Therefore either the tone for a particular channel can be enabled or the noise generator can be enabled on that channel or both. The mixer control register bits are defined as follows :

Mixer Control Register Bit Definition

Bit 7	I/O A enable - see I/O Control section
Bit 6	I/O B enable - see I/O Control section
Bit 5	Channel C Noise Enable
Bit 4	Channel B Noise Enable
Bit 3	Channel A Noise Enable
Bit 2	Channel C Tone Enable
Bit 1	Channel B Tone Enable
Bit 0	Channel A Tone Enable

A zero in a bit position represents an enable whereas a one represents a disable on the channel. Figure 6-2 shows the control register input to the mixer. For example, to enable only tone on the three channels, A, B and C, we would write zeroes into bits 0, 1 and 2 and would write ones into bits 3, 4 and 5.

```
OUT 0,7 'enable register 7 to receive data
OUT 1,&H38  'set bits 3,4,5 into register 7
```

Since the enable data is complemented, bits 0, 1 and 2 are active.

6.2.4. Amplitude Control - Registers 8, 9, 10

The Amplitude Control registers can adjust the output amplitude of their corresponding channel in two ways. One is through directly varying the channel output using four control bits in the applicable register. The other is by letting the envelope generator output be the control for the amplitude of that channel.

The bit assignments in the Amplitude Control registers are:

D7	-	Not Used
D6	-	Not Used
D5	-	Not Used
D4	-	Envelope Generator Controlled Amplitude
D3	-	Bit 4 of programmed amplitude
D2	-	Bit 3 of programmed amplitude
D1	-	Bit 2 of programmed amplitude
D0	-	Bit 1 of programmed amplitude

6.2.5. Envelope Control - Registers 11, 12, 13

Both the period and the shape of the sound envelope can be altered under program control. Registers 11 and 12 control the period of the envelope. Register 13 then controls the shape.

6.2.6. Envelope Period

The frequency of the envelope is calculated in much the same way as for the tone and noise generators. The following equations apply:

$$Efreq = Fclock / (256 * "ep")$$

Where Fclock is the master sound clock of 2 Megahertz, "ep" is the 16 bit envelope period registers (coarse and fine tune). That is,

$$"ep" = (256 * \text{coarse register}) + \text{fine register}$$

Register 11 is the fine tune register and register 12 is the coarse tune register. Efreq is the resultant calculated envelope frequency.

6.2.7. Envelope Shape

The envelope shape is controlled by the least significant four

bits of register 13. The register 13 bit definition is as follows:

Register 13 Bit Definition

D7	-	not used
D6	-	not used
D5	-	not used
D4	-	not used
D3	-	Continue
D2	-	Attack
D1	-	Alternate
D0	-	Hold

CONTINUE - if zero, will cause the envelope counters to go to zero and stay there after one cycle. A one will cause the envelope to function according to state of the HOLD bit.

ATTACK - if zero, will cause the envelope counters to count down from the maximum amplitude value to the minimum. A one will cause it to count from minimum to maximum.

ALTERNATE - if a one, then the D/A envelope counters reverse count after each cycle. If the HOLD bit is set then the counter will go to its initial count before stopping.

HOLD - if set to a one, will cause only one cycle of the envelope to be generated. After the cycle the envelope counter will maintain its last count in the cycle. This can be either a count of 1111 (binary) or 0000 (binary), depending on whether it was counting up or down.

Some experimentation is required to gain an understanding of the variety of sounds which may be produced using combinations of these sound shaping bits.

In Summary, the available sound registers are:

R0 - Channel A Tone Period
R1 - Channel A Tone Period
R2 - Channel B Tone Period
R3 - Channel B Tone Period
R4 - Channel C Tone Period
R5 - Channel C Tone Period
R6 - Noise Period
R7 - Mixer Control
R8 - Channel A Amplitude
R9 - Channel B Amplitude
R10- Channel C Amplitude
R11- Envelope Period
R12- Envelope Period
R13- Envelope Shape/Cycle

7. I/O PORTS

7.1. Overview

Your P-SST Card contains two general purpose parallel I/O ports (Port 1 and Port 2). Although these ports are primarily intended for joystick interfaces, they may be used for any application requiring a parallel I/O port. Port 1 and Port 2 are programmed by writing data to or reading data from one of two P-SST data registers. The ports are enabled by a third enable register.

7.2. Functional Description

The I/O function of your P-SST Card is selected through the use of registers within the sound synthesizer function. Again, the method used to get data into or out of these registers is by issuing an INP or OUT command followed by the address port or data port designation of the P-SST I/O function. These port designatons are as follows:

```
REGISTER SELECTION PORT = "P-SST BASE ADDRESS" + 0
DATA PORT = "P-SST BASE ADDRESS" + 1
```

When the Register Selection port is addressed, the P-SST is designed to interpret the information on the data buss as a register select instruction. The three I/O function control registers are as follows:

```
I/O PORT ENABLE REGISTER IS REGISTER 7
I/O PORT 1 DATA REGISTER IS REGISTER 14
I/O PORT 2 DATA REGISTER IS REGISTER 15
```

I/O Port Commands

The I/O ports are enabled for read/write by the following commands:

```
OUT 0,7 :OUT 1,&H80       'Port 1 set for INPUT
                          'Port 2 set for OUTPUT
OUT 0,7 :OUT 1,&H40       'PORT 1 set for OUTPUT
                          'PORT 2 set for INPUT
```

The first part of the command (OUT 0,7) selects register seven (7) as the register to receive the data given in the second half of the command (OUT 1,&H80 or OUT 1,&H40). The above commands assume that the P-SST BASE ADDRESS is set to zero as suggested in section 3 of this guide.

I/O Data Read and Write Commands.

Reading data from, or writing data to, the I/O Data Ports is accomplished with the following commands:

Read Commands

```
OUT 0,7  : OUT 1, &HBF ' set port 1 to input
OUT 0,14 : DATA = INP(1)  ' read data on port 1
OUT 0,7  : OUT 1, &H7F ' set port 2 to input
OUT 0,15 : DATA = INP(1)  ' read data on port 2
```

Where DATA will contain the value read from the port

Write Commands

```
OUT 0,7  : OUT 1, &H7F ' make port 1 an output
OUT 0,14 : OUT 1, DATA ' write data out port 1
OUT 0,7  : OUT 1, &HBF  'make port 2 an output
OUT 0,15 : OUT 1, DATA  'write data out port 2
```

Where DATA contains the value to be written to the port.

The above commands assume that the P-SST BASE ADDRESS is set to zero as suggested in Section 3 of this guide. If the base address is set to sixteen (16), for example, the write command for Port 1 would read as follows: OUT 16,14 :OUT 17, DATA.

8. SPEECH SYNTHESIZER

8.1. Overview

The P-SST speech function provides your computer with "a voice of its own". The programmable features of the P-SST Card speech function include:

o A directory of 64 Phonemes
o Instantaneous pitch control of voiced phonemes

Phoneme Directory

Phonemes are the building blocks of speech. The P-SST speech function implements a complete directory of the 64 phonemes which are required for an unlimited vocabulary. Appropriately, sequencing these phonemes will produce intelligible speech. Table 8-1 defines the 64 phonemes. Table 8-2 defines the categories into which each phoneme falls. The first six of these categories listed in table 8-2 are audible (fricative categories produce expired voice sounds). The last catagory listed in table 8-2 defines phonemes which produce no sound.

Pitch Control

There are four programmable pitch levels which may be used to alter the voice pitch for individual phonemes, words or phrases. The overall pitch (or tone of voice) of the speech synthesizer is controlled by the setting of variable resistor R26 as described in section 3 of this guide.

The phoneme code is the 6 bit value (listed in hexadecimal) which would be written to relative port 6 to produce that phoneme sound. The phoneme symbol is the standard Votrax symbology which is used to represent that phoneme. To predict the length of time which a phoneme will be voiced, the duration is given listed in milliseconds.

Phoneme Code	Phoneme Symbol	Duration(ms)	Example
00	EH3	59	jacket
01	EH2	71	enlist
02	EH1	121	heavy
03	PA0	47	*no sound*
04	DT	47	butter
05	A2	71	made
06	A1	103	made
07	ZH	90	azure
08	AH2	71	honest
09	I3	55	inhibit
10	I2	80	inhibit
11	I1	121	inhibit
12	M	103	mat
13	N	80	sun
14	B	71	bag
15	V	71	van
16	CH	71	chip
17	SH	121	shop
18	Z	71	zoo
19	AW1	146	lawful
20	NG	121	thing
21	AH1	146	father
22	OO1	103	looking
23	OO	185	book
24	L	103	land
25	K	80	trick
26	J	47	judge
27	H	71	hello
28	G	71	get
29	F	103	fast
30	D	55	paid
31	S	90	pass
32	A	185	day
33	AY	65	day
34	Y1	80	yard
35	UH3	47	mission
36	AH	250	mop
37	P	103	past
38	O	185	cold
39	I	185	pin
40	U	185	move
41	Y	103	any
42	T	71	tap
43	R	90	red

Table 8-1: Phoneme Definitions (Codes 00-43)

Phoneme Code	Phoneme Symbol	Duration(ms)	Example
44	E	185	meet
45	W	80	win
46	AE	185	dad
47	AE1	103	after
48	AW2	90	salty
49	UH2	71	about
50	UH1	103	uncle
51	UH	185	cup
52	O2	80	for
53	O1	121	aboard
54	IU	59	you
55	U1	90	you
56	THV	80	the
57	TH	71	thin
58	ER	146	bird
59	EH	185	get
60	E1	121	be
61	AW	250	call
62	PA1	185	*no sound*
63	STOP	47	*no sound*

Table 8-2: Phoneme Definitions (Codes 44-63)

Voiced			Voiced Fricat.	Voiced Stop	Fricative Stop	Nasal	No Sound
E	EH	AE	Z	B	T	M	PA0
E1	EH1	AE1	ZH	D	DT	N	PA1
Y	EH2	AH	J	G	K	NG	STOP
Y1	EH3	AH1	V		P		
I	A	AH2	THV				
I1	A1	AW					
I2	A2	AW1					
I3	AY	AW2					
UH	OO1						
UH1	R						
UH2	ER						
UH3	L						
O	IU						
O1	U						
O2	U1						
OO	W						

Table 8-3: Phoneme Categories

A few examples follow to show how the phoneme symbology can be written to represent the voicing of typical words:

o "A" ..... A1 AY Y
Corresponds to phoneme codes 20,21,29

o "ADDRESS" ..... AE1 EH3 D R EH1 EH3 S
Corresponds to phoneme codes 2F,00,1E,2B,02,00,1F

o "DISK" ..... D I1 I3 S K
Corresponds to phoneme codes 1E,0B,09,1F,19

o "INSTRUCTION" ..... I1 I3 N S T R UH1 K SH UH1 N
Corresponds to phoneme codes 0B,09,0D,1F,2A,2B,32,19,11,32,0D

o "LIST" ..... L I1 I3 S T
Corresponds to phoneme codes 18,0B,09,1F,2A

8.2. Functional Description

The speech synthesis portion of the P-SST is comprised of six major sections:

o Phoneme register
o Inflection register
o Status port
o Manual Inflection setting
o "Start Phoneme" strobe
o Audio-Voice output

8.2.1. Phoneme Register

As previously mentioned, there are 64 phonemes which are accessed through the use of a six bit code. The phoneme register accepts this 6 bit code when written to through relative port 6 on the P-SST card. After the phoneme is sent, the unit of speech (the phoneme) may be voiced by the speech circuitry through writing to relative port 2. This is the "Start Phoneme" strobe. The data written to this port is not meaningful. For example, to voice the sound which corresponds to the phoneme "A", execute the following Z-BASIC commands:

```
OUT 6,&H20 'set "A" code in phoneme register
OUT 2,0  'command "start phoneme" strobe
```

These commands produce a continuous "A" sound. To silence the speech circuitry, a "silence" phoneme must be written. This is:

```
OUT 6,&H3F 'set "silence" code in phoneme register
OUT 2,0  'command the phoneme to be processed
```

8.2.2. Voice Inflection Control

There are two ways to vary the inflection of the voice output on the P-SST. One is "on the fly" by changing two bits of data in the inflection register and the other is manually varying the voice inflection pot R26. R26 is normally set to a pitch which is pleasing to the listener. Once set, it can be changed to three other settings simply by writing data to relative port 3 on the P-SST. Only the least significant two bits have any affect on the inflection. This register (relative port 3) is write-only.

	Inflection Port Bit Definition							
	D7	D6	D5	D4	D3	D2	D1	D0
	--	--	--	--	--	--	--	--
Pitch 1	x	x	x	x	x	x	0	0
Pitch 2	x	x	x	x	x	x	0	1
Pitch 3	x	x	x	x	x	x	1	0
Pitch 4	x	x	x	x	x	x	1	1

x = don't care/not used

Pitch 1 is the lowest inflection, while Pitch 4 is the highest.

8.2.3. Speech Status Port

Phoneme timing varies for each phoneme voiced. Some phonemes require more time than do others to process. Therefore for timing

purposes, the speech processor will provide a signal which tells the outside world when another phoneme may be sent. While voicing any particular phoneme the speech circuitry provides a "busy" indication. Bit 7 (the most significant bit in the byte) of relative port 2 gives the busy status when read. A zero on this bit indicates a phoneme is being voiced (or busy). A "not busy" status is represented by a one in that bit position.

8.2.4. "Start Phoneme" Strobe

Relative port 2 is used to voice a phoneme once the phoneme code is latched into the phoneme register. The data which is sent in the output instruction has no meaning. Any phoneme can be continuously voiced simply by sending the "Start Phoneme" strobe at the rate at which you wish the phoneme sound to be updated.

8.2.5. Audio Voiced Output

The volume of the voice can be adjusted by changing R25 if an external speaker is connected to J2. Note that this will likewise increase/decrease the volume for the sound synthesizer output. If the audio output J1 is being used, then the volume is dependent upon the setting of the external amplifier. The relative volume difference between the voice and the sound output can be adjusted through varying the amplitude of the applicable sound channel output.

8.3. Speech Example Program

The following example illustrates the way in which phoneme codes may be voiced :

```
10 DIM P(100)
20 CLS
30 SCREEN 0,1
40 LOCATE 2,30,1 : PRINT "P S S T   S P E E C H   D E M O"
50 SCREEN 0,0
60 LOCATE 4,30,1 : PRINT "HIT ANY KEY TO EXIT"
70 NOTBUSY% = &H80
80 BUSY% = 0
90 FOR PHONOME% = 1 TO 92
100       READ P(PHONOME%)
110 NEXT PHONOME%
120 FOR PHONOME% = 1 TO 92
130       OUT 6,P(PHONOME%)    'send phoneme to phoneme latch
140       OUT 2,0              'strobe the SC01
150       GOSUB 230            'go test for busy
160 NEXT PHONOME%
170 OUT 2,&H3F                 'be quiet for a while
180 FOR AWAIT = 1 TO 200 : NEXT AWAIT
190 IF INKEY$ = "" THEN 190    'keep going until key pressed
```

```
200 END
210 '
220 ' wait till SC01 is not busy
230       STATUS% = INP(2)
240       IF (STATUS% AND NOTBUSY%) = BUSY% THEN 230
250       RETURN
260 '
262 ' The following data are the phonemes that represent:
264 '   "HELLO THIS IS THE P-SST CARD SPEAKING FROM L P SYSTEMS,
266 '    BROUGHT TO YOU BY SOFTWARE WIZARDRY."
268 '
270 DATA &H1B,&H02,&H18,&H26,&H3F,&H39,&H27,&H1F,&H3F,&H27,&H1F
275 DATA &H3F,&H39,&H3C,&H3F,&H25,&H2C,&H3F,&H02,&H1F,&H3F,&H02
280 DATA &H1F,&H3F,&H2A,&H2C,&H3F,&H19,&H08,&H2B,&H1E,&H3F,&H1F
285 DATA &H25,&H2C,&H19,&H27,&H14,&H3F,&H1D,&H2B,&H23,&H0C,&H3F
290 DATA &H02,&H18,&H3F,&H25,&H2C,&H3F,&H1F,&H27,&H1F,&H2A,&H23
295 DATA &H0C,&H1F,&H3F,&H0E,&H2B,&H24,&H2A,&H3F,&H2A,&H28,&H3F
300 DATA &H22,&H17,&H3F,&H0E,&H24,&H2C,&H3F,&H3F,&H1F,&H24,&H1D
305 DATA &H2A,&H2D,&H2D,&H00,&H2B,&H3F,&H2D,&H27,&H12,&H3A,&H2B
310 DATA &H1E,&H2B,&H2C,&H3F
```

The sample program reads the DATA statements and stores the phoneme codes in the array P(PHONOME%). The phonemes are then voiced one at a time in steps 120 through 160. Notice that each time a phoneme is voiced the program will wait for a NOTBUSY indication from the routine at 230 through 250. The status bit is read from relative port 2 by line 230. A BUSY condition is then tested for in Line 240. If the most significant bit of the status byte is a 1, the program will stay in the busy loop. After the most significant bit goes to a 0, the program will return to voice the next phoneme stored in the array. Although there are more exotic methods which can be used to voice these phonemes, the above method demonstrates a straight forward manner in which to understand the programming of the speech synthesizer.

9. CIRCUIT DESCRIPTION

The buss interface on the P-SST is fully compatible with the IEEE-696 standard. For a further explanation of this specification a copy of the Z-100 technical manual (Heath/Zenith part number TM-100) should be consulted.

9.1. The S-100 Interface

The data buss within the P-SST card is a bidirectional buss which is a combination of the S-100 DATA IN and DATA OUT lines (Figure 9-1). The address buss bits A4 thru A7 are examined by the address compare logic U13 and DS1 to determine if the current address on the buss matches the setting of switches S1 through S4 on DS1 (Figure 9-2). The address is further qualified by one of the buss status signals SINP or SOUT (an I/O read or I/O write respectively). This generates a signal called SEL, output of U11 pin 3, which indicates selection of the P-SST (Figure 9-3). As explained in the IEEE-696 spec, the address buss is considered valid at the leading edge of pSTVAL* (status valid) during pSYNC. The combination (logical 'AND') of these two signals is used to clock U10 to activate the function decoder U18. At this time, U18 decodes inputs of A0, A1, A2, and A3. The outputs of U18 are the select signals (0* through 7*) which are used to address each function on the P-SST.

Data buss buffers U21 and U22 are used to pass data into or out of the P-SST. Direction of the data is a function of PDBIN (S-100 pin 78) and address bits A0, A1, A2 (which are ones) with A3 being inactive (zero) at the trailing edge of STB (signifying address and buss status stable). The chip select signal "7*" along with PDBIN* (U5-15) gates DS1 switches 5 through 8 to data buss bits D0 through D3 (Figure 9-4). The PDBIN signal at U14-2 along with CARDSEL, U14-1, produce a low on U21-19 and U22-19 and a high on U21-1 and U22-1 (through inverter U8). This causes data to be passed from the P-SST internal data buss to the S-100 DI buss.

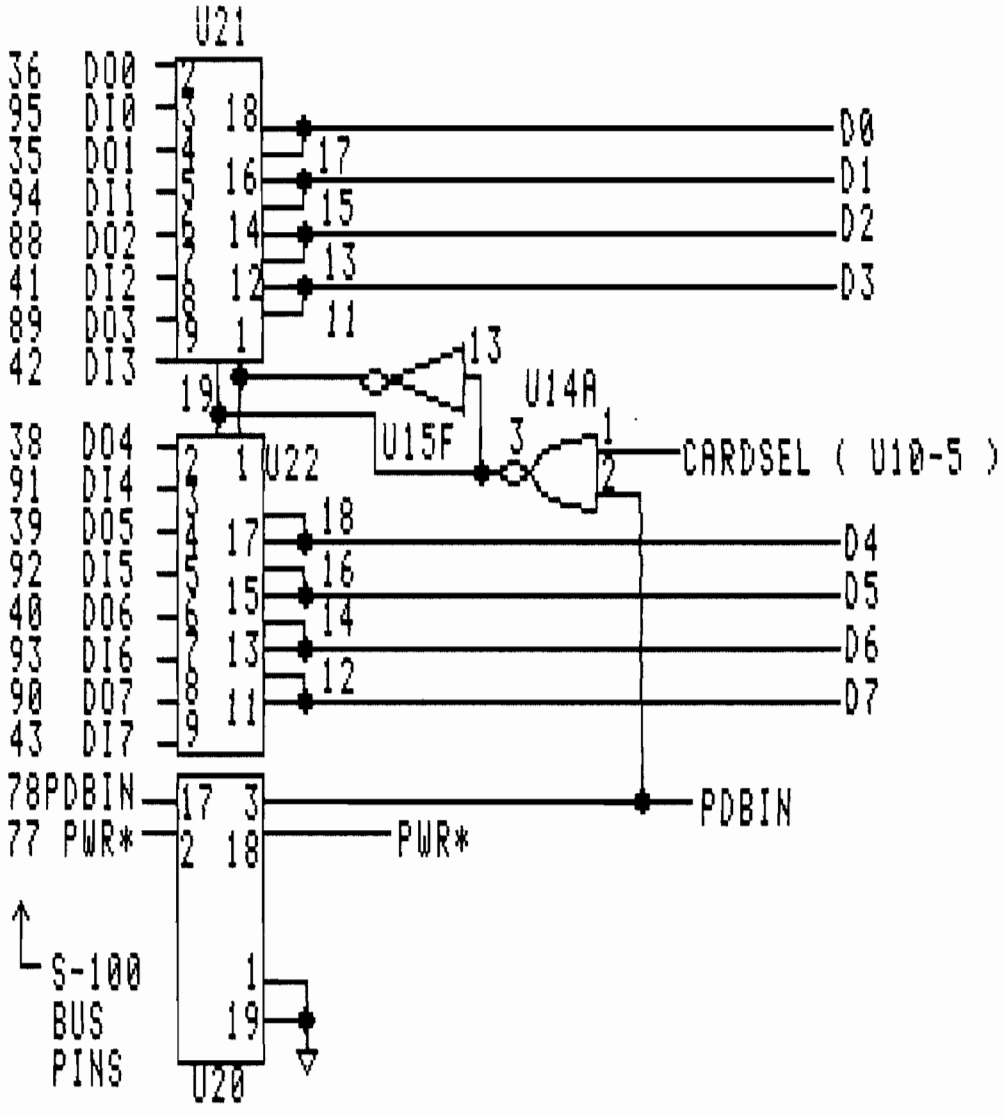


Fig. 9-1: Data Buss Interface

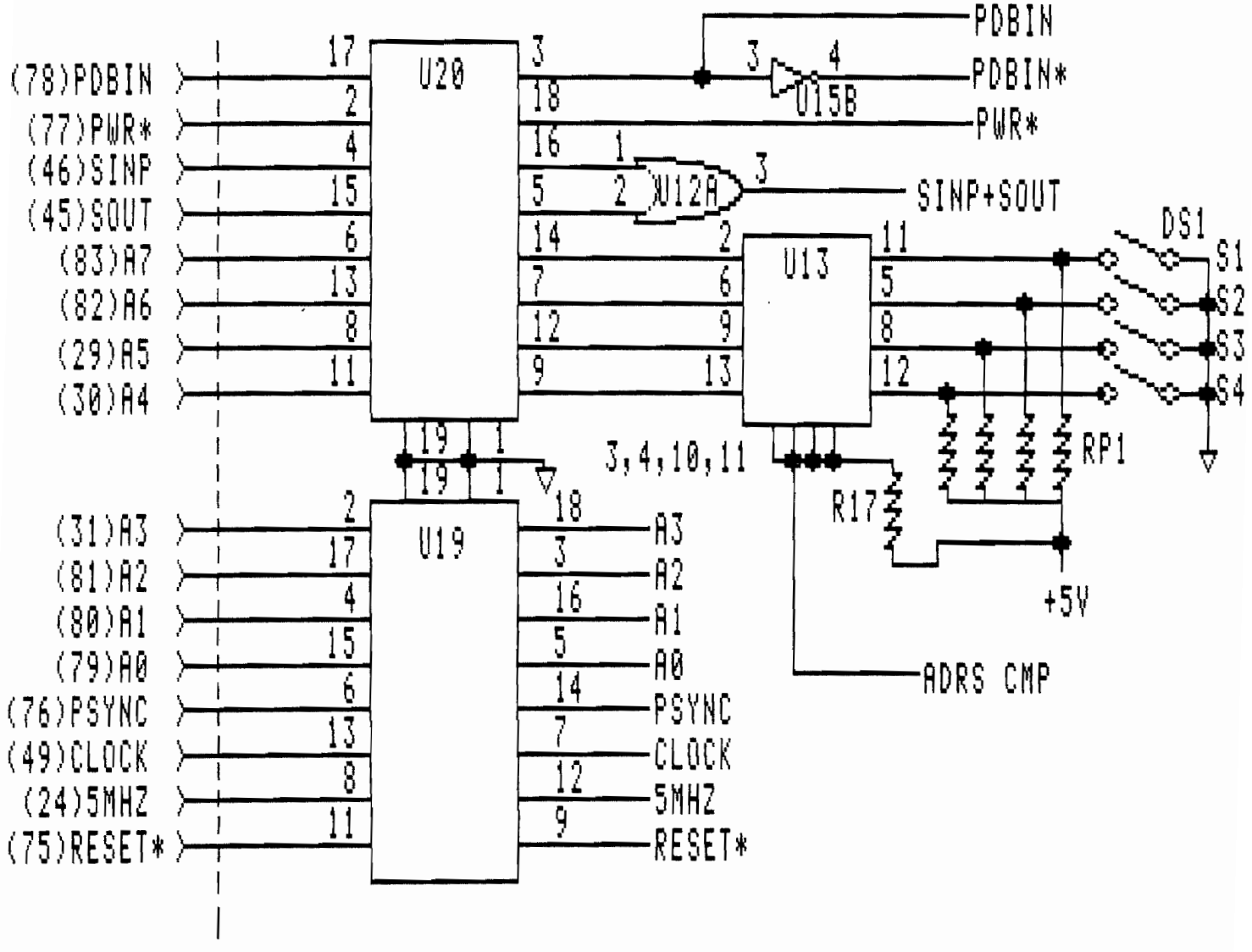


Fig. 9-2: Status/Address Diagram

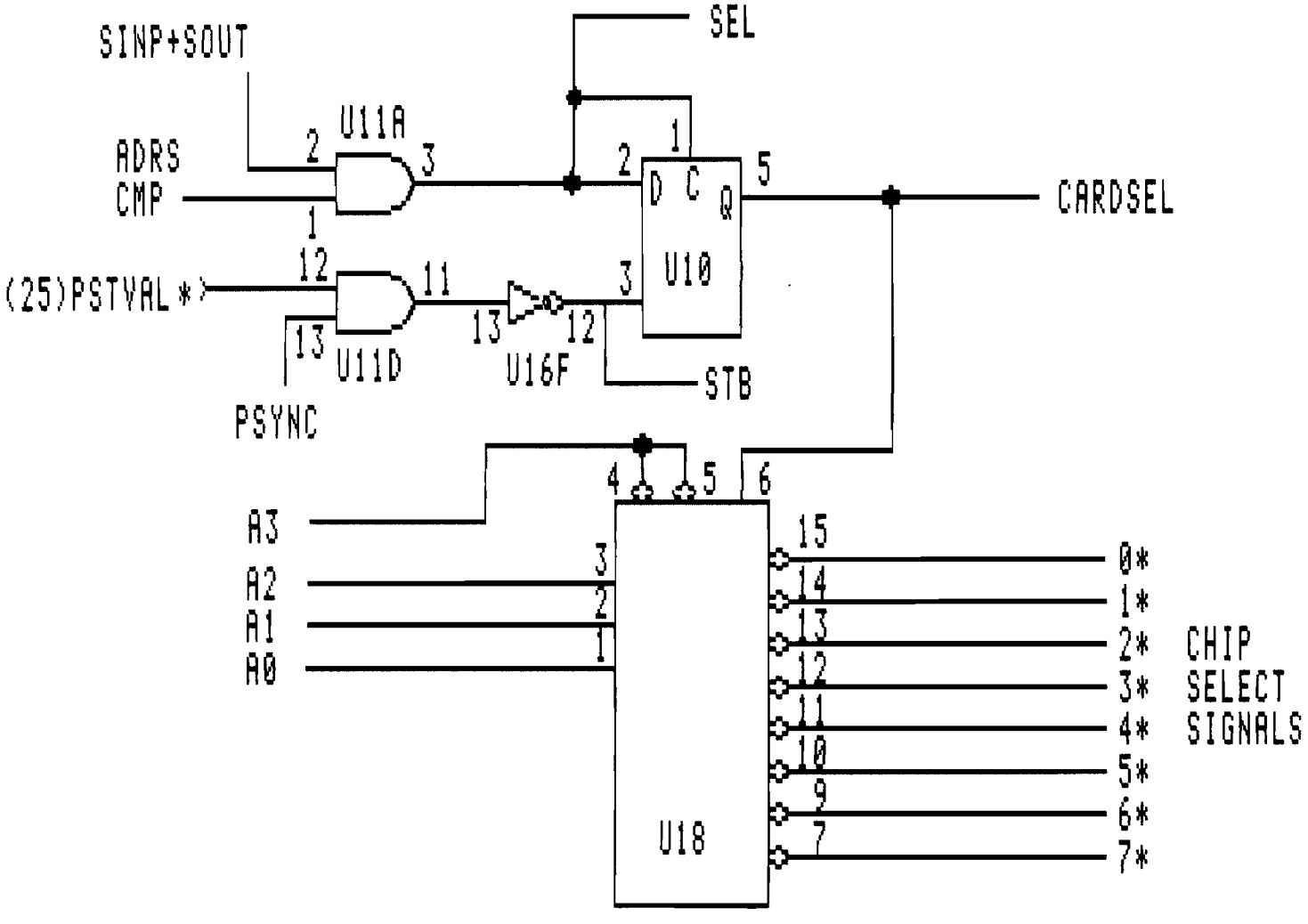


Fig. 9-3: Function Select Diagram

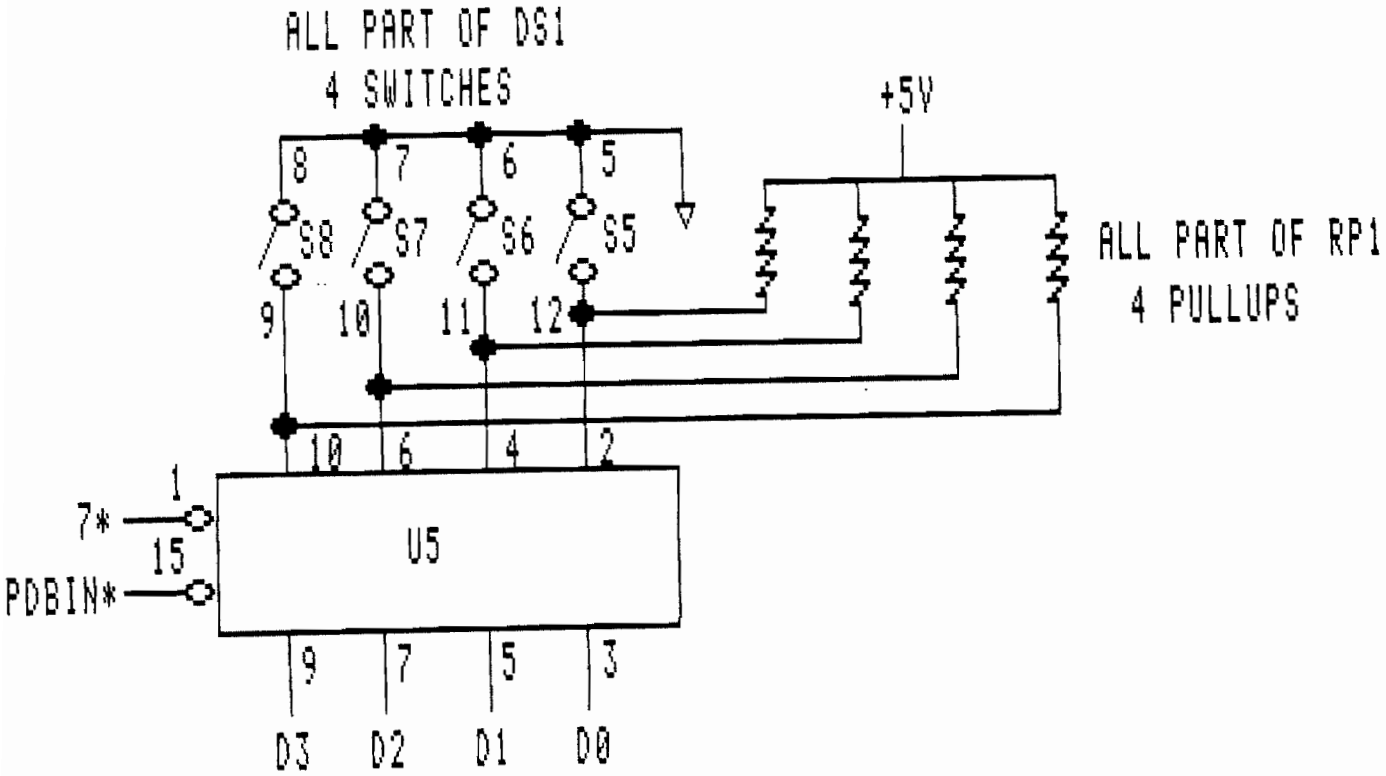


Fig. 9-4: Configuration Switch Port

Due to the relatively slow setup and access times of the VLSI devices, additional time is needed when reading or writing to some of the P-SST functions. This additional time is provided by initiating a wait state via the S-100 buss PRDY* line (Figure 9-5). This wait state provides an additional clock cycle to accomplish the needed access/transfer. Initiation of the wait state takes place when the STB signal becomes active, signaling that the P-SST card is being addressed. The STB signal along with SEL generate an output from U14-11 which is passed through U11-6 to U9-4 and U9-10. This causes U9 flip-flop to be set, thus activating the wait state request line PRDY* (active low) at pin 72 on the S-100 buss. At the falling edge of the 5 Mhz system clock, U9-5 goes low and U9-8 (Q*) remains low (active). When the next falling edge of the 5Mhz clock occurs, the low at U9-12 gets clocked through to cause a high on U9-8 which removes the wait state request and PRDY* goes high. Additional time may be added to the wait state when addressing U3 (MM58167). The output of U3 at pin 4 goes low signaling a BUSY condition and may extend the

wait state for certain operations involving the 58167 clock device (Figure 9-6). This typically results in a one clock cycle extension of the current wait state.

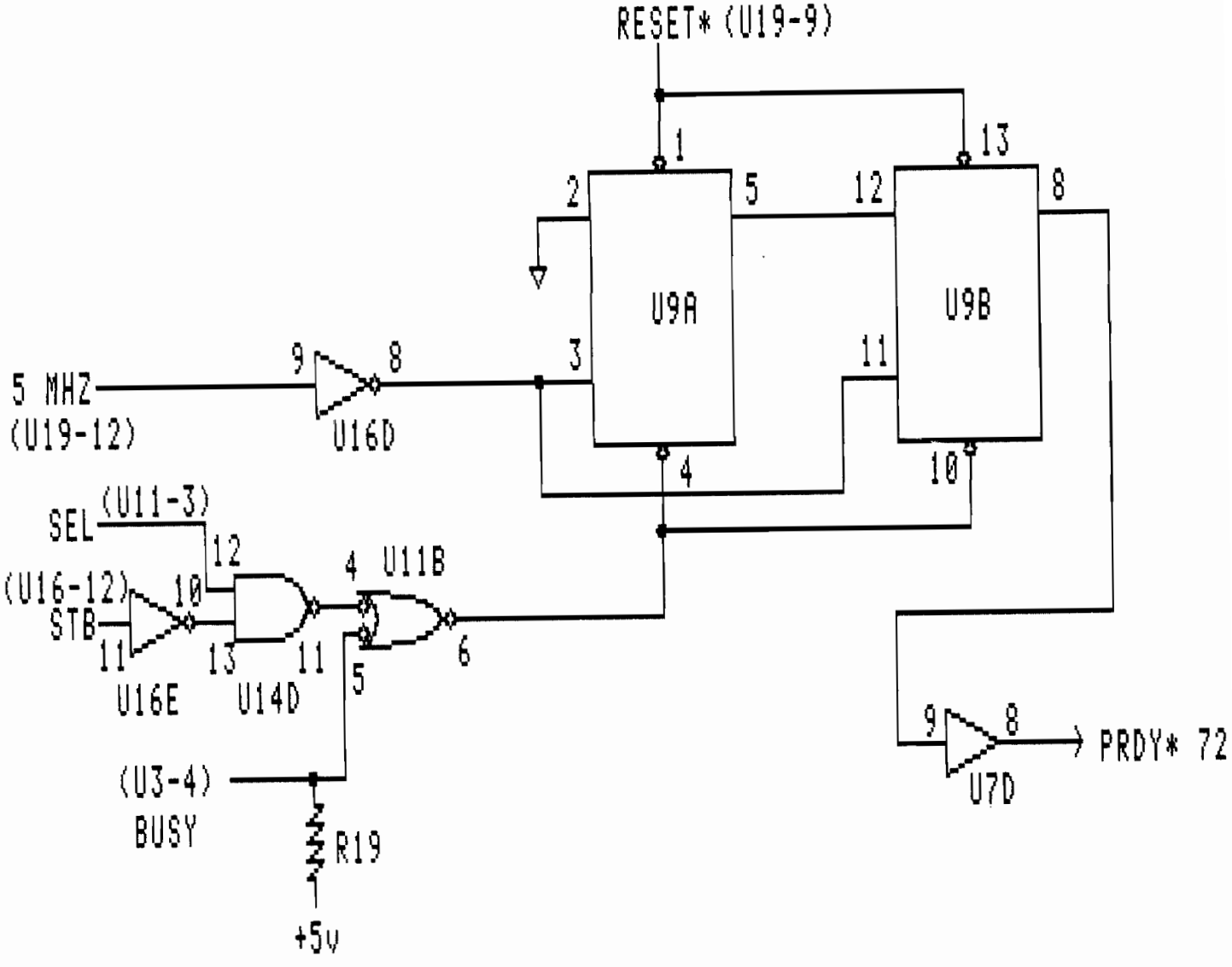


Fig. 9-5: Wait State Generator

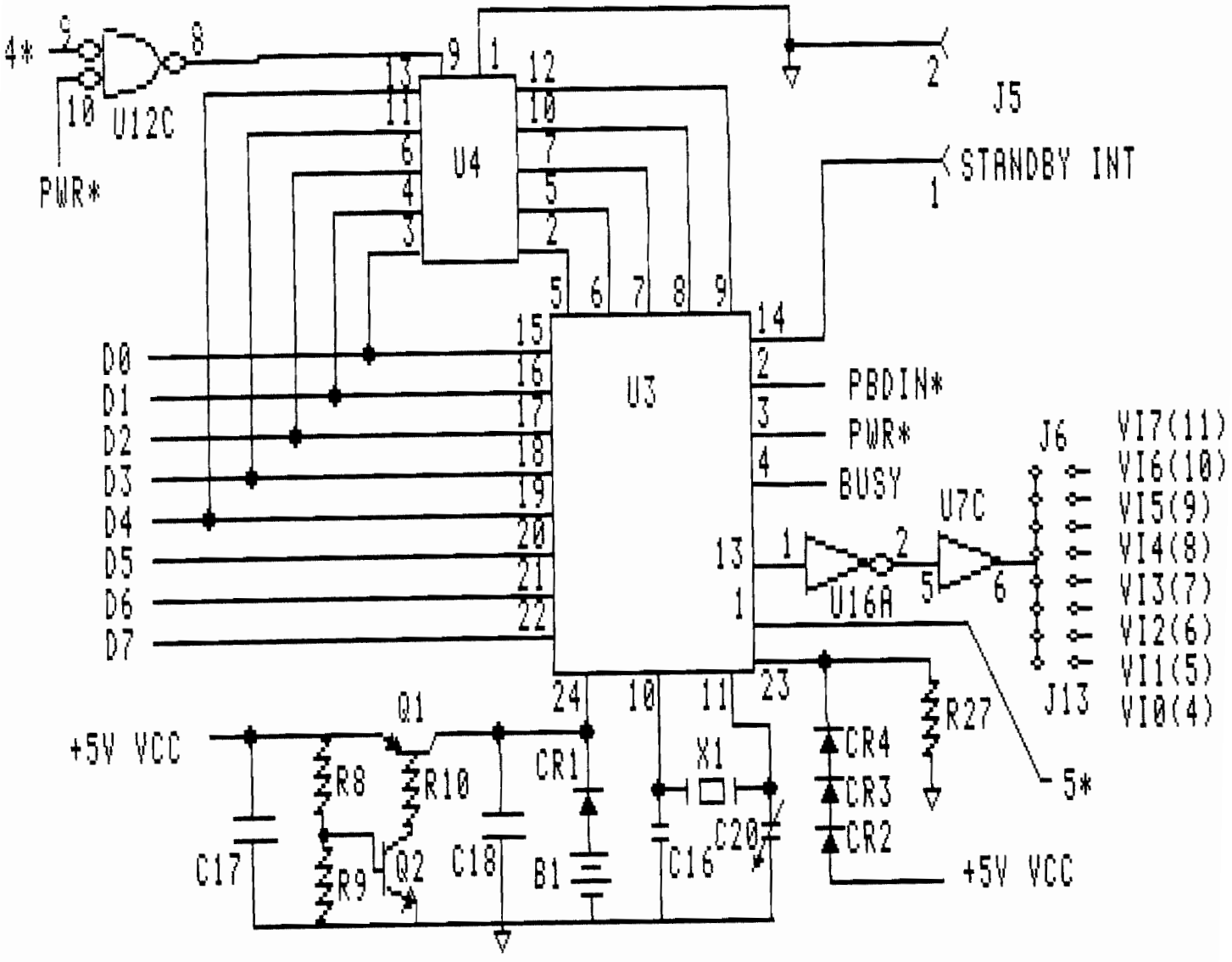


Fig. 9-6: Clock Function

9.2. Clock Circuit

Power Down Mode-Battery Backup

POWER DOWN* pin 23 on U3 is normally held at approximately 3 volts by the drop across diodes CR2 through CR4 (Figure 9-6). In order to insure that no data is lost in U3 the POWER DOWN* signal must be brought low before the supply voltage drops to a level

which would cause false clocking of the signals connected to U3. This is accomplished by holding the voltage at pin 23 at least 2 volts below the supply voltage. Therefore when power is interrupted to the P-SST, pin 23 of U3 reaches a low (placing the 58167 in POWER DOWN MODE) prior to any signals becoming undefined.

Likewise, on power up, the POWER DOWN* input must not be brought to a high logic level until the interface signals to U3 have become valid. Again, this is done by holding the voltage at U3-23 at a logic low by the voltage drop across CR2 through CR4 until the supply has reached a level that will insure the system interface signals are valid. The POWER DOWN* signal will not go high until the supply voltage reaches approximately 4 volts.

Battery Backup Circuit

Shutting off system power will cause the +5 volt supply voltage at the emitter of Q1 to decay. Transistor Q2 is on and will remain conducting until the voltage across Q2 base-emitter (set by the R8/R9 voltage divider) is less than one diode drop. Q2 will begin shutting Q1 off when the supply voltage drops below 4 volts. Normally CR1 is reverse biased by the 5 volts at the collector of Q1. As Q1 turns off, CR1 will begin conducting (Q1 collector < 2.5 volt) resulting in battery current being supplied to Vdd supply voltage pin 24 of U3. Thus, while the system supply is off, the battery maintains power to the clock circuit.

As system power is applied, transistor Q2 will begin conducting when the voltage at the emitter of Q1 reaches approximately 4 volts. This causes Q1 to begin conducting due to the base current flow provided by Q2. The resulting supply voltage at U3-24 causes CR1 to become reverse biased which eliminates the current drain from the backup battery.

Clock Register Select Latch

The 58167 clock (U3) has various registers and modes which may be accessed by providing the proper code on its address buss during buss communications. To reduce the number of Z-100 system I/O addresses which would be taken up by directly tieing the clock address buss to the S-100 address buss, a latch U4 was added to hold these address codes. This approach results in fewer addresses being occupied but requires that the address latch be loaded prior to the clock U3 being accessed (read or write). Data is written to register latch U4 on the falling edge of PWR* in conjunction with address select signal "4*". While the register select information remains on U3 address pins 9, 8, 7, 6, and 5, U3 clock data can be written/read by activating the select signal "5*" and asserting either PDBIN*/PWR*.

Interrupt output U3-13 is connected via inverter U16D and open collector buffer U7C to interface the alarm to one of the S-100 vectored interrupts VI0 through VI7. Selection of one of these interrupts is accomplished by placing the shorting block across the appropriate set of pins marked J6 through J13 on the board.

For a further explanation of these vectored interrupts and how to setup the Z-100 to handle them, consult the Z-100 technical manual.

9.3. Configuration Switches

Switches S5 through S8 on DS1 are designed to indicate to the user the presence or absence of certain functions on the P-SST card (Figure 9-4). Data can be read from these switches by asserting PDBIN* through an I/O read from relative port address 7. This will enable the switch data to be output to the P-SST internal buss via port buffer U5. These four switches when read are the least significant four bits (D0,D1,D2,D3) on the data buss. The most significant four bits are not used and should be ignored. A closed switch (ON) corresponds to a 1 being read in that bit position. Likewise, an open switch (OFF) will result in a 0 being read.

9.4. SC01 Speech Synthesizer

The speech circuitry consists of five major sections. These are: the SC01A speech IC U1, the six bit phoneme latch U2, a coarse inflection level latch U8, status read port and strobe drive (Figure 9-7).

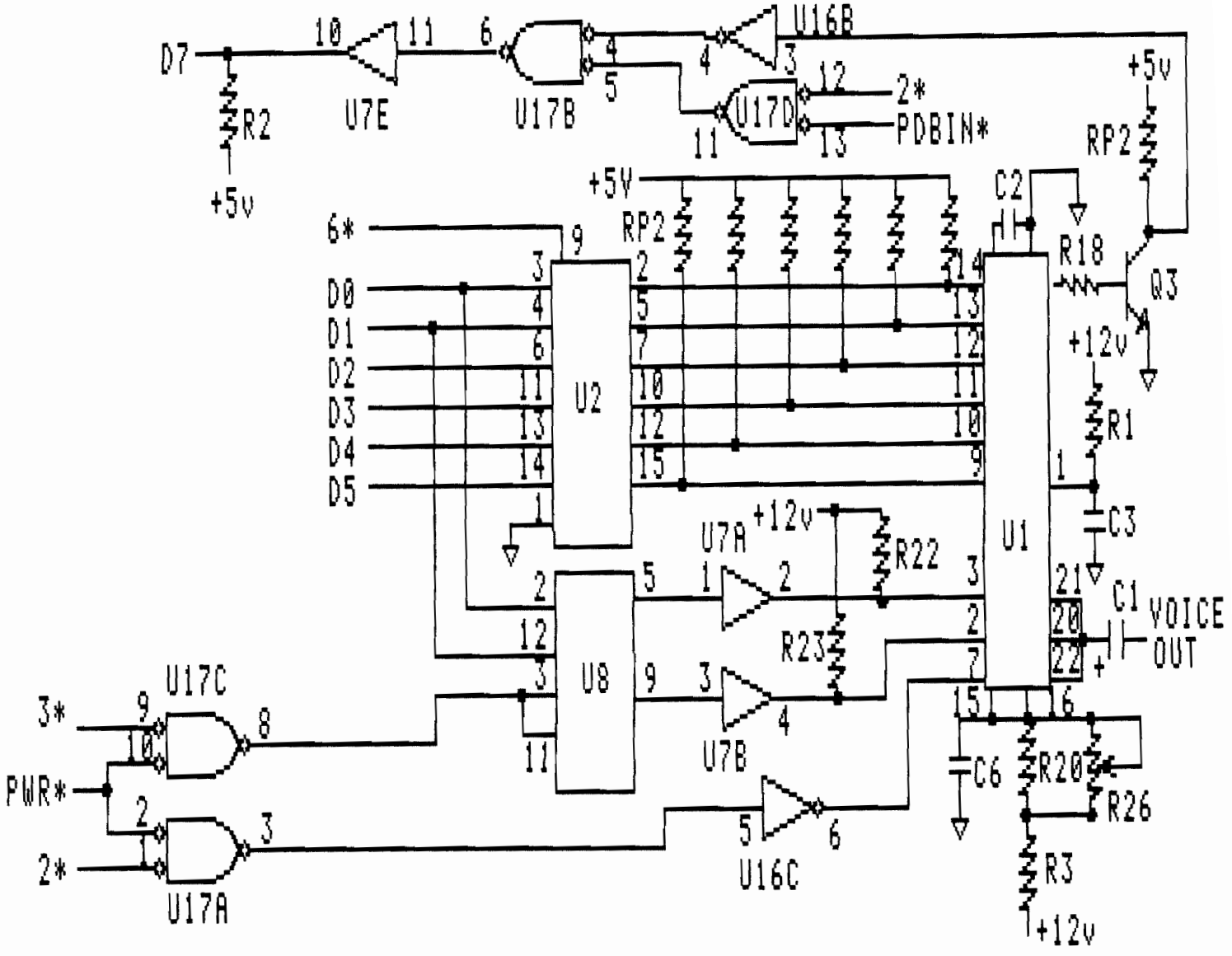


Fig. 9-7: Speech Synthesizer Diagram

The speech IC U1 is powered via a power-up delay circuit of R1 and C3. This delays the recognition of any external signals until C3 reaches approximately 7 volts. Approximately four milliseconds is required for this to occur. The overall pitch of the synthetic voice is set by resistors R3, R20 and pot R26. Manual adjustment of R26 can be done to satisfy individual preference. The voice output from pins 20, 21 and 22 is AC coupled via C1 through isolation resistor R16 to the on-board mixer U24 and audio

amplifier U23. This output is also available at the low level audio output J1 located at the top of the circuit card.

The phoneme latch U2 is a six bit storage device used to hold the phoneme data. Due to the relatively slow input setup time of the SC01's inputs, U2 was used to insure that the phoneme data was stable and met the criteria for Input Setup Time. A phoneme is sent to U2 by writing the data to relative port 6 via an I/O write instruction. At a later time, U1 may be strobed to voice the phoneme which is stored in U2. The pullups in RP2 insure that a logic 1 on any phoneme data line is as close to +5 volts as possible since the SC01 input requires a minimum of +4 volts for recognition of a logic 1.

Coarse inflection levels may be programmed by sending data to relative port 3 (chip select signal "3*". Only two bits (D0 and D1) may be sent to allow a total of four possible inflection levels under program control. The data is stored in flip-flop U8. The outputs of U8, pins 5 and 9, are buffered by open collector gates U7. To be recognized as a logic 1 at the inflection inputs (U1-3,2), the voltage must be a minimum of .8 times Vp (supply voltage). Gates U7 and R22/R23 accomplish this by keeping a logic 1 above the required 9.6 volts.

Once data has been written to the phoneme port, the SC01 may be commanded to voice the phoneme by asserting the strobe signal at U1-7. To command the strobe an I/O write is issued to relative port 2. No data is required to accompany this write instruction. Any data sent will be discarded. The strobe write will cause a short pulse to be sent via U17A-3 and U16C-6 to cause the SC01 to begin processing the currently stored phoneme in U2.

The speech status port is made up of invertor U16B, OR gates U17B and U17D, open collector buffer U7E and MOS to TTL level convertor Q3. When a strobe is received by the SC01 it lowers the BUSY output at U1-8. This turns off Q3 which places a high on pin 3 of U16B. To find out if the SC01 is finished processing a phoneme, initiate an I/O read from relative port 2. This will activate signals "2*" and PDBIN* which in turn will gate the BUSY indication to U17B-6. A logic 0 here indicates that U1 is processing a phoneme. This signal is then gated to the P-SST internal buss by U7E-10. The most significant bit of the data byte read will indicate the status of the SC01. The remaining bits in the byte are undefined. When a logic 1 is read from the status port, the SC01 is not busy and may be commanded to voice the next applicable phoneme.

9.5. Sound Synthesizer and I/O Ports

The Sound Synthesizer U6 is controlled by I/O writes to relative port 0 and I/O writes and reads to relative port 1 (Figure 9-8). To select a particular register for any operation, an I/O write must be issued to relative port 0 (data sent contains the register address code to be used by U6). This activates signals

PWR and 0* and places a high on U6-27 and U6-29. At this time the data present on D0 through D7 is input into U6 as Latch Address information.

Writing data to a register can be done by issuing an I/O write to relative port 1 (after first setting that registers address code into port 0). This activates PWR and 1* which places a logic 0 on U6-29 and a logic 1 on U6-27. This will take the data from the P-SST data buss and latch it into the currently addressed register. To read data from the currently addressed register, an I/O read is done from relative port 1. The active signals in this condition are PDBIN and 1*. This places a logic 1 on U6-29 and a logic 0 on U6-27. The data in the currently addressed register will then be placed on the P-SST buss and transferred to the S-100 buss.

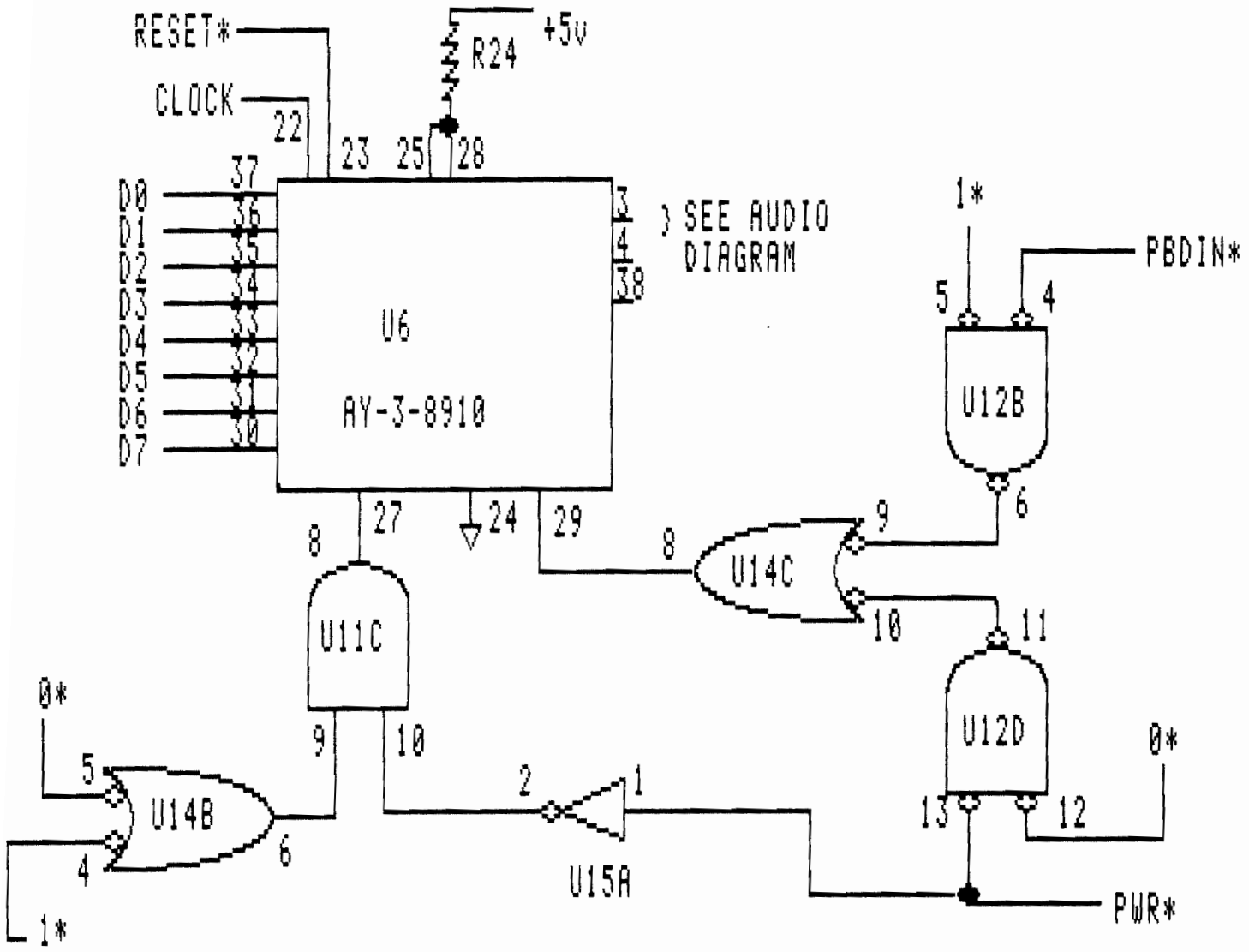


Fig. 9-8: Sound Synthesizer Diagram

9.6. Mixer and Audio Output

Mixer U24 was added to linearly mix the three sound synthesizer channel outputs from U6-3, 4 and 38 (Figure 9-9). The output of U24 is coupled to the J1 low level audio output by DC blocking capacitor C4 and isolation resistor R15. This signal also is input to the U23 on-board amplifier by way of R14 and volume control R25. U23 is an LM386 power amplifier which is set for a

gain of 50. Additional bass boost is provided by C13 and R13. The bass boost provides approximately 5db of additional gain at 100 HZ and rolls off until 2KHZ. The output of U23 is AC coupled to connector J2-2 which can then be used to drive an external speaker with an impedance of 4 to 16 OHM.

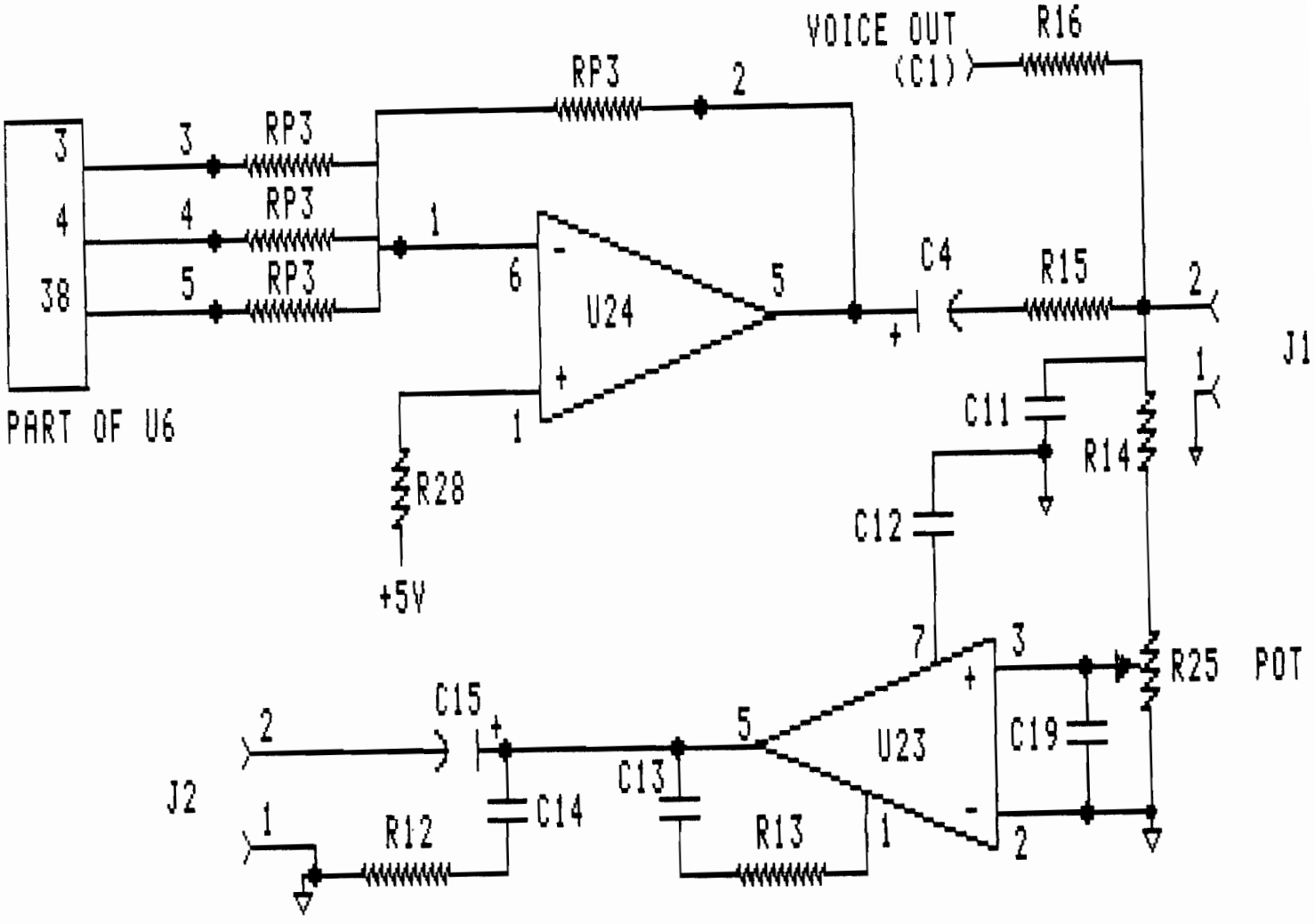


Fig. 9-9: Mixer and Audio Amplifier Diagram

10. IN CASE OF TROUBLE

10.1. Isolating the Problem

The first step in isolating a problem is always to be sure that you actually have a problem. Checking each of the following steps prior to panicking can save you a great deal of frustration, time, and effort. Please do so.

(1) Read the manual. If you've done so before, do so once again. Easily 90% of support questions are covered in the original product documentation, and it's usually far easier to locate the information yourself, than to send your dealer scurrying off for the information.

(2) If you suspect a problem in the software, or have difficulty with the software, make sure you read through section 4 (SOFTWARE DESCRIPTION) before proceeding.

(3) Check to make sure that DS1 is set to a non-conflicting base port address. We suggest setting all the switches to the ON position.

(4) After first powering off your system, check to make sure that the P-SST card is properly seated in one of the S-100 slots. If so, move the card to another slot, in case the problem is simply one of a slight mismatch between the connector halves.

(5) Check any cables that are attached to the P-SST for bad connections, or for location in the wrong position.

(6) Check to see that the battery is properly installed with the positive side (+) showing.

(7) Consult the dealer from whom you purchased the P-SST card.

(8) After performing all of the above, if the problem persists, refer to the section on the Limited 90-Day Warranty.

10.2. Warranty Information

LP SYSTEMS warrants to the original purchaser that the P-SST CARD shall be free from defects in the materials or workmanship for a period of ninety (90) days from the date of purchase. This limited warranty does not apply to the P-SST Distribution Software programs, nor the P-SST documentation manual.

Should you find during the warranty period that the P-SST CARD is defective, LP SYSTEMS will repair or replace the hardware at its option. To receive this in-warranty service, a defective P-SST CARD must be shipped to LP SYSTEMS (at the address below) at the purchaser's expense no later than one week after the end of the

warranty period and must be accompanied by proof of date of purchase satisfactory to LP SYSTEMS. This warranty shall not apply if the P-SST CARD hardware has been damaged by negligence, accident, unreasonable use or by other causes unrelated to defective materials and workmanship. This warranty is expressed in lieu of all other warranties of merchantability and fitness for a particular purpose, and of all other obligations or liabilities on the seller's part.

Defective P-SST cards should be returned to the following address ONLY after notifying Software Wizardry, Inc. of the problem and arranging for a Return Authorization Number. Boards returned without such a number will be returned to the sender at his expense without service being performed:

LP Systems Repair
c/o Software Wizardry, Inc.
1106 First Capitol Drive
St. Charles, Mo. 63301

For more information, contact Software Wizardry, Inc. at (314) 946-1968.

No Warranty of Performance

LP SYSTEMS does not and cannot warrant the performance or results that may be obtained by using the P-SST CARD or its software.

10.3. Out of Warranty Service

LP Systems, Inc. will repair "out of warranty" P-SST cards for a cost (to the original owner) of not more than one hundred (100) dollars. The card must be shipped to LP SYSTEMS at the expense of the original owner and must be accompanied by proof of purchase to the satisfaction of LP SYSTEMS. This service is subject to the availability of parts. The owner will be notified within thirty (30) days after the receipt of the P-SST Card how soon it will be returned.

11. APPENDIX A - SPECIFICATIONS

11.1. J1-J5 Characteristics

I/O	DESCRIPTION	CHARACTERISTICS
J1	To external power amp	Vout=1.5 volt max R_load > 20 Kohm
J2	To external speaker	Pout = 1 watt max R_load = 8 ohm
J3	I/O ports/joysticks	TTL compatible with internal pullups. Will source and sink 1 TTL load. Logic low (sink): V_ol<.5v,I_ol=1.6mA Logic low (source): V_oh>2.4v,I_oh=40uA
J4	(same as J3)	
J5	Standby Interrupt	CMOS open drain output will sink one TTL load. Recommend external 10K pull-up to an external +5 volt supply to drive one external TTL load. Logic low (sink): Vdd=4.5v,I_ol=1.6mA Vmax out=0.4v Logic high (leakage): Vout<Vdd,Imax=10uA

11.2. Power Requirements

+5 volt supply	Pmax = 2 watt
+12 volt supply	Pmax = 1.5 watt

12. APPENDIX B - ADDRESSING INFORMATION

12.1. Clock Function

1. Write the address code for the clock register selection to port 4(corresponds to A0 thru A4 address...5 bits).

2. Read/Write to port 5 to perform I/O on the previously selected register.

o INTERRUPT OUTPUT drives selectable VI0 thru VI7 vectored interrupts on the Z-100.

o STANDBY INTERRUPT output brought out for user at J5.

12.2. Sound Synthesizer/IO Ports

1. Port 0 is the register select port.
2. Port 1 is the data port.

12.3. Speech Synthesizer

1. Port 6 is the phoneme data port (write only)..6 bits
2. Port 2 is the "voice phoneme" port
3. Port 3 is the voice inflection port
4. Port 2 is the voice status port (bit D7..msb - read only)
   D7 = 0; busy voicing phoneme
   D7 = 1; not busy

12.4. Configuration Dip Switch S1 (Switches 5 thru 8)

Read bits D0 thru D3 on port 7.
S5 corresponds to bit D0, etc.

12.5. Address Switch Settings

Base Address	U19 Switch Settings S1	S2	S3	S4
0	0	0	0	0
16	0	0	0	1
32	0	0	1	0
48	0	0	1	1
64	0	1	0	0
80	0	1	0	1
96	0	1	1	0
112	0	1	1	1
128	1	0	0	0
144	1	0	0	1
160	1	0	1	0
176	1	0	1	1
192	1	1	0	0
208	1	1	0	1
224	1	1	1	0
240	1	1	1	1

** NOTE **

When the switch is in the RIGHT (ON) position, it generates a ZERO bit. When the switch is in the LEFT (OFF) position, it generates a ONE.

13. APPENDIX C - PARTS LIST

The below listed parts may be obtained from your computer parts dealer, or from Software Wizardry, Inc., 1106 First Capitol Drive, St. Charles, Mo. 63301, (314) 946-1968. Call or write for current prices.

Q2	2N2222
Q3	2N2222
Q1	2N2907
R23	3.3K 1/4 W 5%
R20	3.3K 1/4 W 5%
R22	3.3K 1/4 W 5%
R24	3.3K 1/4 W 5%
RP1	3.3K SIP PACK 9 R
R8	3.9K 1/4 W 5%
X1	32.768KHZ CRYSTAL
R1	33 OHM 1/4 W 5%
R27	330 OHM 1/4 W 5%
B1	3V LITHIUM BATTERY BR-2320
C17	4.7UF AL.EL.AX 25WVDC
RP2	4.7K SIP PACK 7 R
C20	5-40 PF TRIMMER CAP
U7	7417
U14	74LS00
U16	74LS04
U15	74LS04
U11	74LS08
U18	74LS138
U21	74LS244
U22	74LS244
U20	74LS244
U19	74LS244
U13	74LS266
U17	74LS32
U12	74LS32
U5	74LS366
U2	74LS378
U4	74LS378
U8	74LS74
U9	74LS74
U10	74LS74
U6	AY-3-8910
HS2	HEAT SINK PS2 HS115
HS1	HEAT SINK PS2 HS115
J1-J5	HEADER STRIP SINGLE ROW 36 POST ANGLE
J6-J13	HEADER STRIP SINGLE ROW 36 POST STRAIGHT
U23	LM386
C2	.01UF CERAMIC
C19	.01UF CERAMIC
C21	.01UF CERAMIC
C22	.01UF CERAMIC
C23	.01UF CERAMIC
C24	.01UF CERAMIC

C25	.01UF CERAMIC
C26	.01UF CERAMIC
C27	.01UF CERAMIC
C28	.01UF CERAMIC
C29	.01UF CERAMIC
C30	.01UF CERAMIC
C34	.01UF CERAMIC
C13	.033UF POLY
C14	.047UF POLY
C9	.1UF CERAMIC
C10	.1UF CERAMIC
C11	.1UF CERAMIC
C18	.1UF CERAMIC
C31	.1UF CERAMIC
C32	.1UF CERAMIC
C33	.1UF CERAMIC
C7	.22UF POLY
C8	.22UF POLY
R12	10 OHM 1/4 W 5%
C1	10UF AL.EL.AX 16WVDC
C4	10UF AL.EL.AX 16WVDC
C12	10UF AL.EL.AX 16WVDC
R18	3.3K 1/4 W 5%
R13	10K 1/4 W 5%
R14	10K 1/4 W 5%
R19	10K 1/4 W 5%
R3	10K 1/4 W 5%
R3	150 PF CERAMIC
DS1	8 SWITCH DIP PACK
R2	1K 1/4 W 5%
R9	1K 1/4 W 5%
R10	1K 1/4 W 5%
R15	1K 1/4 W 5%
R16	1K 1/4 W 5%
R17	1K 1/4 W 5%
R28	1K 1/4 W 5%
RP3	1K SIP PACK 5 R
CR1	1N4148
CR2	1N4148
CR3	1N4148
CR4	1N4148
C16	22 PF CERAMIC
C3	220 UF AL.EL.AX 16 WVDC
C15	220 UF AL.EL.AX 16 WVDC
PS1	LM7805CT
PS2	LM7812CT
U3	MM58167
U24	LM3900
R25	10K POT
R26	10K POT
U1	SC01A
S1	22 PIN SOCKET
S3	24 PIN SOCKET
S6	40 PIN SOCKET

14. APPENDIX D - FREQGEN PROGRAM

```
10 '-------------------------------------------------------------
20 '
30 '         F R E Q G E N   ----   generate the data table for
40 '                                8 octaves used to program the
50 '                                PSG register pair
60 '
70 '-------------------------------------------------------------
80 '
90 DIM NOTE$(12),FREQ(12)
100 '
110 '
120 NOTE$(1) = "C "
130 NOTE$(2) = "C#"
140 NOTE$(3) = "D "
150 NOTE$(4) = "D#"
160 NOTE$(5) = "E "
170 NOTE$(6) = "F "
180 NOTE$(7) = "F#"
190 NOTE$(8) = "G "
200 NOTE$(9) = "G#"
210 NOTE$(10) = "A "
220 NOTE$(11) = "A#"
230 NOTE$(12) = "B "
240 '
250 FREQ(1) = 32.703
260 FREQ(2) = 34.648
270 FREQ(3) = 36.708
280 FREQ(4) = 38.891
290 FREQ(5) = 41.203
300 FREQ(6) = 43.654
310 FREQ(7) = 46.249
320 FREQ(8) = 48.999
330 FREQ(9) = 51.913
340 FREQ(10) = 55!
350 FREQ(11) = 58.27
360 FREQ(12) = 61.735
370 CLS
380 PRINT "                  IDEAL          ACTUAL      4 BIT     ";
385 PRINT "8 BIT  "
390 PRINT "NOTE  OCTAVE   FREQUENCY       FREQUENCY   MSBYTE    ";
395 PRINT "LSBYTE "
400 FOR OCTAVES = 0 TO 7
410    FOR INCREMENT = 1 TO 12
420       FREQUENCY = FREQ(INCREMENT) * (2^OCTAVES)
430 '     FREQ(INCREMENT) = FREQUENCY        'next octave base
440       PSGFREQ = ((2 * (10^6))/(FREQUENCY * 16))
450 '
460       GOSUB 670  'hex equivalent of psgfreq
470 '
480       GOSUB 810  'determine actual integer value now
490 '
500       PRINT SPC(2);
510       PRINT NOTE$(INCREMENT);
```

```
520       PRINT SPC(4);
530       PRINT OCTAVES+1;
540       PRINT SPC(5);
550       PRINT USING "####.##";FREQUENCY;
560       PRINT SPC(7);
570       PRINT USING "####.##";ACTFREQUENCY;
580       PRINT SPC(8);
590       PRINT AKAR2$;
600       PRINT SPC(7);
610       PRINT AKAR1$
620    NEXT INCREMENT
630 NEXT OCTAVES
640 '
650 END
660 '
670 '-------------------------------------------------------------
680 '          hex equivalent of PSGFREQ variable
690 '          stored in AKAR2$,AKAR1$
700 '
710 AKAR2 = INT(PSGFREQ / (256))
720 AKAR1 = INT(PSGFREQ - AKAR2*(256))
730 '
740 AKAR =  AKAR2*(256) + AKAR1
750 '
760 AKAR2$ = HEX$(AKAR2)
770 AKAR1$ = HEX$(AKAR1)
780 '
790 '
800 RETURN
810 '-------------------------------------------------------------
820 ' find actual frequency programmed  -- rounding error account
830 '
840 '
850 ACTFREQUENCY = ((2 * (10^6)) / (AKAR * 16 ))
860 '
870 RETURN
880 '-------------------------------------------------------------
890 '
```

15. APPENDIX E - CLOCK ADJUSTMENT PROCEDURE

The recommended procedure for calibrating the P-SST real time clock is listed below. If you do not have the necessary equipment available to you, or if the following seems like a bit much, you may also adjust the clock by trial and error - although this could take a much longer time to complete.

Should you use the trial and error method, we would suggest you note the direction of rotation listed below for both faster and slower clocks, and that you record the position of the trimmer and the results after each alteration. It is a great deal easier to know where you're going, if you know where you came from.

In order to adjust the clock on the P-SST card the following equipment is required:

1. Oscilloscope (does not require high-bandwidth scope)
2. Oscilloscope Probe greater than 10 Megohm, <10 pF
3. S-100 extender card (e.g. Heath/Zenith ZA100-4 extender)
4. Non-metallic screw driver

Before proceeding, insure that your Z-100 is off.

** NOTE **

Extreme care should be taken during this adjustment. Do not attempt it if you have no prior experience working with electronic instrumentation. LP Systems, Inc. and Software Wizardry, Inc. will not be responsible for damages to calibration components or equipment due to insufficient precautions or carelessness on the part of the operator.

1. Remove the top cover of the Z-100.

2. Install the S-100 extender card in one of the available buss slots and place the P-SST card in the extender. Make sure the card (component side) is facing forward toward the front panel.

3. Turn the Z-100 on. No software is required to be run for this procedure. Set the scope for AC coupling, .5 volt per division.

4. Locate the clock device U3 and clock speed trimmer C20.

5. Trimmer C20 is to be adjusted such that the amplitude of the 32.768 KHZ waveform at pin 10 on U3 is at its maximum level. This varies slightly for each clock. Nominally the peak to peak amplitude will be between 1.5 and 3.5 volts.

6. Trimmer C20 may be adjusted either clockwise or counter-

clockwise to attain the reference clocks maximum amplitude. The trimmer action is continuous.

7. This maximum amplitude will be used as a reference. Note the position of rotation for C20 after adjusting.

8. Power off the Z-100. Remove the P-SST and extender.

9. Replace the P-SST in one of the Z-100 buss slots.

The clock should now be running at its fastest speed. You should precisely set the clock now using the CLOCK.COM utility provided as part of the P-SST distribution software. Over a period of time (1 to 2 days) the clock can be checked to verify its speed. C20 can be adjusted, clockwise, a little at a time (1/10 rotation) to slow the clock down to a precise reference. After adjusting, if the clock is running slow, C20 can then be adjusted counter-clockwise slightly to speed it up once more.

16. APPENDIX F - J3/J4 INTERFACE TO REAR PANEL

The cable required to connect the J3 and J4 connectors to the rear panel of the Z-100 is available as part number LPS-PSST-CBL-3. If you wish to build your own cable or to modify the Heath/Zenith Z204-2 cable, the pinout is as follows:

	J3/J4 15 pin header		Z-100 Back Panel DB25S	
	1	<---------->	1	SPARE
	2	<---------->	15	BIT D0
	3	<---------->	2	BIT D1
	4	<---------->	17	BIT D2
	5	<---------->	3	BIT D3
*	6	<---------->	10	BIT D4
	7	<---------->	4	BIT D5
*	8	<---------->	11	BIT D6
	9	<---------->	5	BIT D7
	11	<---------->	6	SPARE
*	12	<---------->	12	SIGNAL GROUND
	13	<---------->	7	SPARE
	14	<---------->	20	SPARE
	15	<---------->	8	SPARE

NOTE : The asterisk (*) by the connection indicates a wire which must be added to the standard Heath/Zenith Z204-2 cable.