As you are now the owner of this document which should have come to you for free, please consider making a donation of £1 or more for the upkeep of the (Radar) website which holds this document. I give my time for free, but it costs me money to bring this document to you. You can donate here https://blunham.com/Misc/Texas

Many thanks.

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Misc/Texas). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you.

Colin Hinson
In the village of Blunham, Bedfordshire.

TEXAS INSTRUMENTS

TM 990

TM 990/306 Speech Module

MICROPROCESSOR SERIES™

User's Guide

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 GENERAL

The TM 990/306 is a speech generation module that can be used in conjunction with TM 990 CPU modules to produce speech from a self-contained data set. It can be used to construct sentences having a somewhat flat inflection. This module has been specially designed to operate in noisy environments such as found in assembly lines, stockrooms or automobiles. Other features of this module include:

- Self-contained vocabulary of 179 words (see Appendix C)
- External interface allows control by non-TM 990 computer systems
- Designed to fit either the TM 990/510 or TM 990/520 card cage
- Contains an interval timer
- On-board amplifier provides up to 1 watt of power to a 8 ohm speaker
- Preamplifier outputs can be used to drive custom amplifier system
- Interrupt or polled-status operation
- Speech data in TMS 2716 EPROM.

The TM 990/306 module with its principal components identified is shown in Figure 1-1 and the block diagram for the module is given in Figure 1-2.

1.2 MANUAL ORGANIZATION

This manual is organized as follows:

- Section 1 covers module characteristics and specifications
- Section 2 shows how to install, power-up, and operate the TM 990/306
- Section 3 covers the programming aspects of the module
- Appendices which include schematics and demonstration software.

1.3 GENERAL SPECIFICATIONS

- Power Requirements:

		Current	
Voltage	Regulation	Typ	Max
+12 V	±3%	195 mA	1445 mA
-12 V	±3%	185 mA	240 mA
+5 V	±3%	415 mA	610 mA

- Operating Temperature: 0 to 70° degrees C

- Module Dimensions: Width - 11 inches (27.9 cm)
  Height - 7.5 inches (19.1 cm)
  Board Thickness - 0.062 inch (1.6 mm)

1.4 APPLICABLE DOCUMENTS

- TTL Data Book

- The MOS Memory Data Book

- The Bipolar Microcomputer Components Data Book

- TM 990/100M Microcomputer User's Guide

- TM 990/101M Microcomputer User's Guide

- TMS 1000 Series Data Manual

- TMS 1000 Series MOS/LSI One-Chip Microcomputer Software User's Guide

- TMS 1000 Series MOS/LSI One-Chip Microcomputer Programmer's Reference Manual.

- TMS 9901 Programmable Systems Interface Data Manual

- Brantingham, Larry, and Richard Wiggins, "Three-Chip System Synthesizer, Human Speech," Electronics, August 31, 1978, pp. 109-116

- Brantingham, Larry, "Speech Synthesizers with Linear Predictive Coding," Interface Age, June 1979, pp. 72-75

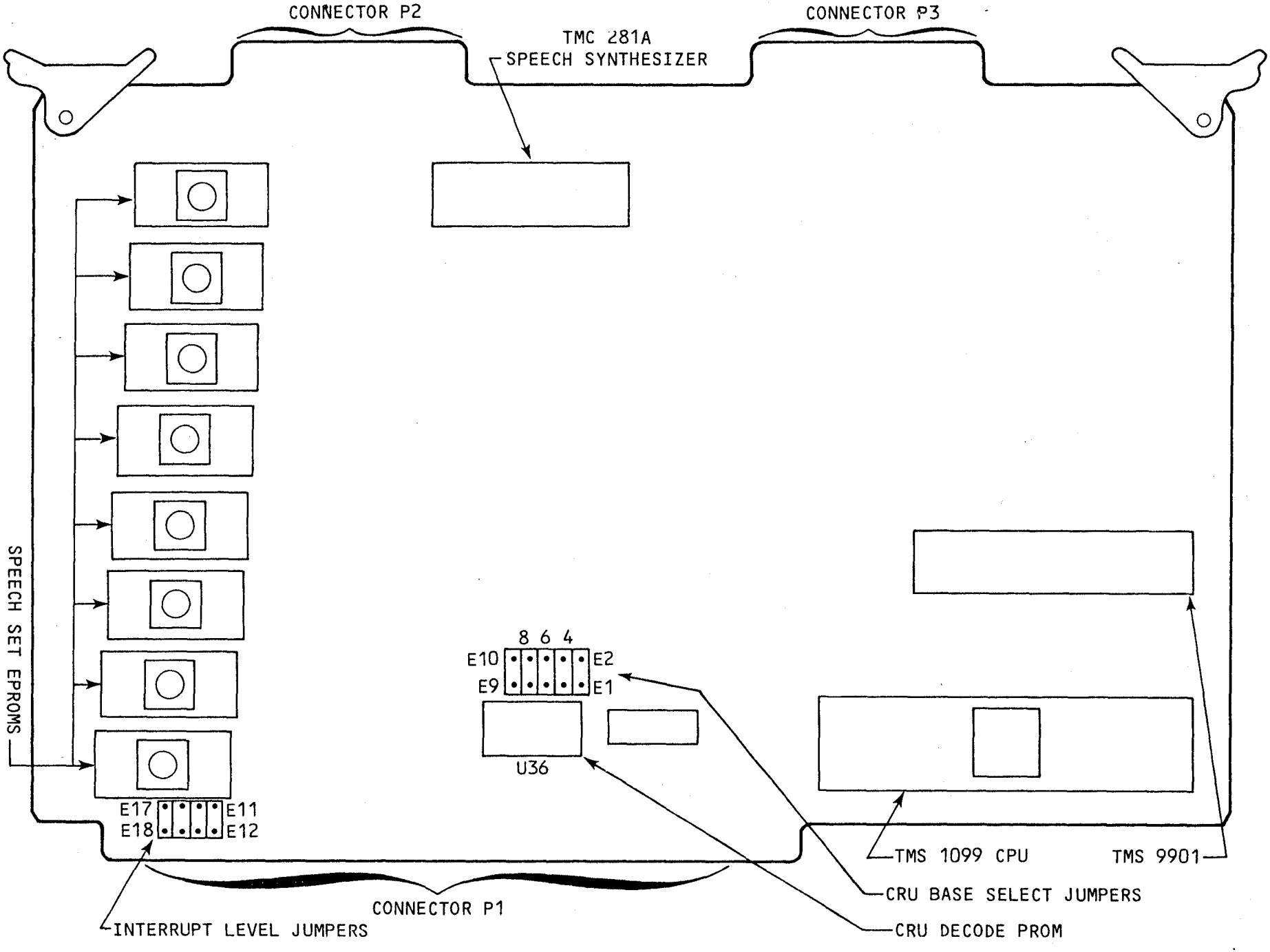


FIGURE 1-1. TM 990/306 PRINCIPAL COMPONENTS

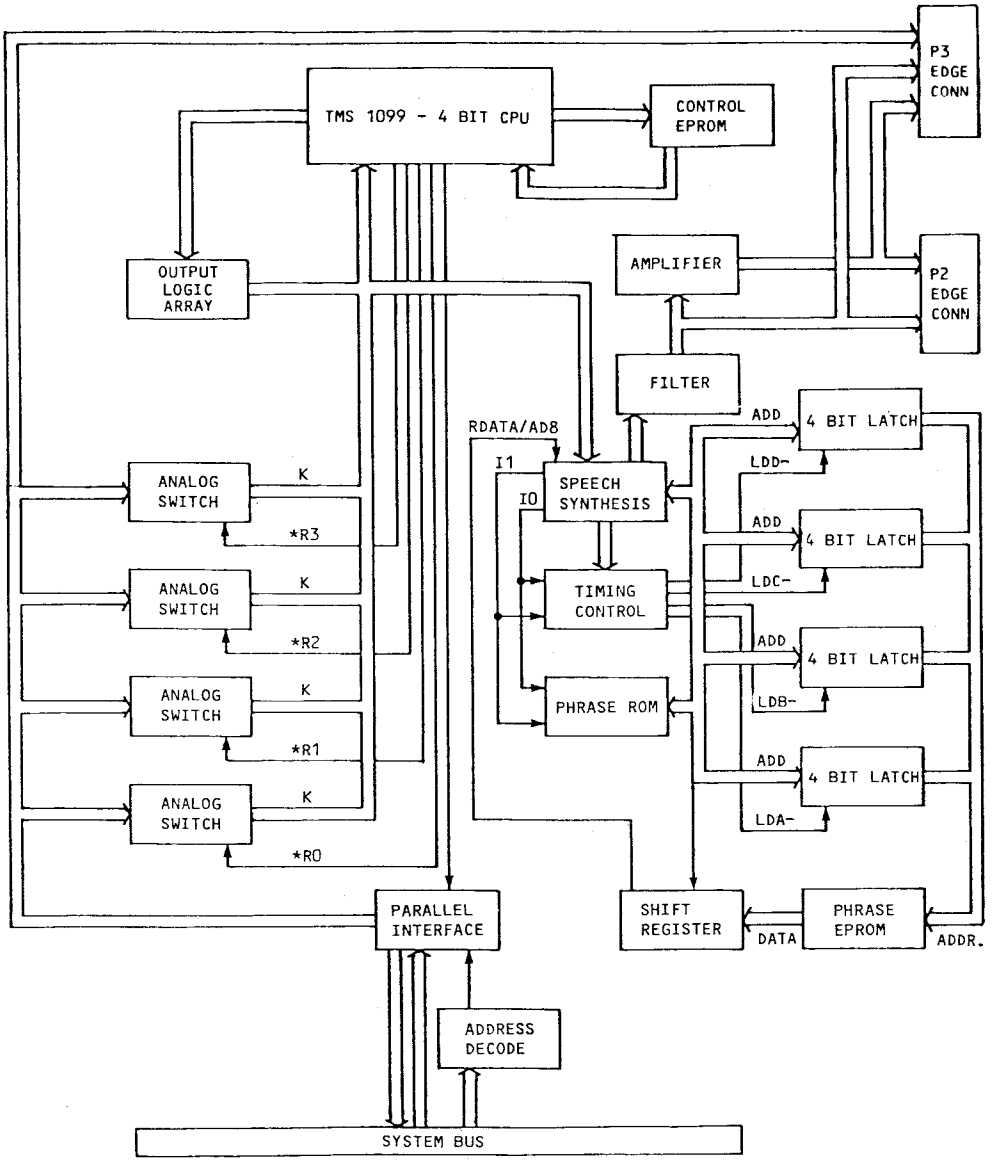


FIGURE 1-2. TM 990/306 BLOCK DIAGRAM

SECTION 2

INSTALLATION AND OPERATION

2.1 GENERAL

The procedures for unpacking and setting up the TM 990/306 module for operation are given in this section along with a test routine that can be used to check out the module.

2.2 UNPACKING AND INSPECTION

Remove the TM 990/306 from its carton and discard any protective wrapping. Inspect the module for any damage that could have occurred during shipping. Report any damage to your TI supplier.

2.3 INTERRUPT LEVEL SELECTION

If the interrupt driven mode of operation (software selected) is used, the processor is free to carry out other functions while the speech module is still talking. A jumper-selectable interrupt is issued by the TM 990/306 when the speech synthesis processor stops talking. When the host CPU receives the interrupt, it decides whether to allow speaking to continue, impose a delay, or stop speaking entirely.

Interrupt levels 9, 10, 11, and 12 are jumper-selectable on the TM 990/306. Table 2-1 shows the required jumper positions to select these interrupts. Figure 2-1 shows interrupt level 12 being selected.

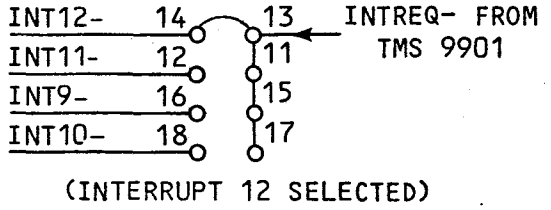


FIGURE 2-1. INTERRUPT LEVEL SELECTION

TABLE 2-1. INTERRUPT LEVEL SELECTION

Interrupt Level Selected	Jumper Position
INT9-	E15 to E16
INT10-	E17 to E18
INT11-	E11 to E12
*INT12-	E13 to E14

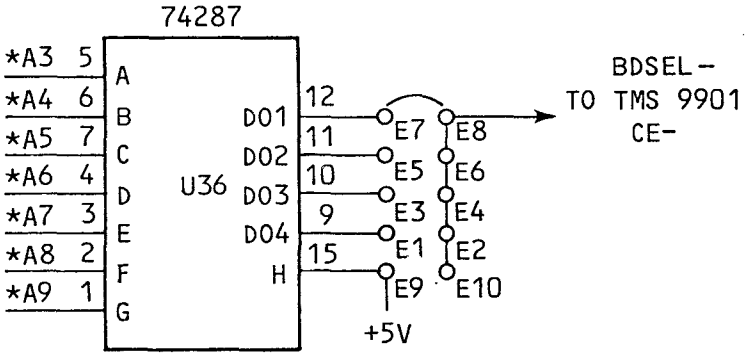
*INT12- is jumpered at factory

2.4 CRU SOFTWARE BASE ADDRESS SELECTION

The TM 990/306 module is selected by a unique CRU software base address (entire 16-bit contents of register 12) that is provided by a 74287 decode PROM at U36 (see Figure 2-2). This PROM decodes address lines *A3 through *A9 and provides outputs D01 through D04. These outputs provide the BDSEL signal for the TMS 9901 I/O controller. The TM 990/306 is shipped with a jumper connected from E7 to E8. This jumper position provides an active CE- signal to enable the TMS 9901 when a CRU software base address of $1FC0_{16}$ is decoded by the PROM. If a conflict exists due to this address being used elsewhere in the system, then other CRU software base addresses can be obtained by re-positioning the jumper to select another address that alleviates the conflict. Table 2-2 shows the available CRU software base addresses and the jumper position that is required to select any of these addresses.

NOTE

If the CRU software base addresses that are provided do not resolve the address conflicts, reprogram another decode PROM that has the desired address in it. Appendix F describes the reprogramming process.



NOTE

Do not jumper E9 to E10 as this will disable the speech module's TMS 9901 address inputs. This position is reserved for future use.

FIGURE 2-2. CRU SOFTWARE BASE ADDRESS SELECTION

TABLE 2-2. CRU SOFTWARE BASE ADDRESS SELECTION

CRU Base Address Selected	Jumper Position
*$1FC0_{16}$	E7 to E8
$0F00_{16}$	E5 to E6
0700_{16}	E3 to E4
0300_{16}	E1 to E2

*$1FC0_{16}$ is the address jumpered at the factory

2.5 JUMPER CONFIGURATION AS SHIPPED

The jumper configuration as shipped is given in Table 2-3.

TABLE 2-3. JUMPER CONFIGURATION AS SHIPPED

FUNCTION	JUMPER	DESCRIPTION
Interrupt level select	E13 to E14	Select interrupt level 12
CRU base address select	E7 to E8	Select CRU base address $1FC0_{16}$

2.6 SAMPLE SYSTEM USING TM 990/306

Figure 2-3 shows a system that can be used to initially checkout the TM 990/306.

2.6.1 Required Equipment

The following items are required to implement the sample program:

- TM 990/306 speech module
- TM 990/10X CPU module
- TM 990/518 or TM 990/518A dc power supply
- TM 990/510 or TM 990/520 card cage
- TM 990/524 connector kit or equivalent
- Suitable terminal that interfaces to CPU module
- Speaker, 8 ohms, 2.5 W.

2.6.2 Hookup Procedure

2.6.2.1 Speaker Hookup. The audio power amplifier circuitry is shown in Figure 2-4. Either connector P2 or P3 can be used for connecting the speaker or an external amplifier to the TM 990/306. The AMP OUT connections (pins P2-16 or P3-22) are used for speaker connections, and the FILTER OUT connections (pins P2-29 or P3-12) are used if an external amplifier system is used.

Connect the 8 ohm speaker to either P3-22 (AMP OUT) and P3-20 (GND) or P2-16 (AMP OUT) and P2-1 (GND). Actually any of the grounded pins that are available on P2 and P3 could have been used for the second (GND) connection.

If it is desired to use an external amplifier system with the TM 990/306, then the external amplifier input should be connected to either P3-12 (FILTER OUT) and P3-20 (GND) or P2-29 (FILTER OUT) and P2-1 (GND). It should be noted that the on-board volume control will have little effect on the volume of an external amplifier, hence a volume control should be provided on the external amplifier if volume control is desired.

P2 and P3 require a 40-pin, 0.100 inch cc mating connector. The TM 990/524 connector kit or the following connectors can be used:

- TI 421121-20 (solder tail)
- 3M 3463-0001 (no ears)
- Viking 3VH20/1JN5 (pierced tail).

2.6.2.2 Power Supply, Terminal, and Card Cage Hookup. The TM 990/10X CPU module should be placed below the speech module in the TM 990/510 or TM 990/520 card cage (See Figure 2-3).

The TM 990/518 or TM 990/518A power supplies can be connected to terminal block TB1 on the rear of the card cage. The power supply connections are shown in Figure 2-5.

The terminal is connected to P2 of the TM 990/10X CPU module (See Figure 2-3).

2.6.2.3 Interrupt Level and CRU Base Address Selection. The "as shipped" interrupt level and CRU base address jumper positions are used for the test routine given in Section 2.8. Interrupt level 12 and CRU base address $1FC0_{16}$ are selected by jumpers connected from E13 to E14 and E7 to E8, respectively.

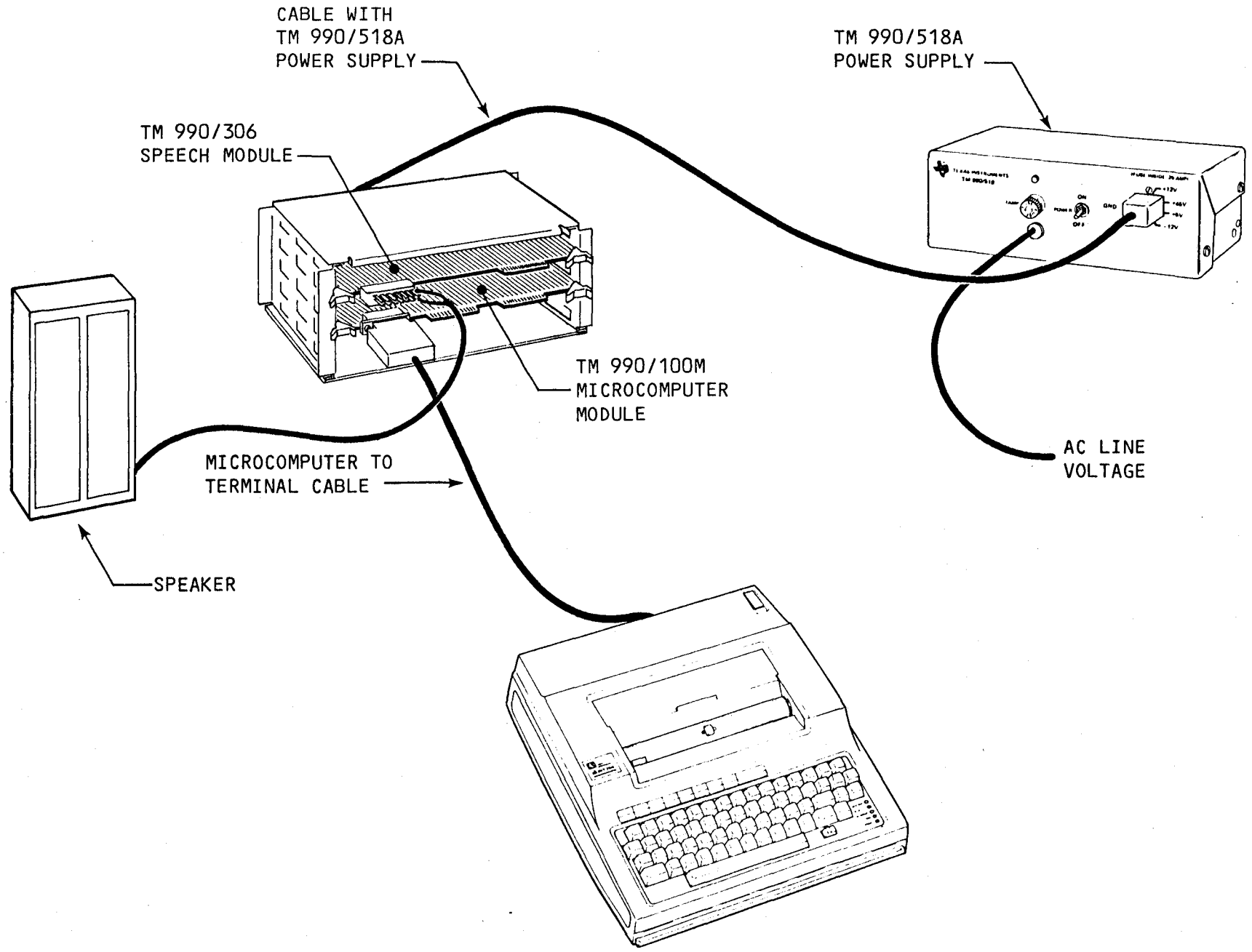
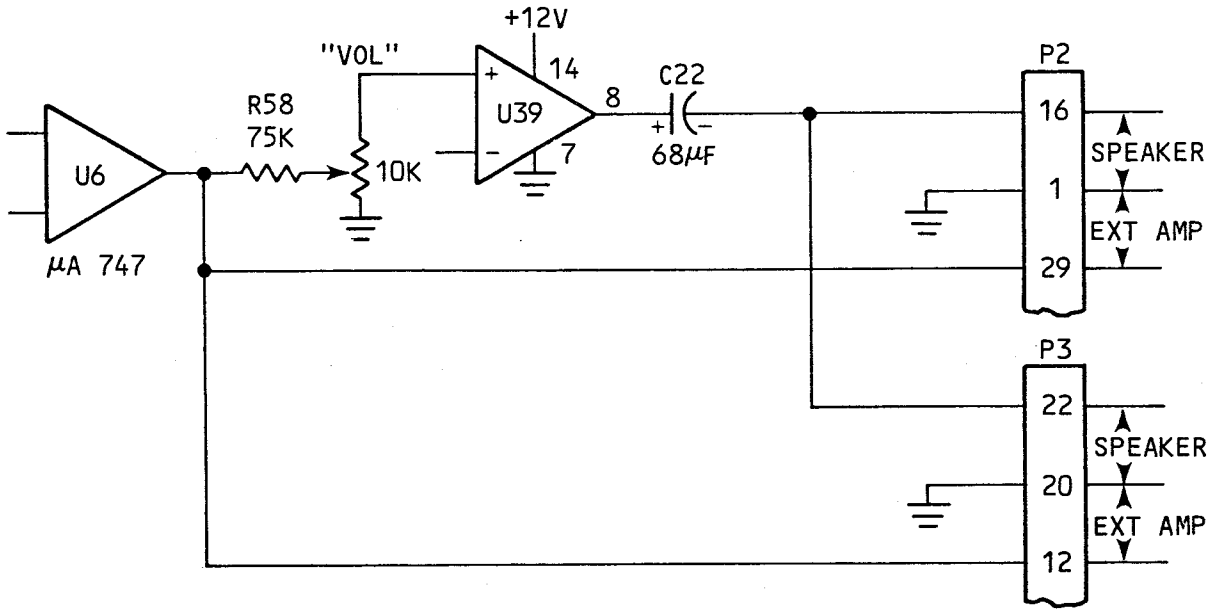


FIGURE 2-3. SAMPLE SYSTEM USING TM 990/306



NOTE

Either connector P2 or P3 can be used for speaker/external amplifier connections.

FIGURE 2-4. SPEAKER/EXTERNAL AMPLIFIER CONNECTIONS

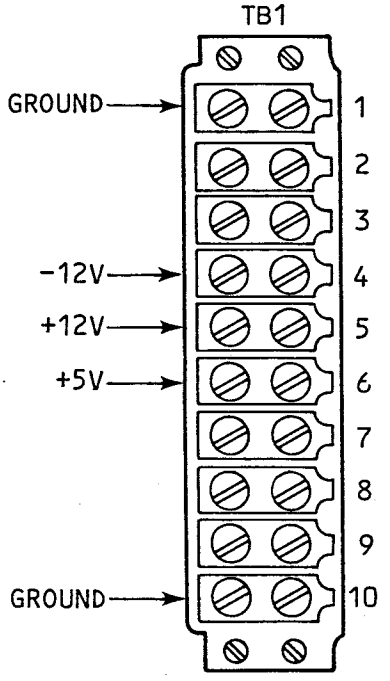


FIGURE 2-5. POWER SUPPLY CONNECTIONS TO TM 990/510 CARD CAGE

2.7 MODES OF OPERATION

The TM 990/306 speech module provides two modes of operation: 1) polled status, and 2) interrupt driven. These two techniques are described in the paragraphs that follow.

2.7.1 Polled Status Mode

The polled status mode is similar to the polling technique used with many CPU module I/O operations. In polling, the CPU module periodically asks I/O devices: "Do you need service?". The CPU's response depends on the answer received from the I/O device. The CPU will either service the I/O device or pass on to check on the need for service of other I/O devices depending on the response from the I/O device.

The TM 990/306's polled status mode involves a similar polling loop procedure - except that the polling involves checking repeatedly to see if the programmed word has been said, and if so, to continue talking as long as any unspoken words from the program exist.

In the polled status mode, the host CPU issues the address of a word (word addresses are included in Appendices C and D), sets the talk command and then waits for the word to be spoken. While the word is being spoken, a talk bit is set: after the talking is completed, this bit is reset. The CPU module now checks to see if any other words to be spoken exist and then directs the speech module to impose a delay, say another word or stop talking.

2.7.2 Interrupt Driven Mode

The interrupt driven mode is similar to the interrupt technique used with many CPU module I/O operations. This technique allows any I/O device connected to the CPU's interrupt line to request service from the CPU. If more than one I/O device should attempt to interrupt the processor simultaneously, then service would be provided depending on the level of the interrupt and the setting of any interrupt masks.

The TM 990/306's interrupt driven mode involves the use of interrupt levels 9 through 12. An interrupt is issued when the speech module stops talking. When the CPU module receives the interrupt, it decides whether to continue talking, delay, or stop talking.

One advantage of using the interrupt driven mode is that it frees the CPU module to do other tasks while the speech module is still talking. This mode results in efficient use of computer time.

2.8 TEST ROUTINE

A sample program that directs the TM 990/306 to say "danger" is given in Figure 2-6. This program can be used to initially checkout the speech module. A TM 990/10X CPU module was used as the TM 990 host CPU. To execute this program, the user can use the TIBUG memory change command to load the opcode values (second column) into the memory address (first column), then execute with the program counter loaded with $FE00_{16}$.

```
M.A.  OPCODE        INSTRUCTION                  DESCRIPTION

FE00  02E0          LWPI  >FE30                  SET WORKSPACE POINTER VALUE
FE02  FE30
FE04  020C          LI    R12,>1FE0              UPPER CRU ADDRESS OF SPEECH MODULE
FE06  1FE0
FE08  1D0F          SBO   15                     SET TMS 2716 EPROM MODE
FE0A  33A0          LDCR  @WORD,14               APPLY WORD TO CRU
FE0C  FE22
FE0E  1E0E          SBZ   14                     START TALK
FE10  020C          LI    R12,>1FC0              LOWER CRU ADDRESS OF SPEECH MODULE
FE12  1FC0
FE14  1F01  WAIT    TB    1                      WAIT FOR SPEECH TO START
FE16  16FE          JNE   WAIT
FE18  020C          LI    R12,>1FE0              UPPER CRU ADDRESS OF SPEECH MODULE
FE1A  1FE0
FE1C  1D0E          SBO   14                     STOP TALKING
FE1E  0460          B     @>0080                 BRANCH TO TIBUG MONITOR
FE20  0080
FE22  01D8  WORD    DATA  >01D8                  ADDRESS OF WORD "DANGER"
```

FIGURE 2-6. TEST ROUTINE

2.9 TM 990/306 AND EXTERNAL HOST CPU CONNECTIONS

The TM 990/306 can be interfaced to other CPUs which do not use the TM 990 system bus. The speaker or external amplifier is connected to front edge connector P2 while the external host CPU is connected to P3 (Note: even-numbered pins, 2-40, are on the top of the connectors and odd-numbered pins, 1-39, are on the bottom; numbering is from left to right). A summary of TM 990/306 P2 and P3 connector pins is given in Table 2-4.

Table 2-5 shows the connections between the TM 990/306 and a TM 990/101 CPU module that is being used as an external host CPU. A polled status example program that uses a TM 990/101 CPU module as an external host is given in Section 3.9. In addition a hardwire connection between INT1- (pin 17 of the CPU module's TMS 9901 I/O controller) and pin P3-23 of connector P3 on the speech module is required. The wire from INT1- can be brought out to the unused pin (P4-4) at the CPU module's P4 connector (which is the microcomputer connector for all signals between the two boards). Also, remove the TMS 9901 on the speech module to prevent contention on the speech module's port P3.

NOTE

If it is desired to use the TM 990/101 as an external host CPU in the interrupt driven mode pullup resistor R12 (2.2K) on the TM990/101 must be removed (As this procedure requires the unsoldering of a component, it is not generally recommended). Removal of R12 allows the *R7 signal from the speech module to drive the external host CPU. Note that this change is NOT needed if the system bus is used as the interface between the CPU and speech modules.

TABLE 2-4. TM 990/306 P2 AND P3 PINS USED IN EXAMPLE PROGRAMS

PIN	FUNCTION	
P2-16	AMPOUT	Speaker Connections
P2-29	FILTEROUT	
P2-1	GROUND	
P3-1	GROUND	
P3-9	S1 (LSB)	14-Bit Address of Word to be Spoken
P3-11	S2	
P3-31	S3	
P3-35	S4	
P3-38	S5	
P3-39	S6	
P3-37	S7	
P3-33	S8	
P3-29	S9	
P3-28	S10	
P3-27	S11	
P3-26	S12	
P3-25	S13	
P3-19	S14 (MSB)	
P3-17	S15	Talk/no-talk bit[1]
P3-15	S16	ROM bit[2]
P3-23	*R7	Status bit[3]

NOTES:

1. Set to zero = talk
   Set to one = do not talk

2. Set to one for TMS 2716 speech set EPROMs

3. Read a one = busy
   Read a zero = not busy

NOTE

Connector plugs of various vendors, including TI, do not necessarily use the numbering scheme on the board edge connector. ALWAYS refer to the board edge when wiring a connector.

TABLE 2-5. PIN LIST FOR EXTERNAL HOST EXAMPLE PROGRAM

Pin at P3 of TM 990/306 Board		Pin at P4 TM 990/101 CPU Board[1]	
P3-1	GROUND	P4-7	GROUND
P3-9	S1 (LSB)	P4-20	P0
P3-11	S2	P4-22	P1
P3-31	S3	P4-14	P2
P3-35	S4	P4-16	P3
P3-38	S5	P4-18	P4
P3-39	S6	P4-10	P5
P3-37	S7	P4-12	P6
P3-33	S8	P4-24	P7
P3-29	S9	P4-26	P8
P3-28	S10	P4-28	P9
P3-27	S11	P4-30	P10
P3-26	S12	P4-32	P11
P3-25	S13	P4-34	P12
P3-19	S14 (MSB)	P4-36	P13
P3-17	S15	P4-38	P14
P3-15	S16	P4-40	P15
P3-23	*R7 (Status bit)	[2]Pin 17 of host TMS 9901 (INT1-)	

NOTES

1. The TM 990/101 CPU module does not communicate to the TM 990/306 via the system backplane; instead, all signals are sent via a cabled connection between edge connector P4 on the TM 990/101 and edge connector P3 on the speech module. This arrangment allows the TM 990/101 board to act entirely as an external host.

2. As pin 17 of the CPU module's TMS 9901 (INT1-) is not connected to the CPU module's connector P4, a hardwire connection is required between pin 17 of the TMS 9901 to the spare pin P4-4 on the CPU module (This allows the user to use only a single cable between P3 (TM 990/306) and P4 (TM 990/101).

SECTION 3

PROGRAMMING

3.1 GENERAL

This section describes the programming aspects relevant to the TM 990/306. The speech module's vocabulary is introduced along with suggestions on possible word construction. Software examples using both modes of operation (polled status and interrupt driven) are given along with additional information regarding the use of an external host CPU.

3.2 VOCABULARY

The speech module's vocabulary consists of the numbers 0 to 12, the alphabet from A to Z, and a base vocabulary consisting of 141 additional words. In addition, there are four words that cause time delays in increments of 80 ms; they are DELAY 1 (80 ms delay), DELAY 2 (160 ms delay), DELAY 3 (240 ms delay), and DELAY 4 (320 ms delay). The user is encouraged to be ingenious in using the speech set. For instance, certain letters sound like words: R like are, N like in, B like be, and C like see. New words can be constructed using a combination of certain letters and exixting words. As an example, N + CREASE sounds like increase and D + CREASE sounds like decrease. A careful study of the speech module's vocabulary (Appendices C and D) would be beneficial to the user. Also demonstration software that illustrates the programming techniques for word construction is given in Appendix B.

3.3 FUNDAMENTAL CONSIDERATIONS

The various words of speech data are contained in TMS 2716 EPROMs which are accessible only to the TMS 1099 4-bit processor on the TM 990/306. This speech data averages 200 bytes per spoken word. An alphabetical listing of the speech set is given in Appendix C and a listing of the speech set by numerical address location is given in Appendix D.

Programming the speech module via the TM 990 system bus consists basically of communicating to the TMS 9901 programmable parallel interface on the speech module via the communications register unit (CRU) interface. Both CRU single-bit and CRU multibit instructions are used in this process which includes:

- Setting a bit that selects the speech data ROM configuration.
- Sending the 14-bit address for the word to be spoken
- Resetting a bit that signals the speech module to start talking
- Monitoring a status bit that indicates that the speech module is through talking (polled status mode) or being notified of this via an interrupt (interrupt driven mode).
- Setting a bit that signals the speech module to stop talking.

3.4 POLLED STATUS TMS 9901 INTERFACE

When a TM 990 CPU module is used as the host computer in the polled mode, all communucation between it and the speech module involves the CRU interface. The CPU communicates with the speech module via the TM 990/306's TMS 9901 parallel I/O controller.

The CPU can address the 32 CRU bits in the speech module's TMS 9901. Figure 3-1 shows the functions of the various bits. The first 16 bits (LSBs) are used for timer control and interrupt masks, while the second 16 bits (MSBs) are used for the parallel data port. Bit number 1 of the first 16 bits is the BUSY or status bit. When this bit is a zero, the speech module is not busy; when it is a one, the speech module is busy and the host computer must wait before applying a new word address to the TMS 9901. Since this bit will not become a one immediately after enabling the speech module due to the time involved in producing speech, the host CPU must monitor this bit and wait for it to become a one (speaking) and then a zero (through speaking) prior to addressing the next word. When the CPU re-tests this bit and finds that it is a zero: the word has been spoken. The address of a word to be spoken occupies 14 bits of the second 16 bits while the remaining 2 bits are used for control. Bit 14 (second bit group) is used to start or stop talk and bit 15 (second bit group) is used to select the TMS 2716 ROM mode.

The TM 990/306 speech module used in the following example program will use a CRU base address of $1FC0_{16}$. Note that this is the starting address for bit 0 of the first group of bits (This address will be referred to as the lower CRU address in Figure 3-2). The starting address for the second group of bits ($1FE0_{16}$) can be obtained by adding 20_{16} to the CRU base address that is selected by jumpers E1-E8 (this address will be referred to as the upper CRU address in Figure 3-2).

Additional information on the TMS 9901 CRU select bit assignments can be found in the TMS 9901 Programmable Systems Interface Manual.

3.5 PROGRAM IMPLEMENTATION

In order to implement a program for the speech module, certain aspects involving communications with the speech module need further comment. There are essentially five aspects involved in communicating with the speech module: each of these aspects are briefly described in the paragraphs that follow. Note that the displacements used with the CRU instructions are taken from Figure 3-1 and refer mainly to the single word example code given in Section 3.6.

3.5.1 Select ROM Mode

The speech data that is available on the TM 990/306 is contained in TMS 2716 EPROMs. The CRU single-bit instruction SBO 15 selects the TMS 2716 EPROM mode while addressing the second half of the TMS 9901.

3.5.2 Loading the Address of the Word to be Spoken

The 14-bit address of the speech data can be loaded using the CRU multibit instruction, LDCR @WORD,14 while addressing the second half of the TMS 9901.

3.5.3 Setting the "Start Talking" Bit

The CRU single-bit instruction SBZ 14 signals the speech module to start talking while addressing the second half of the TMS 9901.

3.5.4 Monitoring the Completion of a Word

Monitoring a status bit to determine if a word has been completed before attempting a new word is necessary because a new address cannot be sent to the

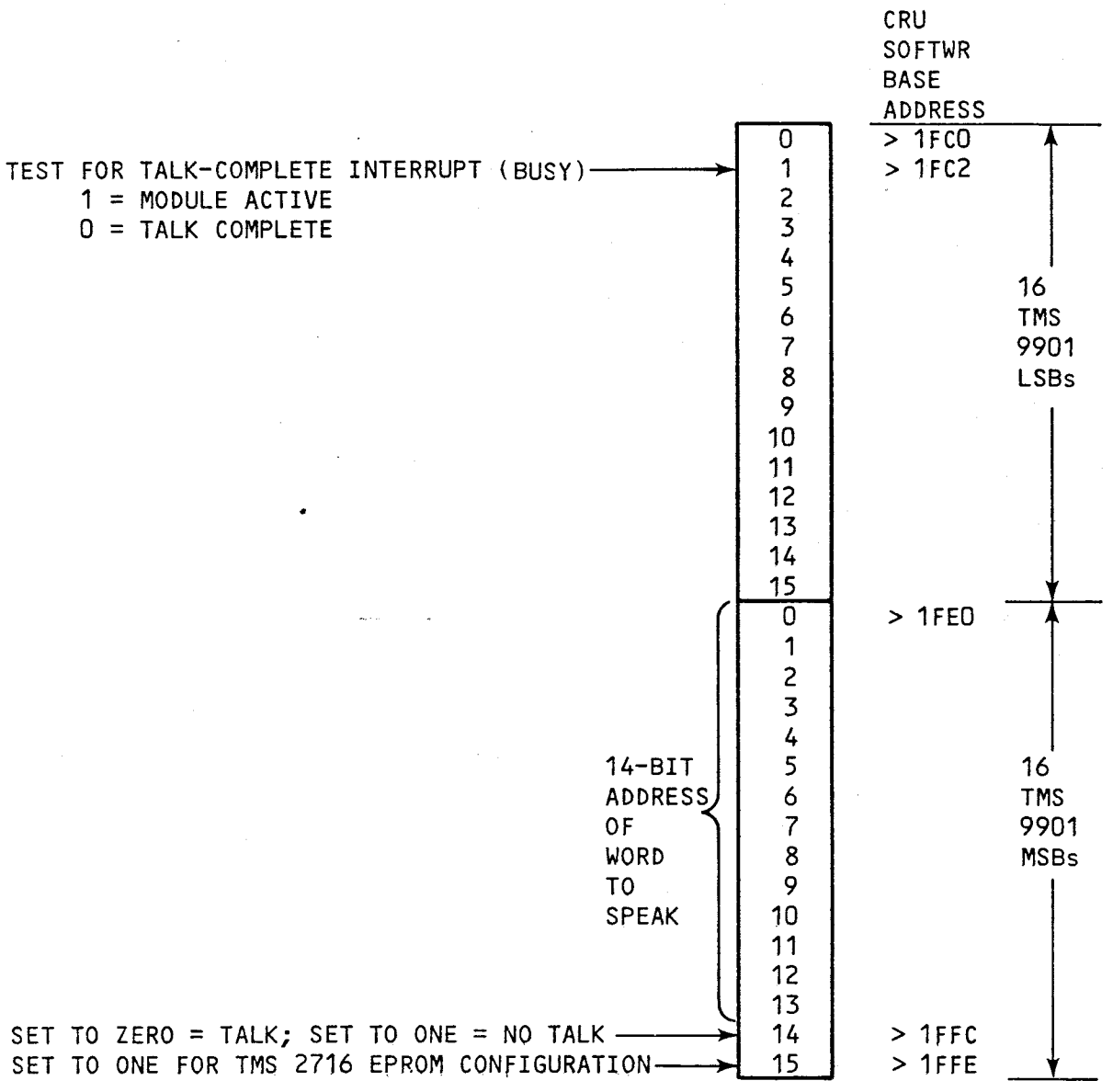


FIGURE 3-1. TMS 9901 INTERFACE USED FOR POLLED COMPLETION

speech module while while it is speaking. If such an address is sent during this time, it will be lost. Since a word of speech takes from 0.5 to 1.0 seconds to complete, it must be determined when the speech module is through so that a new address can be sent.

There are two ways to monitor the completion status of a word: 1) Use the CRU single-bit instruction TB 1 while addressing the first half of the TMS 9901 (tests BUSY bit). This checks to see if the speech module is still talking. A bit value of 1 indicates that the speech module is still active while a bit value of 0 indicates that the previous word has been completed (this technique is used with the polled status mode) or 2) Use the interrupt driven mode of operation, with this mode an interrupt is generated by the speech module's TMS 9901 INTREQ- line. This interrupt is jumper selectable so that the host CPU module's INT9- through INT12- interrupt levels can be utilized. Note that if the first method is used, the BUSY bit will not become active immediately. Thus the program must wait for this bit to be active (a one) before checking for it to become inactive (a zero).

NOTE

After starting to speak, the BUSY bit (CRU bit 1 at the first half of the TMS 9901) can be checked for speaking-completed status (zero = speaking completed). There is a 10 ms time frame between BUSY going to zero (polled mode) or an interrupt issued (interrupt mode) and the start of speaking the next word. During this 10 ms period, a new word address can be presented to the speech module or the stop-talking bit can be enabled. Failure to do one of these operations will cause the previously spoken word to be repeated. Thus, polled operation and interrupt servicing must be timed so that the next operation is set up in this 10 ms time frame.

3.5.5 Setting the "Stop Talking" Bit

The CRU single-bit instruction SBO 14 signals the speech module to stop talking when addressing the second half of the TMS 9901.

3.6 EXAMPLE PROGRAM (POLLED STATUS)

A capsule summary of code using the polled status mode is given in Figure 3-2. The speech module is using CRU address $1FC0_{16}$ (E7 to E8) and the CPU module is using a workspace starting at $FE00_{16}$. This program directs the speech module to say the word "danger".

Paragraphs 3.7 and 3.8 are examples of programs that cause the TM 990/306 to speak phrases. The program in paragraph 3.7 says "Motor five is on fire," and the program in paragraph 3.8 says "Time to call the inspector."

NOTE

The example programs in Figures 3-3, 3-4, and 3-5 can be executed by loading the object code (listing, column 3) at the memory address (column 2) using the memory inspect/change command (M). Then issue the execute command (E) with the program counter set to $FE00_{16}$.

```
BUSY    EQU   1
        LWPI  >FE00       SET WORKSPACE POINTER TO  FE00
        LI    R12,>1FE0   TMS 9901 UPPER 16-BITS ADDRESSED ON CRU
        SBO   15          TMS 2716 EPROM MODE
        LDCR  @WORD,14    SEND ADDRESS OF WORD TO TMS 9901
        SBZ   14          BEGIN TALKING
        LI    R12,>1FC0   TMS 9901 LOWER 16-BITS ADDRESSED ON CRU
NOTACT  TB    BUSY        SPEECH MODULE BECOME BUSY?
        JNE   NOTACT      NO, LOOP UNTIL ACTIVE
SPEKNG  TB    BUSY        THROUGH TALKING?
        JEQ   SPEKNG      NO, LOOP UNTIL THROUGH SPEAKING
        LI    R12,>1FE0   TMS 9901 UPPER 16-BITS ADDRESSED ON CRU
        SBO   14          STOP TALKING
WORD    DATA  >01D8       ADDRESS OF WORD DANGER
```

FIGURE 3-2. SINGLE WORD EXAMPLE PROGRAM

3.7 POLLED STATUS EXAMPLE PROGRAM, TM 990 HOST

An example program that uses the speech module to say the phrase "Motor five is on fire" is given in Figure 3-3. This example uses the polled status mode of operation with a TM 990/100M serving as the TM 990 host CPU module.

3.8 INTERRUPT DRIVEN EXAMPLE PROGRAM, TM 990 HOST

The simplest mode of operation (polled status) uses the TM 990/10XM CPU module as host and polls the speech module for status. For the word "on," approximately 180 ms of computer time are lost waiting for the BUSY line to become inactive while the time for the word "initialize" stretches out to approximately 250 ms.

In order to eliminate the time spent polling the speech module and increase throughput, interrupt driven software may be used. As soon as the speech module is no longer busy, an interrupt occurs notifying the processor that another word may be spoken at that time. As shown in Table 2-1, interrupt levels 9 through 12 may be used depending on the jumper plug position. Figure 3-4 shows the TMS 9901 interface with the interrupt bits shown. Figure 3-5 is an example of software using an interrupt driven (interrupt 12) TM 990/101M as host. this program directs the speech module to say the phrase "Time to call the inspector." The TM 990/101M CPU module contains 401-3 TIBUG which has vectors for these interrupts. The 401-1 TIBUG on the TM 990/100M CPU module does not have vectors for these interrupts.

NOTE

In order to use interrupts 9 to 12 on the TM 990/100M CPU module with TM 990/401-1 TIBUG, the respective interrupt vector locations in lower EPROM must have been loaded by the user with vector values (vector values are already present on the TM 990/101M CPU module with TM 990/401-3 TIBUG). The TM 990/401-3 TIBUG can be inserted on the TM 990/100M board in place of the TM 990/401-1 TIBUG.

```
FIG31      SDSMAC 3.2.0 78.274   08:58:06 THURSDAY, DEC 13, 1979.
TM 990/306 DEMONSTRATION                                    PAGE 0002
0002                    IDT  'FIG31'
0003              ***************************************************************
0004              *                                                             *
0005              *   TITLE:       TM990/306 DEMONSTRATION SOFTWARE             *
0006              *                                                             *
0007              *   DATE:        10/29/79                                     *
0008              *                                                             *
0009              *   FUNCTION:    SOFTWARE FOR POLLED STATUS, TM 990 HOST      *
0010              *                                                             *
0011              *   DESCRIPTION: THE ADDRESSES OF THE WORDS IN THE PHRASE     *
0012              *                "MOTOR FIVE IS ON FIRE" ARE STORED AT        *
0013              *                ADDRESS FEA0 (PH1) AND FOLLOWING.  THROUGH   *
0014              *                INDEXED, INDIRECT ADDRESSING, THE HANDLER    *
0015              *                LOADS THESE ADDRESSES INTO THE CRU AND       *
0016              *                TOGGLES BITS 14 AND 15 TO PLACE THE /306     *
0017              *                INTO EPROM MODE AND START SPEECH OUTPUT.     *
0018              *                                                             *
0019              *                AFTER EACH WORD IS LOADED, THE BUSY BIT IS   *
0020              *                TESTED SO THAT THE SUCCEEDING ADDRESSES      *
0021              *                ARE NOT SENT TO THE SPEECH MODULE WHEN IT    *
0022              *                IS NOT ABLE TO ACCEPT THEM.  FOR LONG        *
0023              *                WORDS SUCH AS "INITIALIZE", THE COMPUTER     *
0024              *                MAY BE TIED UP IN THIS DELAY LOOP FOR AS     *
0025              *                LONG AS 250 MS.                              *
0026              *                                                             *
0027              *   AUTHOR:      DALE R. GIBBLE                               *
0028              *                                                             *
0029              ***************************************************************
0030              *
0031              *  REGISTER EQUATES
0032              *
0033      0000    R0      EQU  0
0034      0001    R1      EQU  1
0035      0002    R2      EQU  2
0036      0003    R3      EQU  3
0037      0004    R4      EQU  4
0038      000C    R12     EQU  12
0039      0080    TIBUG   EQU  >0080
0040      FEA0    PH1     EQU  >FEA0
0041      FE60    WPMAIN  EQU  >FE60
0042              *
0043      0316    FIRE    EQU  >0316        FIRE
0044      03C0    FIVE    EQU  >03C0        FIVE
0045      04B4    IS      EQU  >04B4        IS
0046      0666    ON      EQU  >0666        ON
0047      0F6A    MOTOR   EQU  >0F6A        MOTOR
0048              *
0049      1FC0    CRULOW  EQU  >1FC0        /306 LOWER 16 CRU BITS
0050      1FE0    CRUUP   EQU  >1FE0        /306 UPPER 16 CRU BITS
0051              *
0052              *  MODULE TO SAY PHRASE
0053              *
0054 FE00                 AORG >FE00        SUGGESTED LOADING POINT
```

FIGURE 3-3. PROGRAM LISTING, POLLED STATUS, TM 990 HOST (Sheet 1 of 2)

```
 0055 FE00 02E0  ENTRY  LWPI PH1          PHRASE WORKSPACE
      FE02 FEA0
 0056 FE04 0200         LI   R0,MOTOR     ADDRESS OF MOTOR
      FE06 0F6A
 0057 FE08 0201         LI   R1,FIVE      ADDRESS OF FIVE
      FE0A 03C0
 0058 FE0C 0202         LI   R2,IS        ADDRESS OF IS
      FE0E 04B4
 0059 FE10 0203         LI   R3,ON        ADDRESS OF ON
      FE12 0666
 0060 FE14 0204         LI   R4,FIRE      ADDRESS OF FIRE
      FE16 0316
 0061 FE18 02E0         LWPI WPMAIN       MAIN WORKSPACE
      FE1A FE60
 0062 FE1C 0200         LI   R0,PH1       ADDRESS OF FIRST WORD IN PHRASE
      FE1E FEA0
 0063 FE20 0202         LI   R2,PH1+12    ADDRESS OF LAST WORD IN PHRASE
      FE22 FEAC
 0064 FE24 020C  NEXT   LI   R12,CRUUP    /306 CRU ADDRESS
      FE26 1FE0
 0065 FE28 C070         MOV  *R0+,R1      GET ADDRESS
 0066 FE2A 8080         C    R0,R2        CHECK FOR LAST WORD
 0067 FE2C 130A         JEQ  STOPT        IF EQUAL, STOP TALKING
 0068 FE2E 1D0F         SBO  15           SET /306 TO EPROM MODE
 0069 FE30 3381         LDCR R1,14        SAY NEXT WORD
 0070 FE32 1E0E         SBZ  14           START TALKING
 0071 FE34 020C         LI   R12,CRULOW   LOWER HALF BASE ADDRESS
      FE36 1FC0
 0072 FE38 1F01  I1     TB   1            TEST BUSY BIT FOR ACTIVE
 0073 FE3A 16FE         JNE  I1           NO, CHECK AGAIN
 0074 FE3C 1F01  I2     TB   1            TEST BUSY BIT FOR INACTIVE
 0075 FE3E 13FE         JEQ  I2           YES, CHECK AGAIN
 0076 FE40 10F1         JMP  NEXT         GET NEXT WORD
 0077 FE42 1D0E  STOPT  SBO  14           STOP TALKING
 0078 FE44 0460         B    @TIBUG       GO TO TIBUG MONITOR
      FE46 0080
 0079                   END
NO ERRORS,     NO WARNINGS
```

FIGURE 3-3. PROGRAM LISTING, POLLED STATUS, TM 990 HOST (Sheet 2 of 2)

3.9 POLLED STATUS EXAMPLE PROGRAM, EXTERNAL HOST CPU

The TM 990/306 speech module is unique in that it can easily be interfaced to any external host CPU which does not use the TM 990 system bus whether it be a minicomputer or large mainframe. Speech module connectors P2 and P3 have been designed so that the computer interface can be connected to P3 while the analog signals to the external amplifier or speaker may be connected to P2. The pinout of P2 and P3 is given in Appendix G. Section 2.9 shows the pins used in the examples in this manual which uses a TM 990/101M CPU module as the external host. Although this is a TM 990 CPU, it has been connected externally to the TM 990/306 board; that is, it acts entirely as if it was a non-TM 990 CPU with no signals transmitted via the system bus.

Signals S1 through S14 (at connector P3) are the 14 data lines which represent the address of a stored speech word. S15 is the talk command (set to zero for talk) and S16 determines which ROM type contains the speech data (set to one for TMS 2716 EPROM's). These signals are used identically to those of section 3.7 for the TM 990 host. The major difference between using the system bus for communication and not using the system bus is that the CRU (communications register unit of the TMS 9900) is not used for communication.

In the example used herein, status (talking completed) is read at speech-board signal line *R7 by the TM 990/101 microcomputer via its INT1- line. This connection is made via a hardwire between INT1- (pin 17 of the microcomputer TMS 9901) and pin P3-23 of connector P3 on the speech module. The wire from INT1- is brought out to an unused pin at microcomputer connector P4 (which is the microcomputer connector for all signals between the two boards). Also, remove the TMS 9901 on the speech module in order to avoid line contention to the speech module's port P3.

Software illustrating external host polled status is shown in Figure 3-6. Although a TM 990/101 microcomputer is used, any minicomputer or microcomputer with a parallel I/O port could control the TM 990/306 speech module in a similar manner. In order to use interrupts 9 to 12 on the TM 990/100M CPU module with TM 990/4-1-1 TIBUG, it should be noted that the respective interrupt vector locations in lower EPROM must have been loaded by the user with vector values (vector values are already present on the TM 990/101M CPU

3.10 DEMONSTRATION SOFTWARE

The TM 990/426 demo software provides the capability to utilize the TM 990/306 speech module. Such operations as saying the entire vocabulary of the demonstration software, saying pre-defined phrases, and creating and saying user defined phrases are all provided. Extensive prompting and a help listing are provided. to make the software easy to use. A listing of the TM 990/426 demo software is provided in Appendix B.

3.11 TMS 9901 INTERVAL TIMER

A detailed discussion of the TMS 9901 interval timer can be found in the TMS 9901 data manual. The user can program the timer to issue an interrupt when the programmed countdown reaches zero, or the timer value can be checked through the CRU to check for elapsed time. A fully loaded timer counts down in 349 milliseconds. In the interrupt mode, the TMS 9901 must be set to an interrupt level of 3 or greater. Place the TMS 9901 CRU address in R12; set to clock mode and load the timer; final bit loaded triggers timer countdown. At countdown, an interrupt (level as selected on the board) will be issued.

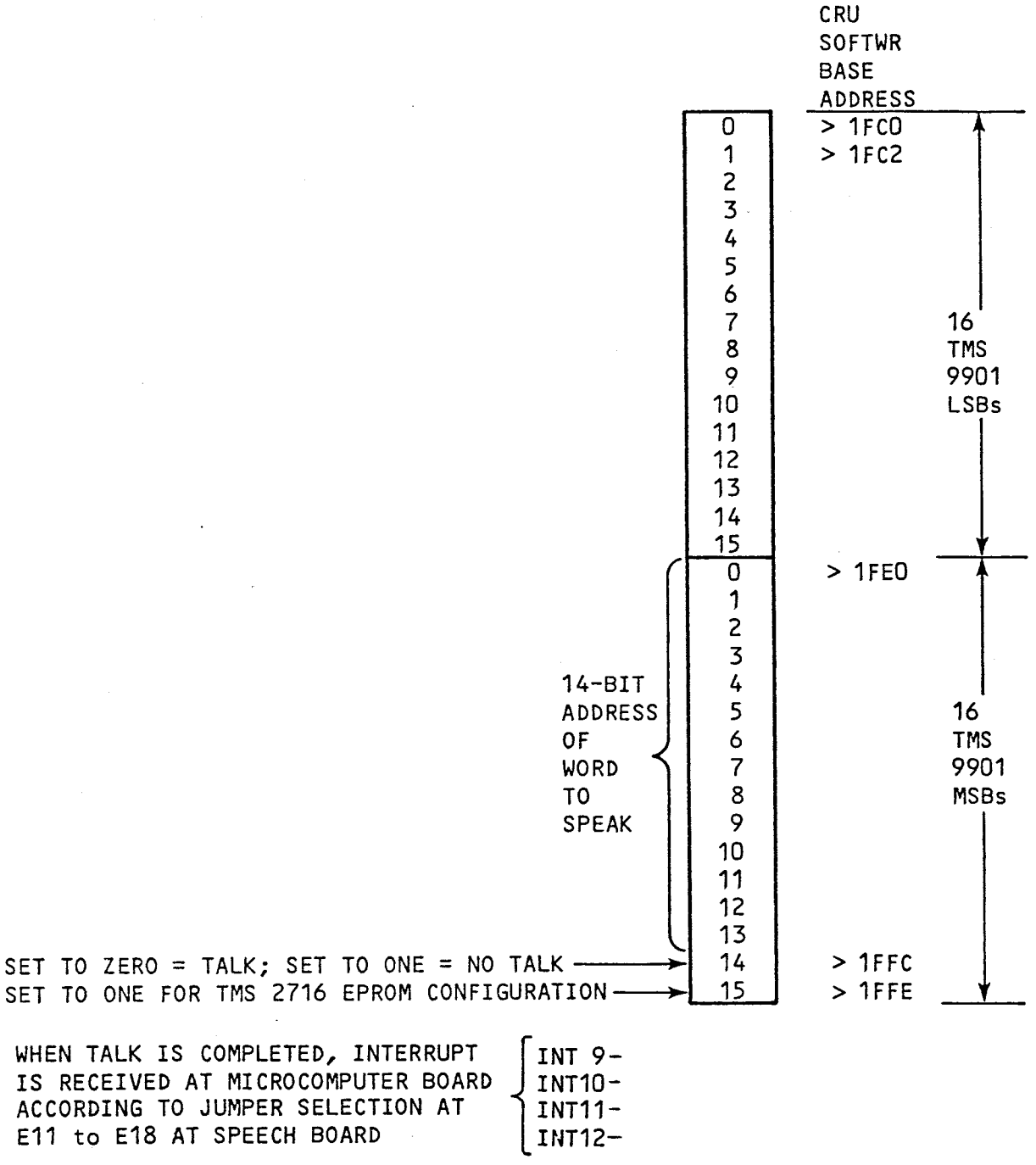


FIGURE 3-4. TMS 9901 INTERFACE USED FOR INTERRUPT DRIVEN COMPLETION

```
0002                    IDT  'FIG32'
0003          ***************************************************************
0004          *                                                             *
0005          *   TITLE:       TM990/306 DEMONSTRATION SOFTWARE             *
0006          *                                                             *
0007          *   DATE:        10/23/79                                     *
0008          *                                                             *
0009          *   FUNCTION:    SOFTWARE FOR INTERRUPT DRIVEN SOFTWARE,      *
0010          *                       TM 990 AS HOST                        *
0011          *                                                             *
0012          *   DESCRIPTION: THE ADDRESSES OF THE WORDS IN THE PHRASE      *
0013          *                "TIME TO CALL THE INSPECTOR" ARE STORED      *
0014          *                AT ADDRESS FEA0 (PH1) AND FOLLOWING.         *
0015          *                THE FIRST WORD IS SPOKEN AFTER EXECUTION     *
0016          *                OF THE MAIN LINE CODE FOLLOWING "ENTRY".     *
0017          *                WHEN THE TALK COMMAND IS GIVEN, THE BUSY     *
0018          *                LINE GOES ACTIVE.  AFTER  THE WORD IS        *
0019          *                SPOKEN, THE BUSY LINE GOES INACTIVE FOR      *
0020          *                10 MS, CAUSING AN INTERRUPT AT THE TMS       *
0021          *                9901 ON THE SPEECH MODULE.  THIS INTERRUPT   *
0022          *                IS JUMPER SELECTABLE AS INTERRUPT 9, 10,     *
0023          *                11 OR 12 OF THE TM 990 COMPUTER MODULE.      *
0024          *                                                             *
0025          *                THE INTERRUPT SERVICE ROUTINE RECOGNIZES     *
0026          *                THAT THE SPEECH MODULE IS NOT TALKING.       *
0027          *                IT THEN LOADS THE ADDRESS OF THE NEXT        *
0028          *                WORD TO BE SPOKEN AND TURNS ON THE TALK      *
0029          *                COMMAND.  AFTER THE LAST WORD IS SPOKEN,     *
0030          *                THE TALK COMMAND IS TURNED OFF, AND          *
0031          *                CONTROL IS PASSED TO THE TIBUG MONITOR.      *
0032          *                                                             *
0033          *   AUTHOR:      DALE R. GIBBLE                               *
0034          *                                                             *
0035          ***************************************************************
0036          *
0037          *      REGISTER EQUATES
0038          *
0039    0000  R0      EQU  0
0040    0001  R1      EQU  1
0041    0002  R2      EQU  2
0042    0003  R3      EQU  3
0043    0004  R4      EQU  4
0044    000C  R12     EQU  12
0045    0080  TIBUG   EQU  >0080
0046    FFA0  PH1     EQU  >FFA0
0047    FF60  WPMAIN  EQU  >FF60
0048    FF40  WPIN12  EQU  >FF40
0049    FED2  IN12PC  EQU  >FED2        INTERRUPT 12 PC VECTOR
0050          *
0051    08B6  TIME    EQU  >08B6        TIME
0052    0900  TWO     EQU  >0900        TWO
0053    15E4  CALL    EQU  >15E4        CALL
0054    087A  THE     EQU  >087A        THE
```

FIGURE 3-5. PROGRAM LISTING, INTERRUPT DRIVEN, TM 990 HOST (Sheet 1 of 3)

```
0055       044A  INSPTR EQU   044A        INSPECTOR
0056             *
0057       0100  CPULOW EQU   0100        CPU LOWER 16 CRU BITS
0058       1FC0  SPHLOW EQU   1FC0        /306 LOWER 16 CRU BITS
0059       1FE0  SPHUP  EQU   1FE0        /306 UPPER 16 CRU BITS
0060             *
0061             *  SET UP INTERRUPT LINKING AREA
0062             *
0063 0000        STARTX
0064 FE00              AORG  FE00        SUGGESTED LOAD POINT
0065 FE00 02E0         LWPI PH1          PHRASE WORKSPACE
     FE02 FFA0
0066 FE04 0201         LI   R1,IN12PC    INTERRUPT 12 PC VECTOR
     FE06 FED2
0067 FE08 0200         LI   R0, 420      BLWP CODE
     FE0A 0420
0068 FE0C CC40         MOV  R0,*R1+      MOVE TO INTERRUPT AREA
0069 FE0E 0200         LI   R0,INTSVC    ADDRESS OF ISR
     FE10 FE64
0070 FE12 CC40         MOV  R0,*R1+      MOVE TO INTERRUPT AREA
0071 FE14 0200         LI   R0, 380      RTWP CODE
     FE16 0380
0072 FE18 C440         MOV  R0,*R1       MOVE TO INTERRUPT AREA
0073             *
0074             *  MODULE TO SAY PHRASE
0075             *
0076 FE1A 0200 ENTRY   LI   R0,TIME      ADDRESS OF TIME
     FE1C 08B6
0077 FE1E 0201         LI   R1,TWO       ADDRESS OF TWO
     FE20 0900
0078 FE22 0202         LI   R2,CALL      ADDRESS OF CALL
     FE24 15E4
0079 FE26 0203         LI   R3,THE       ADDRESS OF THE
     FE28 087A
0080 FE2A 0204         LI   R4,INSPTR    ADDRESS OF INSPECTOR
     FE2C 044A
0081 FE2E 02E0         LWPI WPMAIN       MAIN WORKSPACE
     FE30 FF60
0082 FE32 0200         LI   R0,PH1       ADDRESS OF FIRST WORD IN PHRASE
     FE34 FFA0
0083 FE36 0202         LI   R2,PH1+10    ADDRESS OF LAST WORD IN PHRASE
     FE38 FFAA
0084 FE3A C802         MOV  R2,@WPIN12+4 MOVE TO INTERRUPT WORKSPACE
     FE3C FF44
0085 FE3E 020C         LI   R12,SPHUP    /306 CRU ADDRESS
     FE40 1FE0
0086 FE42 C070         MOV  *R0+,R1      GET ADDRESS
0087 FE44 C800         MOV  R0,@WPIN12   MOVE TO INTERRUPT WORKSPACE
     FE46 FF40
0088 FE48 1D0F         SBO  15           SET /306 TO EPROM MODE
0089 FE4A 3381         LDCR R1,14        SAY FIRST WORD
0090 FE4C 1E0E         SBZ  14           START TALKING
0091             *  SET /306 INTERRUPTS
```

FIGURE 3-5. PROGRAM LISTING, INTERRUPT DRIVEN, TM 990 HOST (Sheet 2 of 3)

```
FIG32      SDSMAC 3.2.0 78.274   09:42:49 THURSDAY, DEC 13, 1979.
TM 990/306 DEMONSTRATION                                    PAGE 0004
 0092 FE4E 020C          LI   R12,SPHLOW   BASE ADDRESS, LOWER HALF
      FE50 1FC0
 0093 FE52 1E00          SBZ  0            /306 TO INT MODE
 0094 FE54 1D01          SBO  1            /306 ENABLE INT1
 0095              *  SET /101 INTERRUPTS
 0096 FE56 020C          LI   R12,CPULOW   BASE ADDRESS FOR 101 9901
      FE58 0100
 0097 FE5A 1E00          SBZ  0            101 TO INT MODE
 0098 FE5C 1D0C          SBO  12           ENABLE INT12
 0099 FE5E 0300          LIMI 12           /101 ENABLE INT12
      FE60 000C
 0100 FE62 10FF          JMP  $            WAIT FOR INTERRUPT
 0101              *
 0102              *  INTERRUPT 12 SERVICE ROUTINE - 281 NOT TALKING
 0103              *
 0104 FE64 FF40  INTSVC DATA WPIN12,$+2    WP,PC
      FE66 FE68
 0105 FE68 020C          LI   R12,SPHLOW   LOWER HALF BASE ADDRESS
      FE6A 1FC0
 0106 FE6C 1F01  I1      TB   1            TEST BUSY BIT FOR ACTIVE
 0107 FE6E 16FE          JNE  I1           NO, CHECK AGAIN
 0108 FE70 8080          C    R0,R2        CHECK FOR LAST PHRASE
 0109 FE72 1605          JNE  I9           IF NOT, RESTART
 0110 FE74 020C          LI   R12,SPHUP    /306 UPPER 16 CRU BASE ADDRESS
      FE76 1FE0
 0111 FE78 1D0E          SBO  14           STOP TALKING
 0112 FE7A 0460          B    @TIBUG       RETURN TO TIBUG MONITOR
      FE7C 0080
 0113 FE7E C070  I9      MOV  *R0+,R1      GET ADDRESS
 0114 FE80 020C          LI   R12,SPHUP    UPPER HALF BASE ADDRESS
      FE82 1FE0
 0115 FE84 33C1          LDCR R1,15        LOAD NEXT WORD
 0116 FE86 1E0E          SBZ  14           START TALKING
 0117 FE88 0380          RTWP
 0118                    END
NO ERRORS,     NO WARNINGS
```

FIGURE 3-5. PROGRAM LISTING, INTERRUPT DRIVEN, TM 990 HOST (Sheet 3 of 3)

```
FIG33      SDSMAC 3.2.0 78.274   09:00:31 THURSDAY, DEC 13, 1979.
TM 990/306 DEMONSTRATION                                    PAGE 0002
 0002                    IDT  'FIG33'
 0003              ***************************************************************
 0004              *                                                             *
 0005              *   TITLE:       TM990/306 DEMONSTRATION SOFTWARE             *
 0006              *                                                             *
 0007              *   DATE:        10/29/79                                     *
 0008              *                                                             *
 0009              *   FUNCTION:    SOFTWARE FOR POLLED STATUS,                  *
 0010              *                     NON-TM 990 HOST                         *
 0011              *                                                             *
 0012              *   DESCRIPTION: THE ADDRESSES OF THE WORDS IN THE PHRASE     *
 0013              *                "THE LINE VOLTS IS LOW" ARE STORED AT        *
 0014              *                ADDRESS FEA0 (PH1) AND FOLLOWNG.  ALL        *
 0015              *                COMMUNICATION IS DONE THROUGH THE            *
 0016              *                PARALLEL I/O PORT OF THE TM 990/101 SINCE    *
 0017              *                IT IS SIMULATING A NON-TM 990 HOST.  A       *
 0018              *                14-BIT ADDRESS IS SENT TO THE SPEECH         *
 0019              *                MODULE ALONG WITH CONTROL BIT 15 WHICH       *
 0020              *                GIVES THE TALK COMMAND AND CONTROL BIT       *
 0021              *                16 WHICH DETERMINES WHETHER SPEECH DATA      *
 0022              *                IS STORED IN EPROM OF MASK PRHASE ROM.       *
 0023              *                                                             *
 0024              *   AUTHOR:      DALE R. GIBBLE                               *
 0025              *                                                             *
 0026              ***************************************************************
 0027              *
 0028              *  REGISTER EQUATES
 0029              *
 0030      0000    R0      EQU  0
 0031      0001    R1      EQU  1
 0032      0002    R2      EQU  2
 0033      0003    R3      EQU  3
 0034      0004    R4      EQU  4
 0035      000C    R12     EQU  12
 0036      0080    TIBUG   EQU  >0080
 0037      FEA0    PH1     EQU  >FEA0
 0038      FE60    WPMAIN  EQU  >FE60
 0039              *
 0040      1E8E    LINE    EQU  >1E8E        LINE
 0041      36FA    VOLTS   EQU  >36FA        VOLTS
 0042      04B4    IS      EQU  >04B4        IS
 0043      281C    LOW     EQU  >281C        LOW
 0044      087A    THE     EQU  >087A        THE
 0045              *
 0046      0100    CRULOW  EQU  >0100        /306 LOWER 16 CRU BITS
 0047      0120    CRUUP   EQU  >0120        /306 UPPER 16 CRU BITS
 0048              *
 0049              *  MODULE TO SAY PHRASE
 0050              *
 0051 0000         ENTRY
 0052 FE00               AORG >FE00         SUGGESTED LOAD POINT
 0053 FE00 02E0          LWPI PH1           PHRASE WORKSPACE
      FE02 FEA0
```

FIGURE 3-6. PROGRAM LISTING, POLLED STATUS, EXTERNAL HOST (Sheet 1 of 2)

```
FIG33      SDSMAC 3.2.0 78.274   09:00:31 THURSDAY, DEC 13, 1979.
TM 990/306 DEMONSTRATION                                    PAGE 0003
 0054 FE04 0200          LI   R0,THE       ADDRESS OF THE
      FE06 087A
 0055 FE08 0201          LI   R1,LINE      ADDRESS OF LINE
      FE0A 1E8E
 0056 FE0C 0202          LI   R2,VOLTS     ADDRESS OF VOLTS
      FE0E 36FA
 0057 FE10 0203          LI   R3,IS        ADDRESS OF IS
      FE12 04B4
 0058 FE14 0204          LI   R4,LOW       ADDRESS OF LOW
      FE16 281C
 0059 FE18 02E0          LWPI WPMAIN       MAIN WORKSPACE
      FE1A FE60
 0060 FE1C 0200          LI   R0,PH1       ADDRESS OF FIRST WORD IN PHRASE
      FE1E FEA0
 0061 FE20 0202          LI   R2,PH1+12    ADDRESS OF LAST WORD IN PHRASE
      FE22 FEAC
 0062 FE24 020C  NEXT    LI   R12,CRUUP    /101 CRU ADDRESS
      FE26 0120
 0063 FE28 C070          MOV  *R0+,R1      GET ADDRESS
 0064 FE2A 8080          C    R0,R2        CHECK FOR LAST WORD
 0065 FE2C 130A          JEQ  STOPT        IF EQUAL, STOP TALKING
 0066 FE2E 1D0F          SBO  15           SET /306 TO EPROM MODE
 0067 FE30 3381          LDCR R1,14        SAY NEXT WORD
 0068 FE32 1E0E          SBZ  14           START TALKING
 0069 FE34 020C          LI   R12,CRULOW   LOWER HALF BASE ADDRESS
      FE36 0100
 0070 FE38 1F01  I1      TB   1            TEST BUSY BIT FOR ACTIVE
 0071 FE3A 16FE          JNE  I1           NO, CHECK AGAIN
 0072 FE3C 1F01  I2      TB   1            TEST BUSY BIT FOR INACTIVE
 0073 FE3E 13FE          JEQ  I2           YES, CHECK AGAIN
 0074 FE40 10F1          JMP  NEXT         GET NEXT WORD
 0075 FE42 1D0E  STOPT   SBO  14           STOP TALKING
 0076 FE44 0460          B    @TIBUG       GO TO TIBUG MONITOR
      FE46 0080
 0077                    END
NO ERRORS,     NO WARNINGS
```

FIGURE 3-6. PROGRAM LISTING, POLLED STATUS, EXTERNAL HOST (Sheet 2 of 2)

APPENDIX A

SCHEMATICS

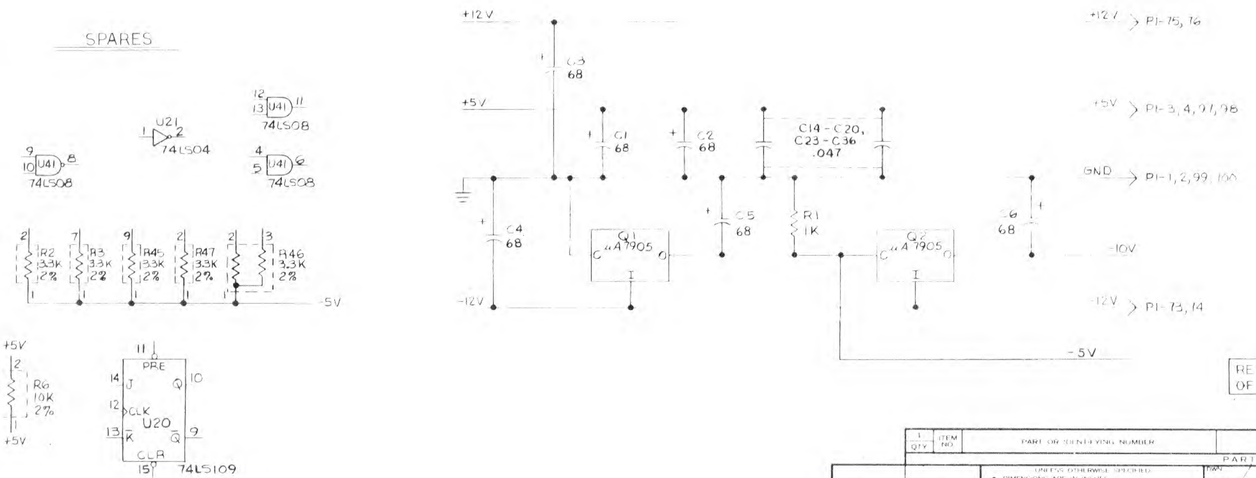
DWG NO 994644 SH 1

REV	DESCRIPTION	DATE	APPROVED
A	DESIGN CHANGE; PRERELEASE	11/5/79	
B	CN454351 INFORMAL 12/19/79	12/19/79	
C	CN454361	5/13/80	

NOTES: UNLESS OTHERWISE SPECIFIED

1. ALL RESISTORS ARE 1/4 W, 5%
2. RESISTANCE VALUES ARE IN OHMS
3. CAPACITANCE VALUES ARE IN MICROFARADS
4. ALL DIODES ARE IN914B
5. FOR FUTURE EXPANSION

REFERENCE DESIGNATORS	
USED	NOT USED
C1-C38	
CR1-CR16	
E1-E18	
P1-P3	
Q1-Q7	
R1-R7, R18-R59	R8-R17
U1-U41	
TP1, TP2	



REV STATUS	REV	C	C	C	A	C
OF SHEETS	SH	1	2	3	4	5

PARTS LIST

QTY	ITEM NO	PART OR IDENTIFYING NUMBER	NOMENCLATURE OR DESCRIPTION	PROCUREMENT SPECIFICATION	NOTES

NEXT ASSY	USED ON
	8119
994646	8117

APPLICATION

UNLESS OTHERWISE SPECIFIED
- DIMENSIONS ARE IN INCHES
- TOLERANCES: ANGLES ±1°; 3 PLACE DECIMALS ±.010; 2 PLACE DECIMALS ±.02
- INTERPRET DRAWING PER MIL-D-1000
- REMOVE ALL BURRS AND SHARP EDGES
- CONCENTRICITY MACHINED DIAMETERS
- DIMENSIONAL LIMITS APPLY BEFORE PROCESSES
- PARENTHETICAL INFO FOR REF ONLY

HOLE TOLERANCE

DWN
CHK 10/15/79
ENGR
APVD
QA
CONTR

TEXAS INSTRUMENTS INCORPORATED Dallas, Texas

DIAGRAM, LOGIC TM 990/306

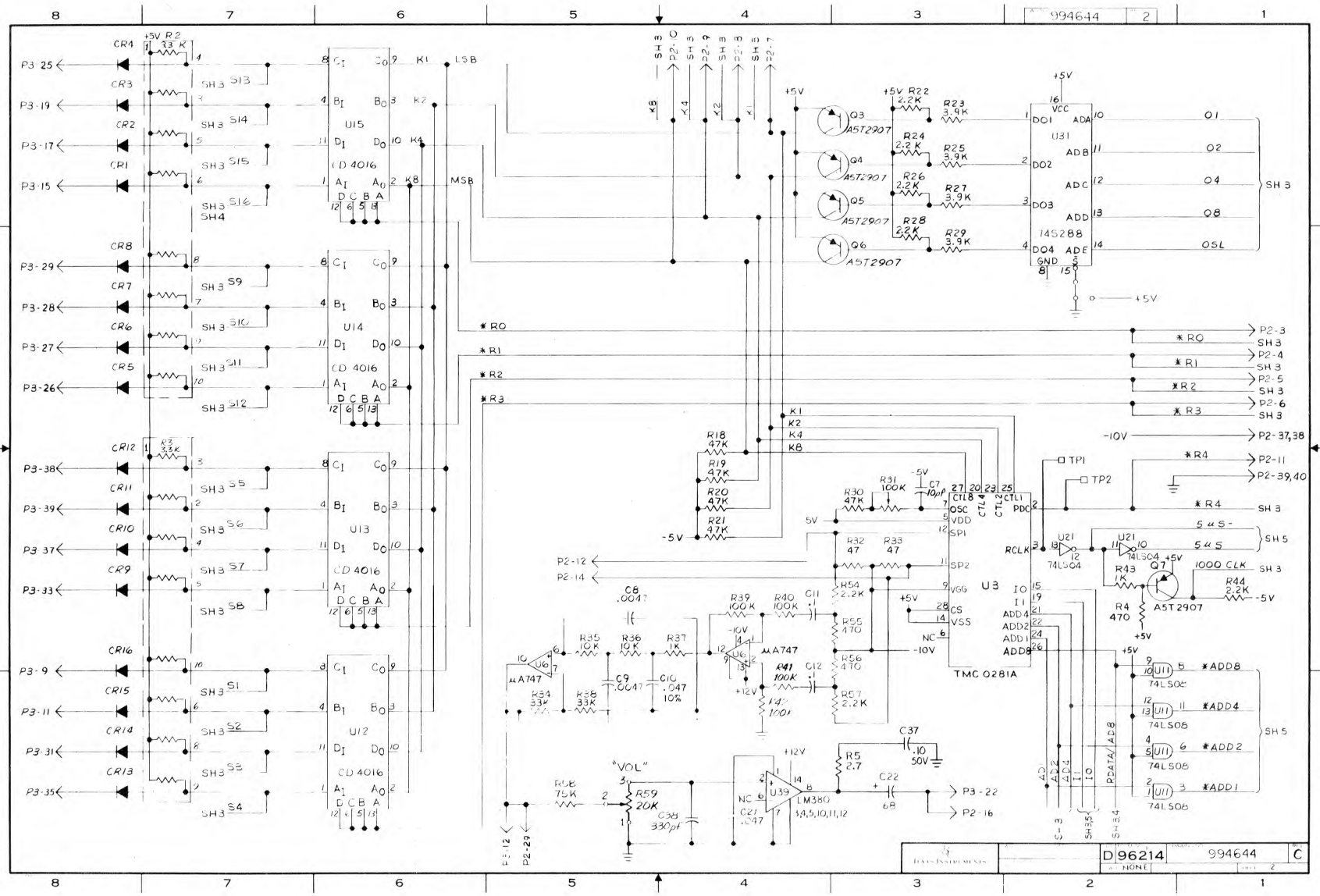
SIZE D CODE IDENT NO 96214 DRAWING NO 994644

SCALE NONE SHEET 1 OF 5

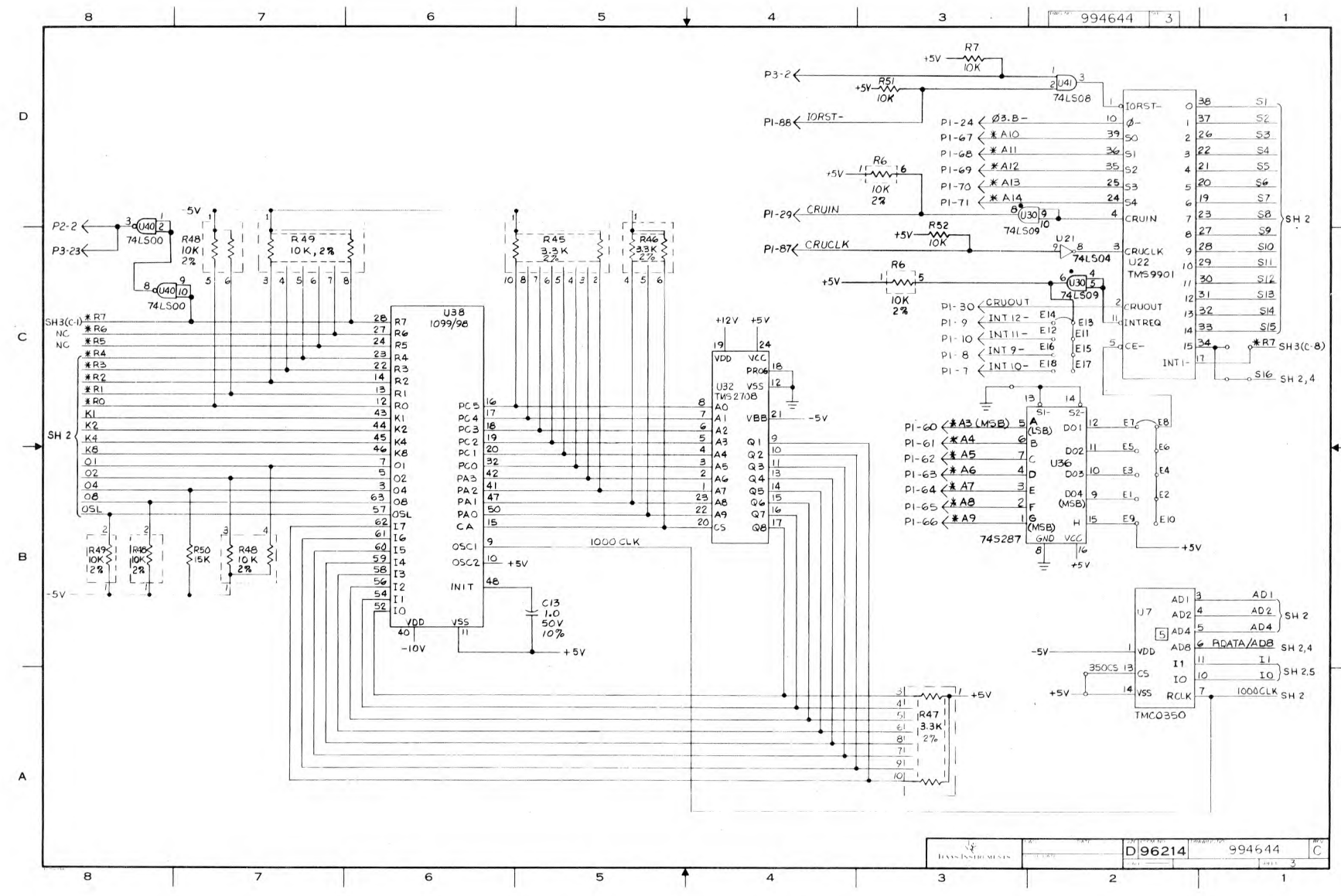
SEQ NO	IDENT	F SPEC	NO	ADDITIONAL	NOTES
	PROCESS			CLASSIFICATION	

PROCESSES — FOR CORRELATION TO GOVT/IND SPECIFICATIONS, SEE TI DRAWING 729467

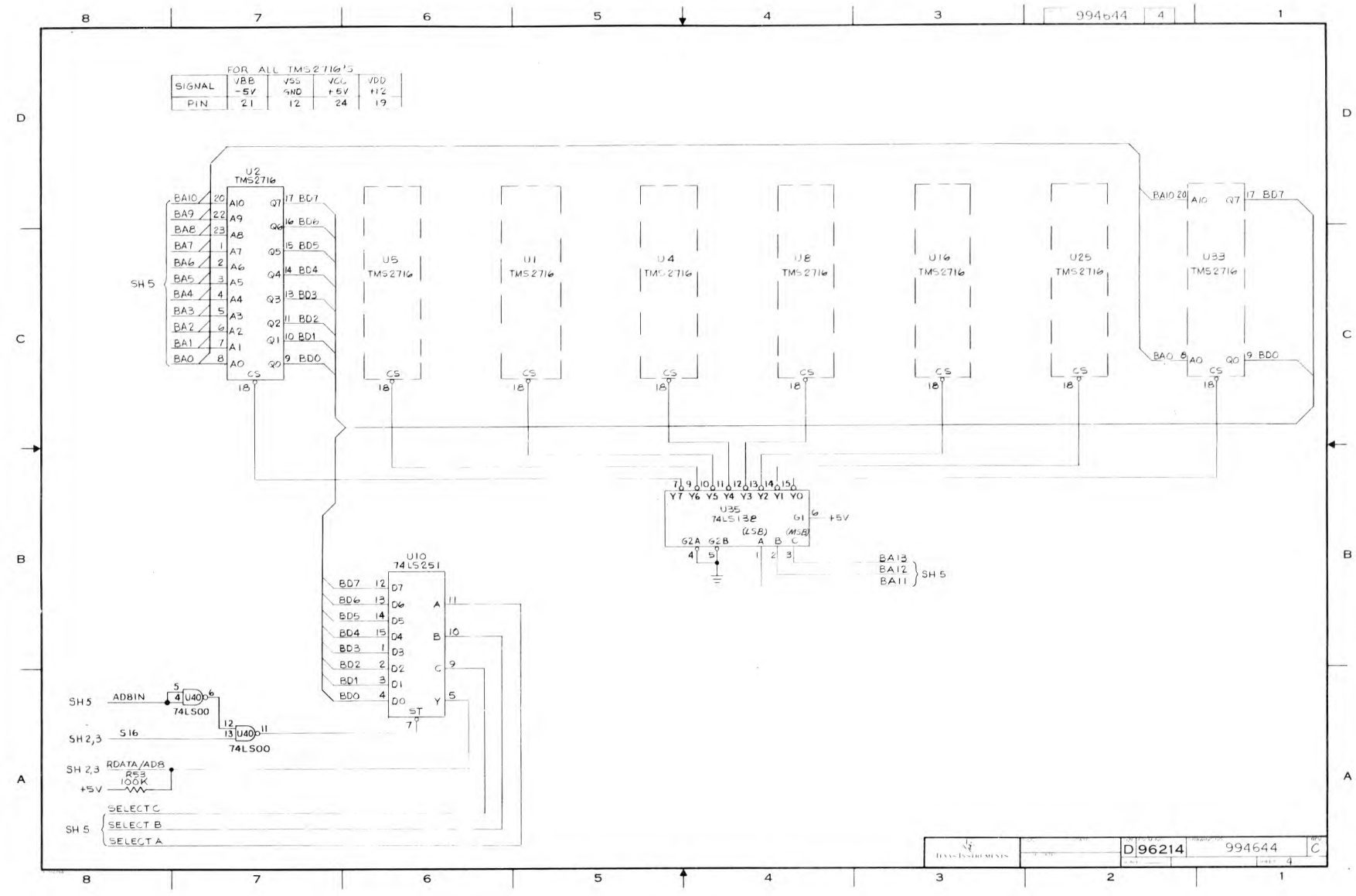
43

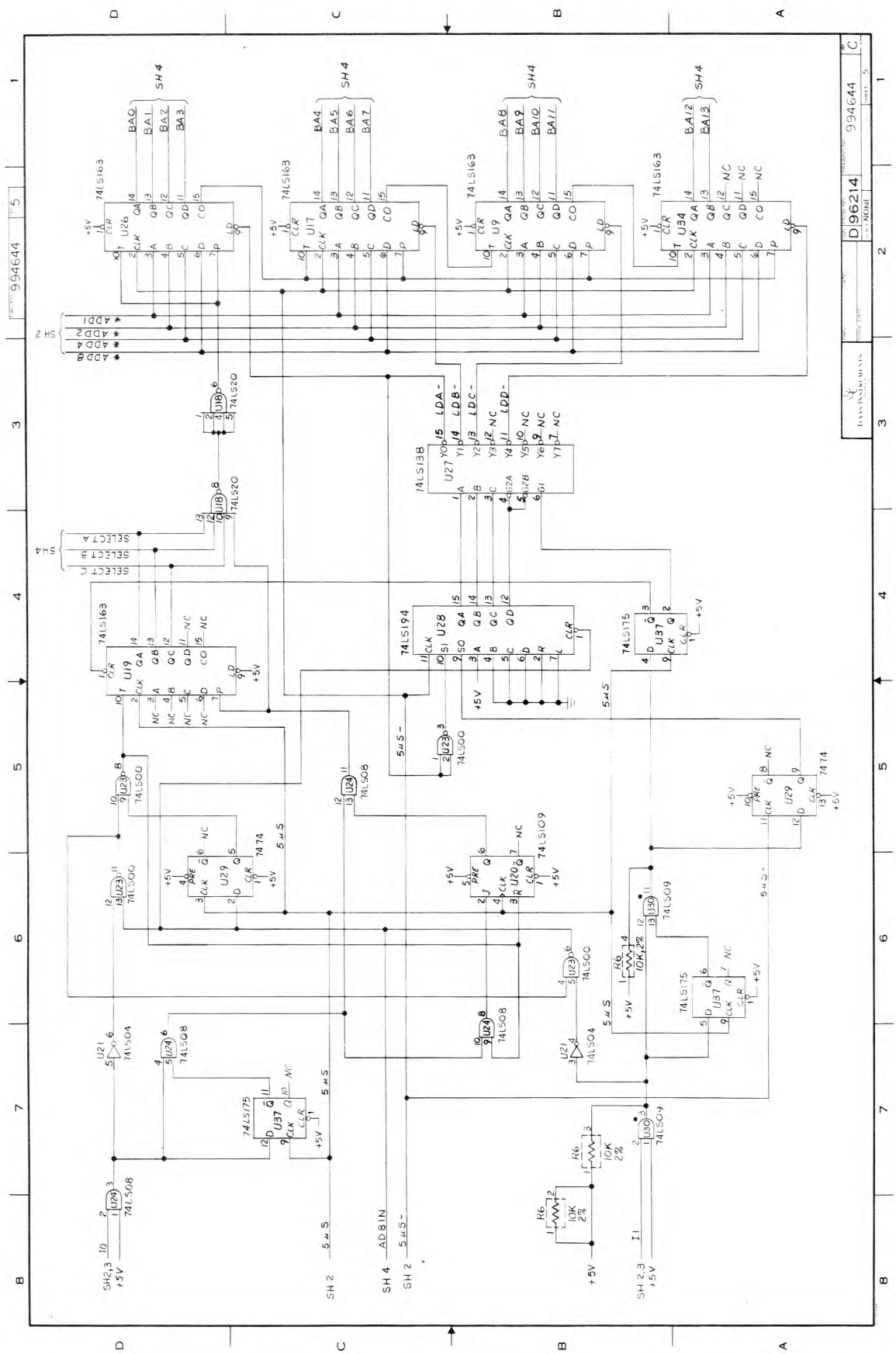

994644 2
CR4
CR3
CR2
CR1
CR8
CR7
CR6
CR5
CR12
CR11
CR10
CR9
CR16
CR15
CR14
CR13
P3-25
P3-19
P3-17
P3-15
P3-29
P3-28
P3-27
P3-26
P3-38
P3-39
P3-37
P3-33
P3-9
P3-11
P3-31
P3-35
+5V R2 33K
R3 33K
U15
U14
U13
U12
CD 4016
K1
K2
K4
K8
LSB
MSB
*R0
*R1
*R2
*R3
Q3 A5T2907
Q4 A5T2907
Q5 A5T2907
Q6 A5T2907
R22 2.2K
R23 3.9K
R24 2.2K
R25 3.9K
R26 2.2K
R27 3.9K
R28 2.2K
R29 3.9K
U31
74S288
O1
O2
O4
O8
OSL
SH 3
R18 47K
R19 47K
R20 47K
R21 47K
-5V
+5V
TMC 0281A
U3
R30 47K
R31 100K
C7 10pf
R32 47
R33 47
R54 2.2K
R55 470
R56 470
R57 2.2K
R39 100K
R40 100K
C11
R41 100K
C12
C8 .0047
C9 .0047
C10 .047 10%
R35 10K
R36 10K
R37 1K
R34 33K
R38 33K
μA747
R58 75K
"VOL"
R59 20K
C38 330pf
C37 .10 50V
R5 2.7
C22 68
C21 .047
U39 LM380
1,4,5,10,11,12
P3-22
P2-16
P2-12
P2-14
P2-29
TP1
TP2
U21 74LS04
R43 1K
Q7 A5T2907
1000 CLK
R44 2.2K
R4 470
*R4
5μS
ADD8
ADD4
ADD2
ADD1
74LS08
DATA/ADD
P2-3
P2-4
P2-5
P2-6
P2-37,38
P2-11
P2-39,40
D 96214
994644
C

A-2


994644
3
R7
10K
+5V
P3-2
R51
10K
P1-88
IORST-
U41
74LS08
IORST-
Ø-
S0
S1
S2
S3
S4
CRUIN
CRUCLK
U22
TMS9901
CRUOUT
INTREQ
CE-
INT-
P1-24
Ø3.B-
P1-67
*A10
P1-68
*A11
P1-69
*A12
P1-70
*A13
P1-71
*A14
R6
10K
2%
P1-29
CRUIN
U30
74LS09
R52
10K
P1-87
CRUCLK
U21
74LS04
P1-30
CRUOUT
P1-9
INT12-
P1-10
INT11-
P1-8
INT9-
P1-7
INT10-
SH 2
SH3(C-8)
S16
SH 2,4
P2-2
P3-23
U40
74LS00
R48
10K
2%
R49
10K,2%
R45
3.3K
2%
R46
3.3K
2%
-5V
U38
1099/98
SH3(C-1)
NC
NC
SH 2
*R7
*R6
*R5
*R4
*R3
*R2
*R1
*R0
K1
K2
K4
K8
O1
O2
O4
O8
OSL
I7
I6
I5
I4
I3
I2
I1
I0
PC5
PC4
PC3
PC2
PC1
PC0
PA3
PA2
PA1
PA0
CA
OSC1
OSC2
INIT
VDD
VSS
-10V
+5V
1000 CLK
C13
1.0
50V
10%
U32
TMS2708
+12V
VCC
PROG
VBB
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
CS
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
R49
10K
2%
R48
10K
2%
R50
15K
R48
10K
2%
R47
3.3K
2%
P1-60
*A3 (MSB)
P1-61
*A4
P1-62
*A5
P1-63
*A6
P1-64
*A7
P1-65
*A8
P1-66
*A9
U36
745287
SI-
SO-
DO1
DO2
DO3
DO4
(MSB)
GND
VCC
E1
E5
E3
E1
E9
E8
E6
E4
E2
E10
U7
TMC0350
350CS
VDD
CS
VSS
AD1
AD2
AD4
AD8
I1
IO
RCLK
RDATA/AD8
1000CLK
SH 2
SH 2,4
SH 2,5
D96214
994644
C

A-3


FOR ALL TMS2716'S
SIGNAL | VBB | VSS | VCC | VDD
PIN | -5V | GND | +5V | +12
21 | 12 | 24 | 19
U2 TMS2716
U5 TMS2716
U1 TMS2716
U4 TMS2716
U8 TMS2716
U16 TMS2716
U25 TMS2716
U33 TMS2716
BA10 BA9 BA8 BA7 BA6 BA5 BA4 BA3 BA2 BA1 BA0
SH 5
A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0
BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
CS
U35 74LS138
Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0
G1 +5V
G2A G2B A B C (LSB) (MSB)
BA13 BA12 BA11 SH 5
U10 74LS251
D7 D6 D5 D4 D3 D2 D1 D0 ST
A B C Y
SH 5 ADBIN
U40 74LS00
SH 2,3 S16
SH 2,3 RDATA/ADB
R53 100K +5V
SH 5 SELECT C SELECT B SELECT A
TEXAS INSTRUMENTS
D 96214 994644 C
SHEET 4



APPENDIX B

DEMONSTRATION SOFTWARE

B.1 GENERAL

The TM 990/306 Demo Software Provides the capability to utilize the TM 990/306 Speech Module. Such operations as saying the entire vocabulary of the demonstration software, saying pre-defined phases, and creating and saying user defined phases are all provided. Extensive prompting and a help listing are provided to make the software easy to use for anyone.

This version of the TM 990/306 Demo Software resides at memory address 1000_{16}. The two TMS 2716 Demonstration Software EPROMS must be placed as marked in sockets U43 and U45 of either a TM 990/100 or TM 990/101 CPU module.

The following jumpers must be set on a TM 990/100 CPU module:

- J2 to 2716
- J3 to 08 — The TIBUG monitor is in TMS 2708 EPROM
- J4 to 16 — The demonstration software is in TMS 2716 EPROM
- J7 to EIA

The following jumpers must be set on a TM 990/101 CPU module:

- E31-E32, E33-E34 — Bank 2 = 2716
- E9-E10 — 2716 memory map
- E13-E14 — EPROM on
- E16-E17 — RAM high
- E39-E40 — EIA
- If TIBUG is in TMS 2708 EPROM, jumpers must appear at E27-E28 and E29-E30 indicating Bank 1 = TMS 2708s

The following jumper must be set on the TM 990/306 Speech Module:

- E7-E8 — CRU software base address

The TIBUG monitor is required to perform the I/O to the terminal. If using a TM 990/101 or TM 990/100 CPU, the terminal must be connected to port P2. This firmware uses the polled mode of word completion; thus, either TIBUG (TM 990/401-1 or TM 990/401-3) can be used.

The following commands are provided by this demonstration software:

A Speak all the Speech Module words
C Create phrases by the operator
H Help (print out this list of user commands)
I Instructions for creating phrases by the user
Q Quit and return to the TIBUG monitor
R Repeat all the words rapidly
W Print out all the words

There are also six prepared phrases, numbered 1 to 6, which the user can execute by entering the phrase number. Three other numbers -- 7, 8, and 9 -- are reserved for phrases the user can build himself.

B.2 EXAMPLE OF EXECUTING THE DEMONSTRATION SOFTWARE

The following are entered after bringing up the TIBUG monitor via the RESET switch on the CPU board and entering a carriage return at the terminal:

```
TIBUG REV.A

?R
W=FFB0
P=FE00  1000
S=000C  0
?E
TM990/306 DEMO SOFTWARE      RELEASE 1.2  12/10/79
```

The first command is an H (help) which displays the user commands:

```
?? H
COMMANDS:

H - HELP, PRINTS THIS HELP LIST
Q - QUIT, BACK TO TIBUG
A - SPEAK ALL THE WORDS SLOWLY
R - REPEAT ALL THE WORDS RAPIDLY
I - INSTRUCTIONS FOR CREATING PHRASES
W - WORD LIST
C - CREATE PHRASES
1 - "HI! I AM THE TM990 306!"
2 - "TIME TO INITIALIZE DEVICE FOUR"
3 - "PUT THE FIRE OUT"
4 - "THE PRESSURE IS TOO HIGH ON CYCLE ONE"
5 - "CONTROL FIVE IS OUT OF RANGE"
6 - "DANGER! SWITCH FIVE IS SET TO CALIBRATE"
7 - USER CREATED PHRASE
8 - USER CREATED PHRASE
9 - USER CREATED PHRASE
<CR> - CARRIAGE RETURN TERMINATES INSTRUCTION
```

The user enters commands to the ?? prompt:

```
?? A  <----------Speak all the words in the demonstration software

?? 1  <----------Speak phrase no. 1

?? 2  <----------Speak phrase no. 2

?? Q  <----------Quit, go back to the TIBUG monitor
?
      <----------TIBUG command prompt
```

The user enters the I command to get instructions to create phrases:

```
?? I
CREATE PHRASES BY USING THE "C" COMMAND
ENTER WORD FROM WORD LIST FOLLOWED BY CARRIAGE RETURN
THE PHRASE MUST BE 16 WORDS OR LESS
ENTER "CONTROL C" TO SIGNAL END OF PHRASE
ENTER PHRASE NUMBER 7, 8, OR 9 THEN CARRIAGE RETURN
LISTEN TO PHRASE BY ENTERING THE PHRASE NUMBER
```

The user creates a phrase using the C command:

```
?? C
WORD?  TWO
WORD?  AND
WORD?  THREE
WORD?  EQUAL
WORD?  FIVE
WORD?  <----------CNTRL C entered at keyboard (hold down CNTRL, press C)
PHRASE NUMBER? 8<--Designate number for phrase (number 8)
?? 8 <------------Execute phrase

??
```

```
          SDSMAC 3.2.0 78.274   16:38:05 MONDAY, DEC 10, 1979.
ACCESS NAMES TABLE                                          PAGE 0001
SOURCE ACCESS NAME=      ENGR.GIBBLE.SPEECH.SRC.DEMO3E
OBJECT ACCESS NAME=      ENGR.GIBBLE.SPEECH.OBJ.DEMO3E
LISTING ACCESS NAME=     ENGR.GIBBLE.SPEECH.LST.DEMO3E
ERROR ACCESS NAME=       ENGR.GIBBLE.SPEECH.ERR.DEMO3E
OPTIONS=                 XREF,TUNLST
MACRO LIBRARY PATHNAME=
```

```
0002                    IDT  'DEMO3E'
0003                *
0004                ****************************************************************
0005                *                                                              *
0006                *     TITLE:          TM990/306 DEMO SOFTWARE                  *
0007                *                                                              *
0008                *     DATE:           12/10/79                                 *
0009                *                                                              *
0010                *     FUNCTION:       TO DEMONSTRATE THE FEATURES AND          *
0011                *                     OPERATION OF THE SPEECH BOARD.           *
0012                *                                                              *
0013                *     DESCRIPTION:    THIS PROGRAM CONSISTS OF A COMMAND       *
0014                *                     SCANNER, MANY HELP LISTS AND PROMPTS,    *
0015                *                     AND A COLLECTION OF SERVICE ROUTINES.    *
0016                *                     THIS VERSION IS NOT EPROM-RELOCATABLE,   *
0017                *                     AND THE EPROMS (2 2716'S) MUST BE        *
0018                *                     PROGRAMMED TO RESIDE AT A FIXED          *
0019                *                     ADDRESS (USUALLY  1000).                 *
0020                *                     RAM IS REQUIRED BETWEEN  FE00 AND  FFFF  *
0021                *                     THE /306 BOARD MUST RESIDE AT CRU BASE   *
0022                *                      1FC0 BY FIXING JUMPER E7-E8.            *
0023                *                                                              *
0024                *     SOFTWARE:       TIBUG XOP'S ARE REQUIRED.  REV.          *
0025                *                     FIRMWARE FOR THE CRU DECODE, OPLA        *
0026                *                     DECODE AND SE1 CONTROL IS SUFFICIENT.    *
0027                *                     IN ORDER TO INCLUDE THE DELAY WORDS      *
0028                *                     THE RELEASE 1.2 DATA SET MUST BE USED.   *
0029                *                                                              *
0030                *     AUTHOR:         DALE R. GIBBLE                           *
0031                *                                                              *
0032                ****************************************************************
0033                *
0034     FE00  LBUF   EQU  >FE00
0035     FEE0  END7   EQU  >FEE0
0036     FEE2  END8   EQU  >FEE2
0037     FEE4  END9   EQU  >FEE4
0038     FEE6  INCFLG EQU  >FEE6          FLAG TO SKIP TEXT ON ALL CMD
0039     FE50  PH1    EQU  >FE50          PHRASE 1 BUFFER
0040     FE70  PH2    EQU  >FE70          "CREATE" BUFFER 1
0041     FE90  PH3    EQU  >FE90          "CREATE" BUFFER 2
0042     FEB0  PH4    EQU  >FEB0          "CREATE" BUFFER 3
0043     FED0  BUFFER EQU  >FED0          "CREATE" ASCII INPUT BUFFER
0044     FF30  WPSUBR EQU  >FF30
0045     FF50  WPMAIN EQU  >FF50
0046     FF70  LOCKBF EQU  >FF70          LOCK BUFFER FOR PHRAS7, 8, 9
0047     FF90  WPTOP  EQU  >FF90
0048     0300  CONTC  EQU  >0300          ASCII CONTROL C
0049     0D00  CARRET EQU  >0D00          ASCII CARRIAGE RETURN
0050                ***************************
0051                      DXOP HEXI,9
0052                      DXOP HEXO,10
0053                      DXOP READ,11
0054                      DXOP PRNT,14
```

```
 0056 0000 0460          B    @START     GO TO PROCEDURE DIVISION
      0002 0922'
 0057              ***************************************************************
 0058              *                                                             *
 0059              *  THIS TABLE CONTAINS ASCII REPRESENTATION OF THE WORDS      *
 0060              *  WHICH CAN BE SPOKEN BY THE SPEECH MODULE.  THE "-"         *
 0061              *  BEFORE THE ASCII TEXT CAUSES THE ASSEMBLER TO CREATE A     *
 0062              *  TWO'S COMPLEMENT OF THE LAST CHARACTER IN THE STRING.      *
 0063              *  THE LISTING PROGRAM USES THIS INFORMATION TO FIND THE      *
 0064              *  END OF THE STRING.                                         *
 0065              *                                                             *
 0066              ***************************************************************
 0067              *
 0068 0004   BF   W001    TEXT -'A'
 0069 0005   BE           TEXT -'B'
 0070 0006   BD           TEXT -'C'
 0071 0007   BC           TEXT -'D'
 0072 0008   BB           TEXT -'E'
 0073 0009   BA           TEXT -'F'
 0074 000A   B9           TEXT -'G'
 0075 000B   B8           TEXT -'H'
 0076 000C   B7           TEXT -'I'
 0077 000D   B6           TEXT -'J'
 0078 000E   B5           TEXT -'K'
 0079 000F   B4           TEXT -'L'
 0080 0010   B3           TEXT -'M'
 0081 0011   B2           TEXT -'N'
 0082 0012   B1           TEXT -'O'
 0083 0013   B0           TEXT -'P'
 0084 0014   AF           TEXT -'Q'
 0085 0015   AE           TEXT -'R'
 0086 0016   AD           TEXT -'S'
 0087 0017   AC           TEXT -'T'
 0088 0018   AB           TEXT -'U'
 0089 0019   AA           TEXT -'V'
 0090 001A   A9           TEXT -'W'
 0091 001B   A8           TEXT -'X'
 0092 001C   A7           TEXT -'Y'
 0093 001D   A6           TEXT -'Z'
 0094 001E   41           TEXT -'ABORT'
 0095 0023   41           TEXT -'ALERT'
 0096 0028   41           TEXT -'ALL'
 0097 002B   41           TEXT -'AMPS'
 0098 002F   41           TEXT -'AND'
 0099 0032   41           TEXT -'ADJUST'
 0100 0038   41           TEXT -'AUTOMATIC'
 0101 0041   42           TEXT -'BACK'
 0102 0045   42           TEXT -'BUTTON'
 0103 004B   43           TEXT -'CALIBRATE'
 0104 0054   43           TEXT -'CALL'
 0105 0058   43   W038    TEXT -'CANCEL'
 0106 005E   43           TEXT -'CARPENTER'
 0107 0067   43           TEXT -'CHECK'
 0108 006C   43           TEXT -'CLOCK'
```

```
0109 0071   43          TEXT -'CLOSE'
0110 0076   43          TEXT -'CONTROL'
0111 007D   43          TEXT -'CRANE'
0112 0082   43          TEXT -'CREASE'
0113 0088   43          TEXT -'CYCLE'
0114 008D   44          TEXT -'DANGER'
0115 0093   44          TEXT -'DAYS'
0116 0097   44          TEXT -'DEGREES'
0117 009E   44          TEXT -'DELAY4'
0118 00A4   44          TEXT -'DELAY3'
0119 00AA   44          TEXT -'DELAY2'
0120 00B0   44          TEXT -'DELAY1'
0121 00B6   44          TEXT -'DEVICE'
0122 00BC   44          TEXT -'DISPLAY'
0123 00C3   44          TEXT -'DOOR'
0124 00C7   44          TEXT -'DOWN'
0125 00CB   44          TEXT -'DIRECTION'
0126 00D4   45          TEXT -'EIGHT'
0127 00D9   45          TEXT -'EAST'
0128 00DD   45          TEXT -'ELECTRICIAN'
0129 00E8   45          TEXT -'ELEVEN'
0130 00EE   45          TEXT -'ENTER'
0131 00F3   45          TEXT -'EQUAL'
0132 00F8   45          TEXT -'EXIT'
0133 00FC   46          TEXT -'FAIL'
0134 0100   46          TEXT -'FARAD'
0135 0105   46          TEXT -'FAST'
0136 0109   46          TEXT -'FEET'
0137 010D   46          TEXT -'FIRE'
0138 0111   46          TEXT -'FIVE'
0139 0115   46          TEXT -'FLOW'
0140 0119   46          TEXT -'FREQUENCY'
0141 0122   46          TEXT -'FROM'
0142 0126   46          TEXT -'FOREMAN'
0143 012D   46          TEXT -'FOUR'
0144 0131   47          TEXT -'GUAGE'
0145 0136   47          TEXT -'GATE'
0146 013A   47          TEXT -'GET'
0147 013D   47          TEXT -'GO'
0148 013F   47          TEXT -'GREEN'
0149 0144   48          TEXT -'HENRY'
0150 0149   48          TEXT -'HERTZ'
0151 014E   48          TEXT -'HIGH'
0152 0152   48          TEXT -'HOLD'
0153 0156   48          TEXT -'HOURS'
0154 015B   48          TEXT -'HUNDRED'
0155 0162   49          TEXT -'INCH'
0156 0166   49          TEXT -'INITIALIZE'
0157 0170   49          TEXT -'INSPECTOR'
0158 0179   49          TEXT -'INTRUDER'
0159 0181   49          TEXT -'IS'
0160 0183   4A          TEXT -'JOG'
0161 0186   4C          TEXT -'LEFT'
0162 018A   4C          TEXT -'LIGHT'
```

```
0163 018F   4C          TEXT -'LINE'
0164 0193   4C          TEXT -'LOW'
0165 0196   4D          TEXT -'MACHINE'
0166 019D   4D          TEXT -'MAKER'
0167 01A2   4D          TEXT -'MANUAL'
0168 01A8   4D          TEXT -'MEASURE'
0169 01AF   4D          TEXT -'MEGA'
0170 01B3   4D          TEXT -'METER'
0171 01B8   4D          TEXT -'MICRO'
0172 01BD   4D          TEXT -'MILL'
0173 01C1   4D          TEXT -'MILLI'
0174 01C6   4D          TEXT -'MINUS'
0175 01CB   4D          TEXT -'MINUTES'
0176 01D2   4D          TEXT -'MOTOR'
0177 01D7   4D          TEXT -'MOVE'
0178 01DB   4E          TEXT -'NINE'
0179 01DF   4E          TEXT -'NORTH'
0180 01E4   4E          TEXT -'NUMBER'
0181 01EA   4F          TEXT -'OF'
0182 01EC   4F          TEXT -'OFF'
0183 01EF   4F          TEXT -'OHMS'
0184 01F3   4F          TEXT -'OPEN'
0185 01F7   4F          TEXT -'OPERATOR'
0186 01FF   4F          TEXT -'ON'
0187 0201   4F          TEXT -'OUT'
0188 0204   4F          TEXT -'OVER'
0189 0208   4F          TEXT -'ONE'
0190 020B   50          TEXT -'PASS'
0191 020F   50          TEXT -'PASSED'
0192 0215   50          TEXT -'PERCENT'
0193 021C   50          TEXT -'PICO'
0194 0220   50          TEXT -'PLUS'
0195 0224   50          TEXT -'POINT'
0196 0229   50          TEXT -'POSITION'
0197 0231   50          TEXT -'POUND'
0198 0236   50          TEXT -'PRESS'
0199 023B   50          TEXT -'PRESSURE'
0200 0243   50          TEXT -'PRIORITY'
0201 024B   50          TEXT -'PROBE'
0202 0250   50          TEXT -'PROCESSING'
0203 025A   50          TEXT -'POWER'
0204 025F   50          TEXT -'PULL'
0205 0263   50          TEXT -'PUSH'
0206 0267   50          TEXT -'PUT'
0207 026A   52          TEXT -'RANGE'
0208 026F   52          TEXT -'READY'
0209 0274   52          TEXT -'RED'
0210 0277   52          TEXT -'REPAIR'
0211 027D   52          TEXT -'REPEAT'
0212 0283   52          TEXT -'REPLACE'
0213 028A   52          TEXT -'RIGHT'
0214 028F   53          TEXT -'SAFE'
0215 0293   53          TEXT -'SECONDS'
0216 029A   53          TEXT -'SET'
```

```
0217 029D   53          TEXT -'SEVEN'
0218 02A2   53          TEXT -'SHUT'
0219 02A6   53          TEXT -'SLOW'
0220 02AA   53          TEXT -'SPEED'
0221 02AF   53          TEXT -'SIX'
0222 02B2   53          TEXT -'START'
0223 02B7   53          TEXT -'SMOKE'
0224 02BC   53          TEXT -'SOUTH'
0225 02C1   53          TEXT -'STOP'
0226 02C5   53          TEXT -'SWITCH'
0227 02CB   54          TEXT -'TEST'
0228 02CF   54          TEXT -'TEN'
0229 02D2   54          TEXT -'TEMPERATURE'
0230 02DD   54          TEXT -'THOUSAND'
0231 02E5   54          TEXT -'THE'
0232 02E8   54          TEXT -'THREE'
0233 02ED   54          TEXT -'TIME'
0234 02F1   54          TEXT -'TOOL'
0235 02F5   54          TEXT -'TWELVE'
0236 02FB   54          TEXT -'TWO'
0237 02FE   54          TEXT -'TURN'
0238 0302   54          TEXT -'TIMER'
0239 0307   55          TEXT -'UNDER'
0240 030C   55          TEXT -'UNIT'
0241 0310   55          TEXT -'UP'
0242 0312   56          TEXT -'VALVE'
0243 0317   56          TEXT -'VOLTS'
0244 031C   57          TEXT -'WAIT'
0245 0320   57          TEXT -'WARD'
0246 0324   57          TEXT -'WATTS'
0247 0329   57          TEXT -'WELDER'
0248 032F   57          TEXT -'WEST'
0249 0333   59          TEXT -'YELLOW'
0250 0339   5A          TEXT -'ZERO'
0251 033D   E0          TEXT -' '
0252 033E   E0          TEXT -' '           LAST WORD FOR WORD LIST
```

```
0254                ****************************************************************
0255                *                                                              *
0256                *  THIS TABLE CONTAINS THE FOURTEEN BIT ADDRESSES TO BE        *
0257                *  SPOKEN BY THE SPEECH MODULE.  THESE ADDRESSES ARE           *
0258                *  POSITIONALLY IN A ONE TO ONE CORRESPONDANCE WITH THE        *
0259                *  TEXT AS LISTED IN THE PREVIOUS TABLE.  THE CREATE           *
0260                *  PROGRAM FINDS THE POSITION OF A WORD ENTERED ON A           *
0261                *  TERMINAL AND SENDS THE ADDRESS LOCATED AT THAT POSITION     *
0262                *  IN THE DATA TABLE TO THE SPEECH MODULE                      *
0263                *                                                              *
0264                ****************************************************************
0265 0340 09CC  TABTOP DATA >09CC        A
0266 0342 14D0         DATA >14D0        B
0267 0344 15A0         DATA >15A0        C
0268 0346 1740         DATA >1740        D
0269 0348 1910         DATA >1910        E
0270 034A 1A9C         DATA >1A9C        F
0271 034C 1CE4         DATA >1CE4        G
0272 034E 1D14         DATA >1D14        H
0273 0350 1DC0  I      DATA >1DC0        I
0274 0352 1DF6         DATA >1DF6        J
0275 0354 305A         DATA >305A        K
0276 0356 1E42         DATA >1E42        L
0277 0358 30F0  M      DATA >30F0        M
0278 035A 05EA         DATA >05EA        N
0279 035C 297C  O      DATA >297C        O
0280 035E 20CC         DATA >20CC        P
0281 0360 2264         DATA >2264        Q
0282 0362 2B72         DATA >2B72        R
0283 0364 2C16         DATA >2C16        S
0284 0366 2404  T      DATA >2404        T
0285 0368 2D74         DATA >2D74        U
0286 036A 2DA2         DATA >2DA2        V
0287 036C 2512         DATA >2512        W
0288 036E 25AA         DATA >25AA        X
0289 0370 2E52         DATA >2E52        Y
0290 0372 25D0         DATA >25D0        Z
0291 0374 135E         DATA >135E        ABORT
0292 0376 13C6         DATA >13C6        ALERT
0293 0378 0A6A         DATA >0A6A        ALL
0294 037A 142E         DATA >142E        AMPS
0295 037C 0AA0         DATA >0AA0        AND
0296 037E 0A0A         DATA >0A0A        ADJUST
0297 0380 146E         DATA >146E        AUTOMATIC
0298 0382 150C         DATA >150C        BACK
0299 0384 1550         DATA >1550        BUTTON
0300 0386 0000  CAL    DATA >0000        CALIBRATE
0301 0388 15E4         DATA >15E4        CALL
0302 038A 1628         DATA >1628        CANCEL
0303 038C 1690         DATA >1690        CARPENTER
0304 038E 006A         DATA >006A        CHECK
0305 0390 1710         DATA >1710        CLOCK
0306 0392 00A0         DATA >00A0        CLOSE
0307 0394 010A  CONTRL DATA >010A        CONTROL
```

```
0308 0396 016E         DATA >016E        CRANE
0309 0398 0B0A         DATA >0B0A        CREASE
0310 039A 0B40  CYCLE  DATA >0B40        CYCLE
0311 039C 01D8  DANGER DATA >01D8        DANGER
0312 039E 178A         DATA >178A        DAYS
0313 03A0 17E4         DATA >17E4        DEGREES
0314 03A2 38A6         DATA >38A6        DELAY4
0315 03A4 38AE         DATA >38AE        DELAY3
0316 03A6 38B6         DATA >38B6        DELAY2
0317 03A8 38BE         DATA >38BE        DELAY1
0318 03AA 0246  DEVICE DATA >0246        DEVICE
0319 03AC 18C2         DATA >18C2        DISPLAY
0320 03AE 0B9E         DATA >0B9E        DOOR
0321 03B0 0BCC         DATA >0BCC        DOWN
0322 03B2 1844         DATA >1844        DIRECTION
0323 03B4 02AC  EIGHT  DATA >02AC        EIGHT
0324 03B6 02DC         DATA >02DC        EAST
0325 03B8 193A         DATA >193A        ELECTRICIAN
0326 03BA 19AC         DATA >19AC        ELEVEN
0327 03BC 1A12         DATA >1A12        ENTER
0328 03BE 0C26         DATA >0C26        EQUAL
0329 03C0 1A5C         DATA >1A5C        EXIT
0330 03C2 1AD0         DATA >1AD0        FAIL
0331 03C4 1B24         DATA >1B24        FARAD
0332 03C6 0C7C         DATA >0C7C        FAST
0333 03C8 0CD6         DATA >0CD6        FEET
0334 03CA 0316  FIRE   DATA >0316        FIRE
0335 03CC 03C0  FIVE   DATA >03C0        FIVE
0336 03CE 1B92         DATA >1B92        FLOW
0337 03D0 1C1E         DATA >1C1E        FREQUENCY
0338 03D2 1C8A         DATA >1C8A        FROM
0339 03D4 1BC6         DATA >1BC6        FOREMAN
0340 03D6 0372  FOUR   DATA >0372        FOUR
0341 03D8 2628         DATA >2628        GUAGE
0342 03DA 040E         DATA >040E        GATE
0343 03DC 0D20         DATA >0D20        GET
0344 03DE 0D5C         DATA >0D5C        GO
0345 03E0 2EF2         DATA >2EF2        GREEN
0346 03E2 2696         DATA >2696        HENRY
0347 03E4 2F30         DATA >2F30        HERTZ
0348 03E6 0DB0  HIGH   DATA >0DB0        HIGH
0349 03E8 1D4E         DATA >1D4E        HOLD
0350 03EA 26E0         DATA >26E0        HOURS
0351 03EC 2F7E         DATA >2F7E        HUNDRED
0352 03EE 2736         DATA >2736        INCH
0353 03F0 0E0C  INIT   DATA >0E0C        INITIALIZE
0354 03F2 044A         DATA >044A        INSPECTOR
0355 03F4 2FF8         DATA >2FF8        INTRUDER
0356 03F6 04B4  IS     DATA >04B4        IS
0357 03F8 276E         DATA >276E        JOG
0358 03FA 27EE         DATA >27EE        LEFT
0359 03FC 309E         DATA >309E        LIGHT
0360 03FE 1E8E         DATA >1E8E        LINE
0361 0400 281C         DATA >281C        LOW
```

```
0362 0402 04F0          DATA >04F0        MACHINE
0363 0404 1F0C          DATA >1F0C        MAKER
0364 0406 0576          DATA >0576        MANUAL
0365 0408 0EA0          DATA >0EA0        MEASURE
0366 040A 2870          DATA >2870        MEGA
0367 040C 312E          DATA >312E        METER
0368 040E 1F5C          DATA >1F5C        MICRO
0369 0410 317A          DATA >317A        MILL
0370 0412 28CE          DATA >28CE        MILLI
0371 0414 0F02          DATA >0F02        MINUS
0372 0416 1FD0          DATA >1FD0        MINUTES
0373 0418 0F6A          DATA >0F6A        MOTOR
0374 041A 2906          DATA >2906        MOVE
0375 041C 31C2  NINE    DATA >31C2        NINE
0376 041E 202A          DATA >202A        NORTH
0377 0420 0FC2          DATA >0FC2        NUMBER
0378 0422 0622  OF      DATA >0622        OF
0379 0424 320C          DATA >320C        OFF
0380 0426 2078          DATA >2078        OHMS
0381 0428 1032          DATA >1032        OPEN
0382 042A 29A4          DATA >29A4        OPERATOR
0383 042C 0666  ON      DATA >0666        ON
0384 042E 06C0  OUT     DATA >06C0        OUT
0385 0430 3250          DATA >3250        OVER
0386 0432 06FC  ONE     DATA >06FC        ONE
0387 0434 2A0C          DATA >2A0C        PASS
0388 0436 328C          DATA >328C        PASSED
0389 0438 20F6          DATA >20F6        PERCENT
0390 043A 2A4C          DATA >2A4C        PICO
0391 043C 32CC          DATA >32CC        PLUS
0392 043E 2146          DATA >2146        POINT
0393 0440 108E          DATA >108E        POSITION
0394 0442 2A82          DATA >2A82        POUND
0395 0444 2188          DATA >2188        PRESS
0396 0446 1116  PRESSU  DATA >1116        PRESSURE
0397 0448 2AC8          DATA >2AC8        PRIORITY
0398 044A 3346          DATA >3346        PROBE
0399 044C 21EA          DATA >21EA        PROCESSING
0400 044E 3300          DATA >3300        POWER
0401 0450 2B42          DATA >2B42        PULL
0402 0452 33A0          DATA >33A0        PUSH
0403 0454 1168  PUT     DATA >1168        PUT
0404 0456 074E  RANGE   DATA >074E        RANGE
0405 0458 33D6          DATA >33D6        READY
0406 045A 229C          DATA >229C        RED
0407 045C 2B9E          DATA >2B9E        REPAIR
0408 045E 341A          DATA >341A        REPEAT
0409 0460 07BC          DATA >07BC        REPLACE
0410 0462 22D4          DATA >22D4        RIGHT
0411 0464 2CCA          DATA >2CCA        SAFE
0412 0466 346C          DATA >346C        SECONDS
0413 0468 1192  SET     DATA >1192        SET
0414 046A 23AE  SEVEN   DATA >23AE        SEVEN
0415 046C 11C2          DATA >11C2        SHUT
```

```
0416 046E 2326          DATA >2326          SLOW
0417 0470 34D8          DATA >34D8          SPEED
0418 0472 355E  SIX     DATA >355E          SIX
0419 0474 236E          DATA >236E          START
0420 0476 2C3E          DATA >2C3E          SMOKE
0421 0478 359C          DATA >359C          SOUTH
0422 047A 2C90          DATA >2C90          STOP
0423 047C 082C  SWITCH  DATA >082C          SWITCH
0424 047E 127A          DATA >127A          TEST
0425 0480 2D06          DATA >2D06          TEN
0426 0482 1202          DATA >1202          TEMPERATURE
0427 0484 35E4          DATA >35E4          THOUSAND
0428 0486 087A  THE     DATA >087A          THE
0429 0488 249A  THREE   DATA >249A          THREE
0430 048A 08B6  TIME    DATA >08B6          TIME
0431 048C 2D3A          DATA >2D3A          TOOL
0432 048E 3662          DATA >3662          TWELVE
0433 0490 0900  TWO     DATA >0900          TWO
0434 0492 12D0          DATA >12D0          TURN
0435 0494 2454          DATA >2454          TIMER
0436 0496 36B4          DATA >36B4          UNDER
0437 0498 130C          DATA >130C          UNIT
0438 049A 24DE          DATA >24DE          UP
0439 049C 093A          DATA >093A          VALVE
0440 049E 36FA          DATA >36FA          VOLTS
0441 04A0 2DD4          DATA >2DD4          WAIT
0442 04A2 3758          DATA >3758          WARD
0443 04A4 255A          DATA >255A          WATTS
0444 04A6 2E04          DATA >2E04          WELDER
0445 04A8 37A8          DATA >37A8          WEST
0446 04AA 37EC          DATA >37EC          YELLOW
0447 04AC 2E94          DATA >2E94          ZERO
0448 04AE 3FFE          DATA >3FFE          "BLANK"
0449 04B0 3FFE          DATA >3FFE          "BLANK"
0450      04B2' TABEND  EQU  $
0451                    ********************************************************************
0452                    *SAGES  ***  MESSAGES  ***  MESSAGES  ***  MESSAGES  ***  ME
0453                    ********************************************************************
0454 04B2   2A  ERMSG1  TEXT '*** SPELLING ERROR ***'
0455 04C8   07          BYTE 7,7,0
     04C9   07
     04CA   00
0456 04CB   2A  ERMSG2  TEXT '*** ONLY 16 WORDS PLEASE ***'
0457 04E7   07          BYTE 7,7,0
     04E8   07
     04E9   00
0458 04EA   2A  ERMSG3  TEXT '*** NO PHRASE HAS BEEN CREATED ***'
0459 050C   07          BYTE 7,7,0
     050D   07
     050E   00
0460 050F   2A  ERMSG4  TEXT '*** USE ONLY 7, 8, OR 9! ***'
0461 052B   07          BYTE 7,7,0
     052C   07
     052D   00
```

```
0462 052E   57  WORD   TEXT 'WORD? '
0463 0534   00         BYTE 0
0464 0535   20  LISTHD TEXT '                              '
0465 0550   54         TEXT 'TM990/306 WORD LIST'
0466 0563   0D         BYTE >0D,>0A
     0564   0A
0467 0566 0D0A  CRLF   DATA >0D0A,0
     0568 0000
0468 056A   00         BYTE 0
0469 056B   54  BANNER TEXT 'TM990/306 DEMO SOFTWARE     '
0470 0587   52         TEXT 'RELEASE 1.2  12/10/79'
0471 059C   0D         BYTE >D,>A,>0
     059D   0A
     059E   00
0472 059F   0D  PROMPT BYTE >D,>A
     05A0   0A
0473 05A1   3F         TEXT '?? '
0474 05A4   07         BYTE >7,>0
     05A5   00
0475 05A6   50  PHRANO TEXT 'PHRASE NUMBER? '
0476 05B5   00         BYTE 0
0477 05B6   0D  HLIST  BYTE >D,>A
     05B7   0A
0478 05B8   43         TEXT 'COMMANDS:'
0479 05C1   0D         BYTE >D,>A,>A
     05C2   0A
     05C3   0A
0480 05C4   48         TEXT 'H - HELP, PRINTS THIS HELP LIST'
0481 05E3   0D         BYTE >D,>A
     05E4   0A
0482 05E5   51         TEXT 'Q - QUIT, BACK TO TIBUG'
0483 05FC   0D         BYTE >D,>A
     05FD   0A
0484 05FE   41         TEXT 'A - SPEAK ALL THE WORDS SLOWLY'
0485 061C   0D         BYTE >D,>A
     061D   0A
0486 061E   52         TEXT 'R - REPEAT ALL THE WORDS RAPIDLY'
0487 063E   0D         BYTE >D,>A
     063F   0A
0488 0640   49         TEXT 'I - INSTRUCTIONS FOR CREATING PHRASES'
0489 0665   0D         BYTE >D,>A
     0666   0A
0490 0667   57         TEXT 'W - WORD LIST'
0491 0674   0D         BYTE >D,>A
     0675   0A
0492 0676   43         TEXT 'C - CREATE PHRASES'
0493 0688   0D         BYTE >D,>A
     0689   0A
0494 068A   31         TEXT '1 - "HI! I AM THE TM990 306!"'
0495 06A7   0D         BYTE >D,>A
     06A8   0A
0496 06A9   32         TEXT '2 - "TIME TO INITIALIZE DEVICE FOUR"'
0497 06CD   0D         BYTE >D,>A
     06CE   0A
```

```
 0498 06CF   33          TEXT '3 - "PUT THE FIRE OUT"'
 0499 06E5   0D          BYTE >D,>A
      06E6   0A
 0500 06E7   34          TEXT '4 - "THE PRESSURE IS TOO HIGH ON CYCLE ONE"'
 0501 0712   0D          BYTE >D,>A
      0713   0A
 0502 0714   35          TEXT '5 - "CONTROL FIVE IS OUT OF RANGE"'
 0503 0736   0D          BYTE >D,>A
      0737   0A
 0504 0738   36          TEXT '6 - "DANGER! SWITCH FIVE IS SET TO CALIBRATE"'
 0505 0765   0D          BYTE >D,>A
      0766   0A
 0506 0767   37          TEXT '7 - USER CREATED PHRASE'
 0507 077E   0D          BYTE >D,>A
      077F   0A
 0508 0780   38          TEXT '8 - USER CREATED PHRASE'
 0509 0797   0D          BYTE >D,>A
      0798   0A
 0510 0799   39          TEXT '9 - USER CREATED PHRASE'
 0511 07B0   0D          BYTE >D,>A
      07B1   0A
 0512 07B2   3C          TEXT '¼CR  - CARRIAGE RETURN TERMINATES INSTRUCTION'
 0513 07DF   0D          BYTE >D,>A,0
      07E0   0A
      07E1   00
 0514 07E2   43  INS     TEXT 'CREATE PHRASES BY USING THE "C" COMMAND'
 0515 0809   0D          BYTE >D,>A
      080A   0A
 0516 080B   45          TEXT 'ENTER WORD FROM WORD LIST FOLLOWED BY CARRIAGE'
 0517 0839   20          TEXT '>RETURN'
 0518 0840   0D          BYTE >D,>A
      0841   0A
 0519 0842   54          TEXT 'THE PHRASE MUST BE 16 WORDS OR LESS'
 0520 0865   0D          BYTE >D,>A
      0866   0A
 0521 0867   45          TEXT 'ENTER "CONTROL C" TO SIGNAL END OF PHRASE'
 0522 0890   0D          BYTE >D,>A
      0891   0A
 0523 0892   45          TEXT 'ENTER PHRASE NUMBER 7, 8, OR 9 THEN CARRIAGE'
 0524 08BE   20          TEXT '>RETURN'
 0525 08C5   0D          BYTE >D,>A
      08C6   0A
 0526 08C7   4C          TEXT 'LISTEN TO PHRASE BY ENTERING THE PHRASE NUMBER'
 0527 08F5   0D          BYTE >D,>A,0
      08F6   0A
      08F7   00
 0528 08F8   2A  INVCMD  TEXT '*** INVALID COMMAND ***'
 0529 090F   00          BYTE >0
 0530 0910 0000  OFFSET  DATA 0,15,30,45,60,>FFFF  OFFSETS FOR LINE BUFFER
      0912 000F
      0914 001E
      0916 002D
      0918 003C
      091A FFFF
```

```
0531 091C 0007  K7      DATA 7          PHRAS7 UNLOCK CONSTANT
0532 091E 0008  K8      DATA 8          PHRAS8 UNLOCK CONSTANT
0533 0920 0009  K9      DATA 9          PHRAS9 UNLOCK CONSTANT
0534 0922       START
0535 0922 02E0          LWPI WPMAIN     LOAD W.P.
     0924 FF50
0536 0926 04E0          CLR  @LOCKBF    SET LOCK FOR PHRAS7
     0928 FF70
0537 092A 04E0          CLR  @LOCKBF+2  SET LOCK FOR PHRAS8
     092C FF72
0538 092E 04E0          CLR  @LOCKBF+4  SET LOCK FOR PHRAS9
     0930 FF74
0539 0932 2FA0          PRNT @CRLF
     0934 0566'
0540 0936 2FA0          PRNT @BANNER    OUTPUT BANNER/DATE
     0938 056B'
0541              *
0542 093A       INSTR
0543 093A 02E0          LWPI WPMAIN     GET WORKSPACE
     093C FF50
0544 093E 0720          SETO @INCFLG    RESET TEXT SKIP FLAG
     0940 FEE6
0545 0942 2FA0          PRNT @PROMPT    OUTPUT PROMPT
     0944 059F'
0546 0946 2EC5          READ R5         READ THE COMMAND CHARACTER
0547 0948 0285          CI   R5,>3100   IF IT IS A '1'= FRAS1
     094A 3100
0548 094C 1353          JEQ  PHRAS1
0549 094E 0285          CI   R5,>3200   IF IT IS A '2'= FRAS2
     0950 3200
0550 0952 1354          JEQ  PHRAS2
0551 0954 0285          CI   R5,>3300   IF IT IS A '3'= FRAS3
     0956 3300
0552 0958 1355          JEQ  PHRAS3
0553 095A 0285          CI   R5,>3400   IF IT IS A '4'= FRAS4
     095C 3400
0554 095E 1356          JEQ  PHRAS4
0555 0960 0285          CI   R5,>3500   IF IT IS A '5'= FRAS5
     0962 3500
0556 0964 1357          JEQ  PHRAS5
0557 0966 0285          CI   R5,>3600   IF IT IS A '6'= FRAS6
     0968 3600
0558 096A 1358          JEQ  PHRAS6
0559 096C 0285          CI   R5,>3700   IF IT IS A '7'= FRAS7
     096E 3700
0560 0970 1359          JEQ  PHRAS7
0561 0972 0285          CI   R5,>3800   IF IT IS A '8'= FRAS8
     0974 3800
0562 0976 135A          JEQ  PHRAS8
0563 0978 0285          CI   R5,>3900   IF IT IS A '9'= FRAS9
     097A 3900
0564 097C 135B          JEQ  PHRAS9
0565 097E 0285          CI   R5,>4100   IF IT IS A 'A'= ALL
     0980 4100
```

```
0566 0982 1326          JEQ   ALL
0567 0984 0285          CI    R5,>4300      IF IT IS A 'C'= CREATE
     0986 4300
0568 0988 1329          JEQ   CREAT
0569 098A 0285          CI    R5,>5100      IF IT IS A 'Q'= QUIT
     098C 5100
0570 098E 1319          JEQ   QUIT
0571 0990 0285          CI    R5,>4800      IF IT IS A 'H'= HELP
     0992 4800
0572 0994 130E          JEQ   HELP
0573 0996 0285          CI    R5,>4900      IF IT IS A 'I'= INST
     0998 4900
0574 099A 130E          JEQ   INST
0575 099C 0285          CI    R5,>5200      IF IT IS A 'R'= REPEAT
     099E 5200
0576 09A0 134D          JEQ   REPEAT
0577 09A2 0285          CI    R5,>5700      IF IT IS A 'W'= WORDL
     09A4 5700
0578 09A6 130F          JEQ   WORDL
0579 09A8 2FA0          PRNT  @CRLF         CARRIAGE RETURN
     09AA 0566'
0580 09AC 2FA0          PRNT  @INVCMD       OTHERWISE PRINT ERROR
     09AE 08F8'
0581 09B0 10C4          JMP   INSTR         GET NEW COMMAND
0582              *
0583 09B2         HELP
0584 09B2 2FA0          PRNT  @HLIST        OUTPUT HELP LIST
     09B4 05B6'
0585 09B6 10C1          JMP   INSTR         GET NEW COMMAND
0586              *
0587 09B8         INST
0588 09B8 2FA0          PRNT  @CRLF
     09BA 0566'
0589 09BC 2FA0          PRNT  @INS          PRINT CREATE INSTRUCTIONS
     09BE 07E2'
0590 09C0 10BC          JMP   INSTR         GET NEW COMMAND
0591              *
0592 09C2         QUIT
0593 09C2 0460          B     @>0080        EXIT, TO TIBUG !
     09C4 0080
0594              *
0595 09C6         WORDL
0596 09C6 2FA0          PRNT  @CRLF
     09C8 0566'
0597 09CA 0460          B     @LIST         PRINT WORD LIST
     09CC 0A64'
0598 09CE 10B5          JMP   INSTR         GET NEW COMMAND
0599              *
0600 09D0         ALL
0601 09D0 04E0          CLR   @INCFLG       SET TEXT SKIP FLAG
     09D2 FEE6
0602 09D4 2FA0          PRNT  @CRLF
     09D6 0566'
0603 09D8 0460          B     @ALLWRD       SAY ALL THE WORDS AND RETURN
```

```
      09DA 0A4E'
 0604              *
 0605 09DC         CREAT
 0606 09DC 2FA0            PRNT @CRLF
      09DE 0566'
 0607 09E0 02E0            LWPI WPSUBR
      09E2 FF30
 0608 09E4 0201            LI   R1,PH1        INIT ADDRESS OF DATA BUFFER
      09E6 FE50
 0609 09E8 02E0            LWPI WPMAIN        GET MAIN WORKSPACE
      09EA FF50
 0610 09EC 0207            LI   R7,-2         INITIALIZE WORD COUNTER
      09EE FFFE
 0611 09F0 0460            B    @PC2
      09F2 0C6E'
 0612              *
 0613 09F4         PHRAS1
 0614 09F4 2FA0            PRNT @CRLF
      09F6 0566'
 0615 09F8 0460            B    @FRAS1
      09FA 0AE4'
 0616 09FC         PHRAS2
 0617 09FC 2FA0            PRNT @CRLF
      09FE 0566'
 0618 0A00 0460            B    @FRAS2
      0A02 0B26'
 0619 0A04         PHRAS3
 0620 0A04 2FA0            PRNT @CRLF
      0A06 0566'
 0621 0A08 0460            B    @FRAS3
      0A0A 0B4E'
 0622 0A0C         PHRAS4
 0623 0A0C 2FA0            PRNT @CRLF
      0A0E 0566'
 0624 0A10 0460            B    @FRAS4
      0A12 0B72'
 0625 0A14         PHRAS5
 0626 0A14 2FA0            PRNT @CRLF
      0A16 0566'
 0627 0A18 0460            B    @FRAS5
      0A1A 0BA6'
 0628 0A1C         PHRAS6
 0629 0A1C 2FA0            PRNT @CRLF
      0A1E 0566'
 0630 0A20 0460            B    @FRAS6
      0A22 0BD2'
 0631 0A24         PHRAS7
 0632 0A24 2FA0            PRNT @CRLF
      0A26 0566'
 0633 0A28 0460            B    @FRAS7
      0A2A 0C02'
 0634 0A2C         PHRAS8
 0635 0A2C 2FA0            PRNT @CRLF
      0A2E 0566'
```

```
0636 0A30 0460          B     @FRAS8
     0A32 0C24'
0637 0A34         PHRAS9
0638 0A34 2FA0          PRNT  @CRLF
     0A36 0566'
0639 0A38 0460          B     @FRAS9
     0A3A 0C46'
0640              *
0641 0A3C         REPEAT
0642 0A3C 2FA0          PRNT  @CRLF
     0A3E 0566'
0643 0A40 0200          LI    R0,1
     0A42 0001
0644 0A44 C800          MOV   R0,@INCFLG   SET DELAY FLAG TO 1, REPEAT MODE
     0A46 FEE6
0645 0A48 06A0  RP1     BL    @ALLWRD      SAY ALL THE WORDS ONCE
     0A4A 0A4E'
0646 0A4C 10FD          JMP   RP1          DO IT AGAIN
0647              *
0648 0A4E 02E0  ALLWRD  LWPI  WPMAIN       MAINLINE TO SPEAK ALL THE WORDS
     0A50 FF50
0649              *
0650 0A52 020C          LI    R12,>1FE0    /306 CRU ADDRESS
     0A54 1FE0
0651 0A56 0200          LI    R0,TABTOP    ADDRESS OF TABLE
     0A58 0340'
0652 0A5A 0202          LI    R2,TABEND    LAST ADDRESS
     0A5C 04B2'
0653 0A5E 06A0  ALL1    BL    @SPEAK       SAY A WORD
     0A60 0E14'
0654 0A62 10FD          JMP   ALL1         DO ANOTHER
0655              *
0656              *  BLANK THE LINE BUFFER WITH ASCII SPACES " 2020"
0657              *
0658 0A64         LIST
0659 0A64 2FA0          PRNT  @LISTHD      WORD LIST HEADER
     0A66 0535'
0660 0A68 04E0          CLR   @WPSUBR      RESET RESTART COUNTER
     0A6A FF30
0661 0A6C 0206          LI    R6,W001      ADDRESS OF FIRST WORD
     0A6E 0004'
0662 0A70 0207          LI    R7,W038      ADDRESS OF LAST WORD
     0A72 0058'
0663 0A74 04C9          CLR   R9           RESET SKIP COUNTER
0664 0A76 0202  CLRBUF  LI    R2,>24       NUMBER OF WORDS IN BUFFER
     0A78 0024
0665 0A7A 0420          BLWP  @ENDCK       CHECK FOR TERMINATING CARRIAGE RETU
     0A7C 0E7C'
0666 0A7E 0200          LI    R0,LBUF      GET ADDRESS OF LINE BUFFER
     0A80 FE00
0667 0A82 0201          LI    R1,>2020     ASCII SPACE CODE
     0A84 2020
0668 0A86 CC01  CMOV    MOV   R1,*R0+      PUT CODE IN BUFFER
0669 0A88 0602          DEC   R2           STEP COUNTER
```

```
0670 0A8A 16FD          JNE  CMOV          LAST ONE?
0671 0A8C 0202          LI   R2,LBUF+72    GET OFFSET FOR END CODES
     0A8E FE48
0672 0A90 0201          LI   R1,CRLF       ADDRESS OF "0D,0A,00"
     0A92 0566'
0673 0A94 CCB1          MOV *R1+,*R2+      MOVE BYTES TO END OF
0674 0A96 CCB1          MOV *R1+,*R2+      LINE BUFFER FOR CRLF
0675               *
0676               *  MOVE FIVE WORDS TO LINE BUFFER
0677               *
0678 0A98 0200          LI   R0,OFFSET     ADDRESS OF OFFSET NUMBERS
     0A9A 0910'
0679 0A9C 04C3  NEWOFF  CLR  R3            RESET END OF STRING FLAG
0680 0A9E C070          MOV  *R0+,R1       GET VALUE OF OFFSET
0681 0AA0 0281          CI   R1,>FFFF      LAST WORD ON LINE?
     0AA2 FFFF
0682 0AA4 1316          JEQ  PRTLIN        IF YES, PRINT NEXT LINE
0683 0AA6 D236  NEXCHR  MOVB *R6+,R8       GET CHARACTER IN HOLD REG
0684 0AA8 0888          SRA  R8,8          PUT BYTE IN LOWER HALF FOR ABS
0685 0AAA 0748          ABS  R8            CHECK FOR LAST CHAR IN STRING
0686 0AAC 1101          JLT  TSTCHR        IF NEGATIVE, LAST CHAR
0687 0AAE 0703          SETO R3            SET FLAG
0688 0AB0 0A88  TSTCHR  SLA  R8,8          MOVE BYTE BACK TO UPPER HALF
0689 0AB2 D848          MOVB R8,@LBUF(R1)  MOVE TO LINE BUFFER
     0AB4 FE00
0690 0AB6 0581          INC  R1            STEP POSITION COUNTER
0691 0AB8 C0C3          MOV  R3,R3         LAST CHAR IN STRING?
0692 0ABA 1302          JEQ  SKIP          IF YES, SKIP 36 WORDS
0693 0ABC 04C3          CLR  R3            IF NOT, RESET FLAG
0694 0ABE 10F3          JMP  NEXCHR        MOVE NEXT CHARACTER
0695 0AC0 D236  SKIP    MOVB *R6+,R8       GET NEXT CHARACTER
0696 0AC2 0748          ABS  R8            END OF WORD?
0697 0AC4 15FD          JGT  SKIP          IF NO, CHECK NEXT CHARACTER
0698 0AC6 0589          INC  R9            STEP SKIP COUNTER
0699 0AC8 0289          CI   R9,36         NUMBER OF WORDS/COLUMN
     0ACA 0024
0700 0ACC 16F9          JNE  SKIP          IF NOT, SKIP ANOTHER
0701 0ACE 04C9          CLR  R9            RESET SKIP COUNTER
0702 0AD0 10E5          JMP  NEWOFF        IF YES, NEW OFFSET
0703               *
0704 0AD2 2FA0  PRTLIN  XOP  @LBUF,14      PRINT LINE BUFFER
     0AD4 FE00
0705 0AD6 0420          BLWP @RESTRT       RESET R6 TO FIRST CHAR, NEXT LINE
     0AD8 0E5E'
0706 0ADA 04C9          CLR  R9            RESET SKIP COUNTER
0707 0ADC 8187          C    R7,R6         IS IT THE LAST WORD?
0708 0ADE 15CB          JGT  CLRBUF        CLEAR BUFFER & GET NEW LINE
0709 0AE0 0460          B    @INSTR        GET NEXT INSTRUCTION
     0AE2 093A'
0710               *
0711               *  MODULE TO SAY PHRASE 1
0712               *
0713 0AE4 02E0  FRAS1   LWPI PH1
     0AE6 FE50
```

```
0714 0AE8 C020          MOV  @HIGH,R0      HIGH
     0AEA 03E6'
0715 0AEC C060          MOV  @I,R1         I
     0AEE 0350'
0716 0AF0 C0A0          MOV  @M,R2         M
     0AF2 0358'
0717 0AF4 C0E0          MOV  @THE,R3       THE
     0AF6 0486'
0718 0AF8 C120          MOV  @T,R4         T
     0AFA 0366'
0719 0AFC C160          MOV  @M,R5         M
     0AFE 0358'
0720 0B00 C1A0          MOV  @NINE,R6      NINE
     0B02 041C'
0721 0B04 C1C6          MOV  R6,R7         NINE
0722 0B06 C220          MOV  @T,R8         T
     0B08 0366'
0723 0B0A C260          MOV  @THREE,R9     THREE
     0B0C 0488'
0724 0B0E C2A0          MOV  @O,R10        O
     0B10 035C'
0725 0B12 C2E0          MOV  @SIX,R11      SIX
     0B14 0472'
0726 0B16 02E0          LWPI WPMAIN
     0B18 FF50
0727 0B1A 0200          LI   R0,PH1        ADDRESS OF PHRAS1 DATA
     0B1C FE50
0728 0B1E 0202          LI   R2,PH1+24     ADDRESS OF END PHRAS1
     0B20 FE68
0729 0B22 0460          B    @SAYPHR
     0B24 0C68'
0730                *
0731 0B26 02E0  FRAS2   LWPI PH1
     0B28 FE50
0732 0B2A C020          MOV  @TIME,R0      TIME
     0B2C 048A'
0733 0B2E C060          MOV  @TWO,R1       TWO  (USE AS "TO")
     0B30 0490'
0734 0B32 C0A0          MOV  @INIT,R2      INITIALIZE
     0B34 03F0'
0735 0B36 C0E0          MOV  @DEVICE,R3    DEVICE
     0B38 03AA'
0736 0B3A C120          MOV  @FOUR,R4      FOUR
     0B3C 03D6'
0737 0B3E 02E0          LWPI WPMAIN
     0B40 FF50
0738 0B42 0200          LI   R0,PH1        ADDRESS OF PHRAS2 DATA
     0B44 FE50
0739 0B46 0202          LI   R2,PH1+10     ADDRESS OF END PHRAS2
     0B48 FE5A
0740 0B4A 0460          B    @SAYPHR
     0B4C 0C68'
0741                *
0742 0B4E 02E0  FRAS3   LWPI PH1
```

```
      0B50 FE50
 0743 0B52 C020          MOV  @PUT,R0       PUT
      0B54 0454'
 0744 0B56 C060          MOV  @THE,R1       THE
      0B58 0486'
 0745 0B5A C0A0          MOV  @FIRE,R2      FIRE
      0B5C 03CA'
 0746 0B5E C0E0          MOV  @OUT,R3       OUT
      0B60 042E'
 0747 0B62 02E0          LWPI WPMAIN
      0B64 FF50
 0748 0B66 0200          LI   R0,PH1        ADDRESS OF PHRAS3 DATA
      0B68 FE50
 0749 0B6A 0202          LI   R2,PH1+8      ADDRESS OF END PHRAS3
      0B6C FE58
 0750 0B6E 0460          B    @SAYPHR
      0B70 0C68'
 0751 0B72 02E0  FRAS4   LWPI PH1
      0B74 FE50
 0752 0B76 C020          MOV  @THE,R0       THE
      0B78 0486'
 0753 0B7A C060          MOV  @PRESSU,R1    PRESSURE
      0B7C 0446'
 0754 0B7E C0A0          MOV  @IS,R2        IS
      0B80 03F6'
 0755 0B82 C0E0          MOV  @TWO,R3       TWO
      0B84 0490'
 0756 0B86 C120          MOV  @HIGH,R4      HIGH
      0B88 03E6'
 0757 0B8A C160          MOV  @ON,R5        ON
      0B8C 042C'
 0758 0B8E C1A0          MOV  @CYCLE,R6     CYCLE
      0B90 039A'
 0759 0B92 C1E0          MOV  @ONE,R7       ONE
      0B94 0432'
 0760 0B96 02E0          LWPI WPMAIN
      0B98 FF50
 0761 0B9A 0200          LI   R0,PH1        ADDRESS OF PHRAS4 DATA
      0B9C FE50
 0762 0B9E 0202          LI   R2,PH1+16     ADDRESS OF END PHRAS4
      0BA0 FE60
 0763 0BA2 0460          B    @SAYPHR
      0BA4 0C68'
 0764            *
 0765 0BA6 02E0  FRAS5   LWPI PH1
      0BA8 FE50
 0766 0BAA C020          MOV  @CONTRL,R0    CONTROL
      0BAC 0394'
 0767 0BAE C060          MOV  @FIVE,R1      FIVE
      0BB0 03CC'
 0768 0BB2 C0A0          MOV  @IS,R2        IS
      0BB4 03F6'
 0769 0BB6 C0E0          MOV  @OUT,R3       OUT
      0BB8 042E'
```

```
 0770 0BBA C120          MOV  @OF,R4        OF
      0BBC 0422'
 0771 0BBE C160          MOV  @RANGE,R5     RANGE
      0BC0 0456'
 0772 0BC2 02E0          LWPI WPMAIN
      0BC4 FF50
 0773 0BC6 0200          LI   R0,PH1        ADDRESS OF PHRAS5 DATA
      0BC8 FE50
 0774 0BCA 0202          LI   R2,PH1+12     ADDRESS OF END PHRAS5
      0BCC FE5C
 0775 0BCE 0460          B    @SAYPHR
      0BD0 0C68'
 0776              *
 0777 0BD2 02E0  FRAS6   LWPI PH1
      0BD4 FE50
 0778 0BD6 C020          MOV  @DANGER,R0    DANGER
      0BD8 039C'
 0779 0BDA C060          MOV  @SWITCH,R1    SWITCH
      0BDC 047C'
 0780 0BDE C0A0          MOV  @FIVE,R2      FIVE
      0BE0 03CC'
 0781 0BE2 C0E0          MOV  @IS,R3        IS
      0BE4 03F6'
 0782 0BE6 C120          MOV  @SET,R4       SET
      0BE8 0468'
 0783 0BEA C160          MOV  @TWO,R5       TWO
      0BEC 0490'
 0784 0BEE C1A0          MOV  @CAL,R6       CALIBRATE
      0BF0 0386'
 0785 0BF2 02E0          LWPI WPMAIN
      0BF4 FF50
 0786 0BF6 0200          LI   R0,PH1        ADDRESS OF PHRAS6 DATA
      0BF8 FE50
 0787 0BFA 0202          LI   R2,PH1+14     ADDRESS OF END PHRAS6
      0BFC FE5E
 0788 0BFE 0460          B    @SAYPHR       SAY PHRASE
      0C00 0C68'
 0789              *
 0790 0C02 02E0  FRAS7   LWPI WPMAIN
      0C04 FF50
 0791 0C06 C020          MOV  @LOCKBF,R0    GET LOCK FLAG
      0C08 FF70
 0792 0C0A 0280          CI   R0,0          IS LOCK SET?
      0C0C 0000
 0793 0C0E 1604          JNE  F7            IF NOT SET, SAY PHRAS7
 0794 0C10 2FA0          PRNT @ERMSG3       "NO PHRASE HAS BEEN CREATED"
      0C12 04EA'
 0795 0C14 0460          B    @INSTR        GET NEXT MONITOR COMMAND
      0C16 093A'
 0796 0C18 0200  F7      LI   R0,PH2        "CREATE" WORKSPACE 1
      0C1A FE70
 0797 0C1C C0A0          MOV  @END7,R2
      0C1E FEE0
 0798 0C20 0460          B    @SAYPHR
```

```
      0C22 0C68'
 0799              *
 0800 0C24 02E0    FRAS8  LWPI WPMAIN
      0C26 FF50
 0801 0C28 C020           MOV  @LOCKBF+2,R0 GET LOCK FLAG
      0C2A FF72
 0802 0C2C 0280           CI   R0,0         IS LOCK SET?
      0C2E 0000
 0803 0C30 1604           JNE  F8           IF NOT SET, SAY PHRAS8
 0804 0C32 2FA0           PRNT @ERMSG3      "NO PHRASE HAS BEEN CREATED"
      0C34 04EA'
 0805 0C36 0460           B    @INSTR       GET NEXT MONITOR COMMAND
      0C38 093A'
 0806 0C3A 0200    F8     LI   R0,PH3       "CREATE" WORKSPACE 2
      0C3C FE90
 0807 0C3E C0A0           MOV  @END8,R2     NUMBER OF BYTES IN PHRASE
      0C40 FEE2
 0808 0C42 0460           B    @SAYPHR
      0C44 0C68'
 0809              *
 0810 0C46 02E0    FRAS9  LWPI WPMAIN
      0C48 FF50
 0811 0C4A C020           MOV  @LOCKBF+4,R0 GET LOCK FLAG
      0C4C FF74
 0812 0C4E 0280           CI   R0,0         IS LOCK SET?
      0C50 0000
 0813 0C52 1604           JNE  F9           IF NOT SET, SAY PHRAS9
 0814 0C54 2FA0           PRNT @ERMSG3      "NO PHRASE HAS BEEN CREATED"
      0C56 04EA'
 0815 0C58 0460           B    @INSTR       GET NEXT MONITOR COMMAND
      0C5A 093A'
 0816 0C5C 0200    F9     LI   R0,PH4       "CREATE" WORKSPACE 3
      0C5E FEB0
 0817 0C60 C0A0           MOV  @END9,R2     NUMBER OF BYTES IN PHRASE
      0C62 FEE4
 0818 0C64 0460           B    @SAYPHR
      0C66 0C68'
 0819              *
 0820 0C68 06A0    SAYPHR BL   @SPEAK       SPEAK A WORD
      0C6A 0E14'
 0821 0C6C 10FD           JMP  SAYPHR
 0822              *
 0823              *
 0824 0C6E 020C    PC2    LI   R12,>1FE0    /306 UPPER HALF ADDRESS
      0C70 1FE0
 0825 0C72 1D0F           SBO  15           SET TO EPROM MODE
 0826 0C74 0206           LI   R6,-2        INITIALIZE WORD COUNTER
      0C76 FFFE
 0827 0C78 020A           LI   R10,PH1      ADDRESS OF "CREATE" BUFFER
      0C7A FE50
 0828 0C7C 0202    NEX    LI   R2,BUFFER    INPUT STRING BUFFER ADDRESS
      0C7E FED0
 0829 0C80 04C3           CLR  R3           RESET INPUT STRING LENGTH
 0830 0C82 04C5           CLR  R5           RESET TEST STRING LENGTH
```

```
0831 0C84 0286          CI   R6,30         16 WORDS INPUT?
     0C86 001E
0832 0C88 1607          JNE  PWORD         IF NOT, CONTINUE
0833 0C8A 2FA0          PRNT @CRLF         CARRIAGE RETURN
     0C8C 0566'
0834 0C8E 2FA0          PRNT @ERMSG2       "ONLY 16 WORDS PLEASE"
     0C90 04CB'
0835 0C92 05C6          INCT R6            BUMP WORD COUNTER
0836 0C94 0460          B    @GETNO        GET PHRASE NUMBER
     0C96 0CAA'
0837 0C98 2FA0  PWORD   PRNT @WORD         "WORD? "
     0C9A 052E'
0838 0C9C 05C6          INCT R6            STEP WORD COUNTER
0839 0C9E C806          MOV  R6,@LOCKBF+8  WORD STATUS TO UNLOCK ROUTINE
     0CA0 FF78
0840 0CA2 2EC1  READ1   READ R1            ECHO CHARACTER
0841 0CA4 0281          CI   R1,>0300      IS IT A CONTROL C?
     0CA6 0300
0842 0CA8 1666          JNE  MB            IF NOT, MOVE BYTE TO BUFFER
0843 0CAA 2FA0  GETNO   PRNT @CRLF         PRINT CARRIAGE RETURN
     0CAC 0566'
0844 0CAE 2FA0          PRNT @PHRANO       'PHRASE NUMBER?'
     0CB0 05A6'
0845 0CB2 2E40          HEXI R0            GET NUMBER
0846 0CB4 0CB8'         DATA $+4,ERR1      CORRECT/ERROR EXECUTION
     0CB6 0E08'
0847 0CB8 0280          CI   R0,7          IS IT A 7?
     0CBA 0007
0848 0CBC 1308          JEQ  GETNO7
0849 0CBE 0280          CI   R0,8          IS IT A 8?
     0CC0 0008
0850 0CC2 1321          JEQ  GETNO8
0851 0CC4 0280          CI   R0,9          IS IT A 9
     0CC6 0009
0852 0CC8 133A          JEQ  GETNO9
0853 0CCA 0460          B    @ERR1         "USE ONLY 7, 8, OR 9"
     0CCC 0E08'
0854 0CCE 0200  GETNO7  LI   R0,SEVEN      /306 ADDRESS OF SEVEN
     0CD0 046A'
0855 0CD2 06A0          BL   @SPEAK        SPEAK "SEVEN"
     0CD4 0E14'
0856 0CD6 C820          MOV  @K7,@LOCKBF+6 PHRASE7 ID TO UNLOCK ROUTINE
     0CD8 091C'
     0CDA FF76
0857 0CDC 0420          BLWP @UNLOCK       UNLOCK PHRAS7?
     0CDE 0EBA'
0858 0CE0 0200          LI   R0,PH1        GET TEMP STORAGE BUFFER
     0CE2 FE50
0859 0CE4 0201          LI   R1,PH2        GET PERM STORAGE BUFFER
     0CE6 FE70
0860 0CE8 0202          LI   R2,END7       END OF PHRASE 7
     0CEA FEE0
0861 0CEC A181          A    R1,R6         GET LAST WORD IN R6
0862 0CEE C486          MOV  R6,*R2        STORAGE FOR PHRAS7
```

```
 0863 0CF0 CC70  MOVE7  MOV  *R0+,*R1+    MOVE TO PERM STORAGE
 0864 0CF2 0280         CI   R0,PH1+32    ALL DONE?
      0CF4 FE70
 0865 0CF6 16FC         JNE  MOVE7        NO, DO ANOTHER
 0866 0CF8 C820         MOV  @K7,@LOCKBF+6 PHRASE7 ID TO UNLOCK ROUTINE
      0CFA 091C'
      0CFC FF76
 0867 0CFE 0420         BLWP @UNLOCK      UNLOCK PHRAS7?
      0D00 0EBA'
 0868 0D02 0460         B    @INSTR       GET NEXT INSTRUCTION
      0D04 093A'
 0869 0D06 0200  GETNO8 LI   R0,EIGHT     /306 ADDRESS OF EIGHT
      0D08 03B4'
 0870 0D0A 06A0         BL   @SPEAK       SPEAK "EIGHT"
      0D0C 0E14'
 0871 0D0E C820         MOV  @K8,@LOCKBF+6 PHRAS8 ID TO UNLOCK ROUTINE
      0D10 091E'
      0D12 FF76
 0872 0D14 0420         BLWP @UNLOCK      UNLOCK PHRAS8?
      0D16 0EBA'
 0873 0D18 0200         LI   R0,PH1       GET TEMP STORAGE BUFFER
      0D1A FE50
 0874 0D1C 0201         LI   R1,PH3       GET PERM STORAGE BUFFER
      0D1E FE90
 0875 0D20 0202         LI   R2,END8      END OF PHRASE 8
      0D22 FEE2
 0876 0D24 A181         A    R1,R6        ADDR OF PHRAS8 END IN R6
 0877 0D26 C486         MOV  R6,*R2       MOVE TO BUFFER
 0878 0D28 CC70  MOVE8  MOV  *R0+,*R1+    MOVE TO PERM STORAGE
 0879 0D2A 0280         CI   R0,PH1+32    ALL DONE?
      0D2C FE70
 0880 0D2E 16FC         JNE  MOVE8        NO, DO ANOTHER
 0881 0D30 C820         MOV  @K8,@LOCKBF+6 PHRAS8 ID TO UNLOCK ROUTINE
      0D32 091E'
      0D34 FF76
 0882 0D36 0420         BLWP @UNLOCK      UNLOCK PHRAS8?
      0D38 0EBA'
 0883 0D3A 0460         B    @INSTR       GET NEXT INSTRUCTION
      0D3C 093A'
 0884 0D3E 0200  GETNO9 LI   R0,NINE      /306 ADDRESS OF NINE
      0D40 041C'
 0885 0D42 06A0         BL   @SPEAK       SPEAK "NINE"
      0D44 0E14'
 0886 0D46 C820         MOV  @K9,@LOCKBF+6 PHRAS9 ID TO UNLOCK ROUTINE
      0D48 0920'
      0D4A FF76
 0887 0D4C 0420         BLWP @UNLOCK      UNLOCK PHRAS9?
      0D4E 0EBA'
 0888 0D50 0200         LI   R0,PH1       GET TEMP STORAGE BUFFER
      0D52 FE50
 0889 0D54 0201         LI   R1,PH4       GET PERM STORAGE BUFFER
      0D56 FEB0
 0890 0D58 0202         LI   R2,END9      END OF PHRASE 9
      0D5A FEE4
```

```
0891 0D5C A181          A    R1,R6          ADDR OF PHRAS9 END IN R6
0892 0D5E C486          MOV  R6,*R2         MOVE TO BUFFER
0893 0D60 CC70  MOVE9   MOV  *R0+,*R1+      MOVE TO PERM STORAGE
0894 0D62 0280          CI   R0,PH1+32      ALL DONE?
     0D64 FE70
0895 0D66 16FC          JNE  MOVE9          NO, DO ANOTHER
0896 0D68 C820          MOV  @K9,@LOCKBF+6 PHRAS9 ID TO UNLOCK ROUTNE
     0D6A 0920'
     0D6C FF76
0897 0D6E 0420          BLWP @UNLOCK        UNLOCK PHRAS9?
     0D70 0EBA'
0898 0D72 0460          B    @INSTR         GET NEXT INSTRUCTION
     0D74 093A'
0899 0D76 DC81  MB      MOVB R1,*R2+        PUT CHARACTER IN BUFFER
0900 0D78 0281          CI   R1,>0D00       IS IT A CRLF?
     0D7A 0D00
0901 0D7C 1302          JEQ  MB1            IF CRLF, COMPENSATE COUNTERS
0902 0D7E 0583          INC  R3             NO CRLF, STEP WORD COUNT
0903 0D80 1090          JMP  READ1          READ NEXT CHARACTER
0904 0D82 0207  MB1     LI   R7,W001        TEXT STRING ADDRESS
     0D84 0004'
0905 0D86 0209          LI   R9,-2          RESET DISPLACEMENT FROM W001
     0D88 FFFE
0906 0D8A 0204  T0      LI   R4,WPTOP       RESET TEXT STRING BUFF POINTER
     0D8C FF90
0907 0D8E 0202          LI   R2,BUFFER      RESET INPUT STRING BUFFER
     0D90 FED0
0908 0D92 04C5  T05     CLR  R5             RESET TEXT STRING COUNT
0909 0D94 05C9          INCT R9             DISPLCAEMENT FROM W001
0910 0D96 0289          CI   R9,>170        END OF TEXT TABLE?
     0D98 0170
0911 0D9A 1109          JLT  T1             NO, CHECK THIS TEXT STRING
0912 0D9C 2FA0          PRNT @CRLF          CARRIAGE RETURN
     0D9E 0566'
0913 0DA0 2FA0          PRNT @ERMSG1        PRINT "SPELLING ERROR"
     0DA2 04B2'
0914 0DA4 2FA0          PRNT @CRLF          CARRIAGE RETURN
     0DA6 0566'
0915 0DA8 0646          DECT R6             BACKUP WORD COUNTER
0916 0DAA 0460          B    @NEX           GET ANOTHER WORD
     0DAC 0C7C'
0917 0DAE D037  T1      MOVB *R7+,R0        GET TEXT STRING CHARACTER
0918 0DB0 0880          SRA  R0,8           MOVE TO LSB IN PREP TO ABS
0919 0DB2 0740          ABS  R0             TWO'S COMPLEMENT
0920 0DB4 1104          JLT  T2             IF LESS THAN 0, END OF STRING
0921 0DB6 0A80          SLA  R0,8           RETURN CHAR TO MSB
0922 0DB8 DD00          MOVB R0,*R4+        MOVE TO TEXT STRING TMP BUFF
0923 0DBA 0585          INC  R5             STEP TEXT STRING COUNT
0924 0DBC 10F8          JMP  T1             CONTINUE
0925 0DBE 0A80  T2      SLA  R0,8           RETURN CHAR TO MSB
0926 0DC0 DD00          MOVB R0,*R4+        MOVE TO TEXT STRING TMP BUFF
0927 0DC2 0204          LI   R4,WPTOP       RESET TEXT STRING BUFF POINTER
     0DC4 FF90
0928 0DC6 0585          INC  R5             STEP TEXT STRING COUNT
```

```
 0929 0DC8 8143          C    R3,R5          INP. STR LENGTH/TEXT STR LENGTH
 0930 0DCA 16E3          JNE  T05            NO MATCH, TRY NEXT STRING
 0931 0DCC 04C8          CLR  R8             COMPARISON BYTE COUNTER
 0932 0DCE 0204          LI   R4,WPTOP       RESET TEXT BUFF POINTER
      0DD0 FF90
 0933 0DD2 9D32  COMP    CB   *R2+,*R4+      INP STR TO TEXT STRING
 0934 0DD4 16DA          JNE  T0             NO MATCH, TRY NEXT STRING
 0935 0DD6 0588          INC  R8             STEP COMPARISON COUNT
 0936 0DD8 80C8          C    R8,R3          ALL BYTES COMPARED?
 0937 0DDA 16FB          JNE  COMP           NO, DO NEXT BYTE
 0938 0DDC 0200          LI   R0,TABTOP      ADD OF /306 WORD ADDRESSES
      0DDE 0340'
 0939 0DE0 A009          A    R9,R0          FIND /306 WORD ADDRESS
 0940 0DE2 3390          LDCR *R0,14         SEND /306 WORD ADDRESS
 0941 0DE4 1E0E          SBZ  14             SPEAK WORD
 0942 0DE6 0420          BLWP @WAIT          WAIT FOR /306 TO GO NOT BUSY
      0DE8 0DF6'
 0943 0DEA 1D0E          SBO  14             STOP SPEECH
 0944 0DEC CE90          MOV  *R0,*R10+      PUT RECORDED DATA IN BUFFER
 0945 0DEE 2FA0          PRNT @CRLF          PRINT CARRIAGE RETURN
      0DF0 0566'
 0946 0DF2 0460          B    @NEX           INPUT NEW STRING
      0DF4 0C7C'
 0947 0DF6 FF30  WAIT    DATA WPSUBR,$+2     WP,PC
      0DF8 0DFA'
 0948 0DFA 020C          LI   R12,>1FC0      /306 BASE ADDRESS
      0DFC 1FC0
 0949 0DFE 1F01  WAIT1   TB   1              TEST BUSY FLAG
 0950 0E00 16FE          JNE  WAIT1          IF BUSY, CHECK AGAIN
 0951 0E02 1F01  WAIT2   TB   1              TEST BUSY FLAG
 0952 0E04 13FE          JEQ  WAIT2          IF NOT BUSY, CHECK AGAIN
 0953 0E06 0380          RTWP
 0954 0E08 2FA0  ERR1    PRNT @CRLF
      0E0A 0566'
 0955 0E0C 2FA0          PRNT @ERMSG4
      0E0E 050F'
 0956 0E10 0460          B    @GETNO         GET ANOTHER NUMBER
      0E12 0CAA'
```

```
0958                ****************************************************************
0959                *                                                              *
0960                *                        SUBROUTINES                           *
0961                *                                                              *
0962                ****************************************************************
0963                *
0964                *  SUBROUTINE SPEAK
0965                *
0966                *       R0 IS ADDRESS OF WORD IN BUFFER BEING SPOKEN
0967                *       R2 IS ADDRESS OF THE LAST WORD TO BE SPOKEN
0968                *
0969 0E14 02E0     SPEAK  LWPI WPMAIN        GET OLD WP
     0E16 FF50
0970 0E18 C070            MOV  *R0+,R1       GET ADDRESS
0971 0E1A 020C            LI   R12,>1FE0     UPPER HALF BASE ADDRESS
     0E1C 1FE0
0972 0E1E 3381            LDCR R1,14         LOAD WORD
0973 0E20 1E0E            SBZ  14            START TALKING
0974                *  CHECK FOR TALKING
0975 0E22 0420            BLWP @WAIT         WAIT FOR INACTIVE
     0E24 0DF6'
0976 0E26 0420            BLWP @ENDCK        CHECK FOR TERMINATING CARRIAGE RETU
     0E28 0E7C'
0977 0E2A 1D0E            SBO  14            STOP TALKING
0978 0E2C 8080            C    R0,R2         CHECK FOR LAST PHARSE
0979 0E2E 160F            JNE  S4            IF NOT, RESTART
0980 0E30 02E0            LWPI WPTOP         GET RELOCATOR WORKSPACE
     0E32 FF90
0981 0E34 020C            LI   R12,>1FE0     /306 CRU BASE ADDRESS
     0E36 1FE0
0982 0E38 1D0E            SBO  14            STOP TALKING
0983 0E3A 0200            LI   R0,1
     0E3C 0001
0984 0E3E 8020            C    @INCFLG,R0    IS INCFLG A "1"?
     0E40 FEE6
0985 0E42 1302            JEQ  S3            IF YES, RETURN TO ALLWRD
0986 0E44 0460            B    @INSTR        GET NEXT INSTRUCION
     0E46 093A'
0987 0E48 020B  S3        LI   R11,ALLWRD    ADDRESS OF SPEAK ALL ROUTINE
     0E4A 0A4E'
0988 0E4C 045B            RT                 RETURN TO ALLWRD
0989 0E4E C820  S4        MOV  @INCFLG,@INCFLG
     0E50 FEE6
     0E52 FEE6
0990 0E54 1603            JNE  S5            SKIP TEXT IF FLAG IS SET
0991 0E56 0703            SETO R3            DELAY VALUE, "FFFF"
0992 0E58 0603  S41       DEC  R3            DELAY BETWEEN WORDS IN
0993 0E5A 16FE            JNE  S41           ALL WORDS MODE
0994 0E5C 045B  S5        RT
0995                *
0996                * SUBROUTINE RESTART - FINDS ALPHABETICAL START OF NEXT LINE
0997                *
0998 0E5E FF30  RESTRT DATA WPSUBR,$+2
     0E60 0E62'
```

```
 0999 0E62 0580          INC  R0             STEP WORD START COUNTER
 1000 0E64 04C3          CLR  R3             RESET TEMP SLIP COUNTER
 1001 0E66 0201          LI   R1,W001        ADDRESS OF FIRST WORD
      0E68 0004'
 1002 0E6A D0B1  RS1     MOVB *R1+,R2        GET CHARACTER
 1003 0E6C 0742          ABS  R2             END OF WORD?
 1004 0E6E 15FD          JGT  RS1            IF NOT, GET NEXT CHARACTER
 1005 0E70 0583          INC  R3             STEP SLIP COUNTER
 1006 0E72 8003          C    R3,R0          NEXT WORD FOUND?
 1007 0E74 16FA          JNE  RS1            FIND NEXT WORD
 1008 0E76 C801          MOV  R1,@WPMAIN+12 RESET MAINLNE R6 POINTER
      0E78 FF5C
 1009 0E7A 0380          RTWP
 1010              *
 1011              *  SUBROUTINE ENDCK - CHECKS FOR TERMINATING CARRIAGE RETURN
 1012              *
 1013 0E7C FF30  ENDCK   DATA WPSUBR,$+2     WP,PC
      0E7E 0E80'
 1014 0E80 020C          LI   R12,>80        SET 9902 CRU BASE ADDRESS
      0E82 0080
 1015 0E84 1F15          TB   21             IS RCV. BUFFER FULL?
 1016 0E86 1618          JNE  ESCRTN         NO ENTRY, GO BACK TO TEST
 1017 0E88 04C4          CLR  R4             PREPEARE TO STORE CHAR.
 1018 0E8A 35C4          STCR R4,7           GET 7 BIT ASCII CHAR.
 1019 0E8C 1E12          SBZ  18             RESET 9902 REV. BUFFER
 1020 0E8E 0284          CI   R4,CONTC       IS IT A CONTROL C?
      0E90 0300
 1021 0E92 1600          JNE  END1           NO, CHECK FOR CRLF
 1022 0E94 0284  END1    CI   R4,CARRET      IS IT A CARRIAGE RETURN?
      0E96 0D00
 1023 0E98 160F          JNE  ESCRTN         NO, RETURN
 1024 0E9A 02E0          LWPI WPTOP          GET RELOCATOR WORKSPACE
      0E9C FF90
 1025 0E9E 0300          LIMI 0              INT LOCKOUT FOR RESTART
      0EA0 0000
 1026 0EA2 020C          LI   R12,>1FE0      /306 CRU BASE ADDRESS
      0EA4 1FE0
 1027 0EA6 1D0E          SBO  14             STOP TALKING
 1028 0EA8 020C          LI   R12,>1FC0      LOWER HALF BASE ADDRESS
      0EAA 1FC0
 1029 0EAC 1E01          SBZ  1              ENABLE /306 INT 1
 1030 0EAE 020C          LI   R12,>80        /101 9901 CRU ADDRESS
      0EB0 0080
 1031 0EB2 1E0C          SBZ  12             ENABLE /101 INT 12
 1032 0EB4 0460          B    @INSTR         GET NEXT INSTRUCION
      0EB6 093A'
 1033 0EB8 0380  ESCRTN  RTWP                RETURN TO CALLER
 1034              *
 1035              *  SUBROUTINE UNLOCK - UNLOCKS MONITOR COMMANDS 7, 8, 9
 1036              *
 1037 0EBA FF70  UNLOCK  DATA LOCKBF,$+2
      0EBC 0EBE'
 1038 0EBE 0284          CI   R4,0           CHECK NUMBER OF WORDS INPUT
      0EC0 0000
```

```
 1039 0EC2 130D         JEQ  LK1          IF 0, DO NOT UNLOCK
 1040 0EC4 0283         CI   R3,7         UNLOCK PHRAS7?
      0EC6 0007
 1041 0EC8 1601         JNE  UN1          NO, CHECK PHRAS8
 1042 0ECA 0700         SETO R0           PHRAS7 FLAG TO UNLOCK
 1043 0ECC 0283  UN1    CI   R3,8         UNLOCK PHRAS8?
      0ECE 0008
 1044 0ED0 1601         JNE  UN2          NO, CHECK PHRAS9
 1045 0ED2 0701         SETO R1           PHRAS8 FLAG TO UNLOCK
 1046 0ED4 0283  UN2    CI   R3,9         UNLOCK PHRAS9?
      0ED6 0009
 1047 0ED8 160E         JNE  UN9          NO, RETURN
 1048 0EDA 0702         SETO R2           PHRAS9 FLAG TO UNLOCK
 1049 0EDC 100C         JMP  UN9          RETURN
 1050 0EDE 0283  LK1    CI   R3,7         LOCK PHRAS7?
      0EE0 0007
 1051 0EE2 1601         JNE  LK2          NO, CHECK PHRAS8
 1052 0EE4 04C0         CLR  R0           LOCK PHRAS7
 1053 0EE6 0283  LK2    CI   R3,8         LOCK PHRAS8?
      0EE8 0008
 1054 0EEA 1601         JNE  LK3          NO, CHECK PHRAS9
 1055 0EEC 04C1         CLR  R1           LOCK PHRAS8
 1056 0EEE 0283  LK3    CI   R3,9         LOCK PHRAS9?
      0EF0 0009
 1057 0EF2 1601         JNE  UN9          NO, RETURN
 1058 0EF4 04C2         CLR  R2           LOCK PHRAS9
 1059 0EF6 0380  UN9    RTWP
 1060      0EF8' ENDPGM EQU  $            UNPROGRAMMED EPROM FOLLOWS
 1061                   END
NO ERRORS,     NO WARNINGS
```

LABEL	VALUE	DEFN	REFERENCES
$	0EF8'		0450 0846 0947 0998 1013 1037 1060
ALL	09D0'	0600	0566
ALL1	0A5E'	0653	0654
ALLWRD	0A4E'	0648	0603 0645 0987
BANNER	056B'	0469	0540
BUFFER	FED0	0043	0828 0907
CAL	0386'	0300	0784
CARRET	0D00	0049	1022
CLRBUF	0A76'	0664	0708
CMOV	0A86'	0668	0670
COMP	0DD2'	0933	0937
CONTC	0300	0048	1020
CONTRL	0394'	0307	0766
CREAT	09DC'	0605	0568
CRLF	0566'	0467	0539 0579 0588 0596 0602 0606 0614 0617 0620 0623 0626 0629 0632 0635 0638 0642 0672 0833 0843 0912 0914 0945 0954
CYCLE	039A'	0310	0758
DANGER	039C'	0311	0778
DEVICE	03AA'	0318	0735
EIGHT	03B4'	0323	0869
END1	0E94'	1022	1021
END7	FEE0	0035	0797 0860
END8	FEE2	0036	0807 0875
END9	FEE4	0037	0817 0890
ENDCK	0E7C'	1013	0665 0976
ENDPGM	0EF8'	1060	
ERMSG1	04B2'	0454	0913
ERMSG2	04CB'	0456	0834
ERMSG3	04EA'	0458	0794 0804 0814
ERMSG4	050F'	0460	0955
ERR1	0E08'	0954	0846 0853
ESCRTN	0EB8'	1033	1016 1023
F7	0C18'	0796	0793
F8	0C3A'	0806	0803
F9	0C5C'	0816	0813
FIRE	03CA'	0334	0745
FIVE	03CC'	0335	0767 0780
FOUR	03D6'	0340	0736
FRAS1	0AE4'	0713	0615
FRAS2	0B26'	0731	0618
FRAS3	0B4E'	0742	0621
FRAS4	0B72'	0751	0624
FRAS5	0BA6'	0765	0627
FRAS6	0BD2'	0777	0630
FRAS7	0C02'	0790	0633
FRAS8	0C24'	0800	0636
FRAS9	0C46'	0810	0639
GETNO	0CAA'	0843	0836 0956
GETNO7	0CCE'	0854	0848
GETNO8	0D06'	0869	0850
GETNO9	0D3E'	0884	0852
HELP	09B2'	0583	0572
HEXI		0051	

LABEL	VALUE	DEFN	REFERENCES								
HEXO		0052									
HIGH	03E6'	0348	0714	0756							
HLIST	05B6'	0477	0584								
I	0350'	0273	0715								
INCFLG	FEE6	0038	0544	0601	0644	0984	0989	0989			
INIT	03F0'	0353	0734								
INS	07E2'	0514	0589								
INST	09B8'	0587	0574								
INSTR	093A'	0542	0581	0585	0590	0598	0709	0795	0805	0815	0868
			0883	0898	0986	1032					
INVCMD	08F8'	0528	0580								
IS	03F6'	0356	0754	0768	0781						
K7	091C'	0531	0856	0866							
K8	091E'	0532	0871	0881							
K9	0920'	0533	0886	0896							
LBUF	FE00	0034	0666	0671	0689	0704					
LIST	0A64'	0658	0597								
LISTHD	0535'	0464	0659								
LK1	0EDE'	1050	1039								
LK2	0EE6'	1053	1051								
LK3	0EEE'	1056	1054								
LOCKBF	FF70	0046	0536	0537	0538	0791	0801	0811	0839	0856	0866
			0871	0881	0886	0896	1037				
M	0358'	0277	0716	0719							
MB	0D76'	0899	0842								
MB1	0D82'	0904	0901								
MOVE7	0CF0'	0863	0865								
MOVE8	0D28'	0878	0880								
MOVE9	0D60'	0893	0895								
NEWOFF	0A9C'	0679	0702								
NEX	0C7C'	0828	0916	0946							
NEXCHR	0AA6'	0683	0694								
NINE	041C'	0375	0720	0884							
O	035C'	0279	0724								
OF	0422'	0378	0770								
OFFSET	0910'	0530	0678								
ON	042C'	0383	0757								
ONE	0432'	0386	0759								
OUT	042E'	0384	0746	0769							
PC2	0C6E'	0824	0611								
PH1	FE50	0039	0608	0713	0727	0728	0731	0738	0739	0742	0748
			0749	0751	0761	0762	0765	0773	0774	0777	0786
			0787	0827	0858	0864	0873	0879	0888	0894	
PH2	FE70	0040	0796	0859							
PH3	FE90	0041	0806	0874							
PH4	FEB0	0042	0816	0889							
PHRANO	05A6'	0475	0844								
PHRAS1	09F4'	0613	0548								
PHRAS2	09FC'	0616	0550								
PHRAS3	0A04'	0619	0552								
PHRAS4	0A0C'	0622	0554								
PHRAS5	0A14'	0625	0556								
PHRAS6	0A1C'	0628	0558								
PHRAS7	0A24'	0631	0560								

LABEL	VALUE	DEFN	REFERENCES
PHRAS8	0A2C'	0634	0562
PHRAS9	0A34'	0637	0564
PRESSU	0446'	0396	0753
PRNT		0054	
PROMPT	059F'	0472	0545
PRTLIN	0AD2'	0704	0682
PUT	0454'	0403	0743
PWORD	0C98'	0837	0832
QUIT	09C2'	0592	0570
R0	0000		0643 0644 0651 0666 0668 0678 0680 0714 0727
			0732 0738 0743 0748 0752 0761 0766 0773 0778
			0786 0791 0792 0796 0801 0802 0806 0811 0812
			0816 0845 0847 0849 0851 0854 0858 0863 0864
			0869 0873 0878 0879 0884 0888 0893 0894 0917
			0918 0919 0921 0922 0925 0926 0938 0939 0940
			0944 0970 0978 0983 0984 0999 1006 1042 1052
R1	0001		0608 0667 0668 0672 0673 0674 0680 0681 0689
			0690 0715 0733 0744 0753 0767 0779 0840 0841
			0859 0861 0863 0874 0876 0878 0889 0891 0893
			0899 0900 0970 0972 1001 1002 1008 1045 1055
R10	000A		0724 0827 0944
R11	000B		0725 0987
R12	000C		0650 0824 0948 0971 0981 1014 1026 1028 1030
R2	0002		0652 0664 0669 0671 0673 0674 0716 0728 0734
			0739 0745 0749 0754 0762 0768 0774 0780 0787
			0797 0807 0817 0828 0860 0862 0875 0877 0890
			0892 0899 0907 0933 0978 1002 1003 1048 1058
R3	0003		0679 0687 0691 0691 0693 0717 0735 0746 0755
			0769 0781 0829 0902 0929 0936 0991 0992 1000
			1005 1006 1040 1043 1046 1050 1053 1056
R4	0004		0718 0736 0756 0770 0782 0906 0922 0926 0927
			0932 0933 1017 1018 1020 1022 1038
R5	0005		0546 0547 0549 0551 0553 0555 0557 0559 0561
			0563 0565 0567 0569 0571 0573 0575 0577 0719
			0757 0771 0783 0830 0908 0923 0928 0929
R6	0006		0661 0683 0695 0707 0720 0721 0758 0784 0826
			0831 0835 0838 0839 0861 0862 0876 0877 0891
			0892 0915
R7	0007		0610 0662 0707 0721 0759 0904 0917
R8	0008		0683 0684 0685 0688 0689 0695 0696 0722 0931
			0935 0936
R9	0009		0663 0698 0699 0701 0706 0723 0905 0909 0910
			0939
RANGE	0456'	0404	0771
READ		0053	
READ1	0CA2'	0840	0903
REPEAT	0A3C'	0641	0576
RESTRT	0E5E'	0998	0705
RP1	0A48'	0645	0646
RS1	0E6A'	1002	1004 1007
S3	0E48'	0987	0985
S4	0E4E'	0989	0979
S41	0E58'	0992	0993
S5	0E5C'	0994	0990

LABEL	VALUE	DEFN	REFERENCES								
SAYPHR	0C68'	0820	0729	0740	0750	0763	0775	0788	0798	0808	0818
			0821								
SET	0468'	0413	0782								
SEVEN	046A'	0414	0854								
SIX	0472'	0418	0725								
SKIP	0AC0'	0695	0692	0697	0700						
SPEAK	0E14'	0969	0653	0820	0855	0870	0885				
START	0922'	0534	0056								
SWITCH	047C'	0423	0779								
T	0366'	0284	0718	0722							
TO	0D8A'	0906	0934								
TO5	0D92'	0908	0930								
T1	0DAE'	0917	0911	0924							
T2	0DBE'	0925	0920								
TABEND	04B2'	0450	0652								
TABTOP	0340'	0265	0651	0938							
THE	0486'	0428	0717	0744	0752						
THREE	0488'	0429	0723								
TIME	048A'	0430	0732								
TSTCHR	0AB0'	0688	0686								
TWO	0490'	0433	0733	0755	0783						
UN1	0ECC'	1043	1041								
UN2	0ED4'	1046	1044								
UN9	0EF6'	1059	1047	1049	1057						
UNLOCK	0EBA'	1037	0857	0867	0872	0882	0887	0897			
W001	0004'	0068	0661	0904	1001						
W038	0058'	0105	0662								
WAIT	0DF6'	0947	0942	0975							
WAIT1	0DFE'	0949	0950								
WAIT2	0E02'	0951	0952								
WORD	052E'	0462	0837								
WORDL	09C6'	0595	0578								
WPMAIN	FF50	0045	0535	0543	0609	0648	0726	0737	0747	0760	0772
			0785	0790	0800	0810	0969	1008			
WPSUBR	FF30	0044	0607	0660	0947	0998	1013				
WPTOP	FF90	0047	0906	0927	0932	0980	1024				

APPENDIX C

SPEECH SET, ALPHABETICAL LISTING

Word	Address (Hexadecimal)	
A	09CC	
Abort	135E	
Adjust	0A0A	
Alert	13C6	
All	0A6A	
Amps	142E	
And	0AA0	
Automatic	146E	
B	14D0	
Back	150C	
Button	1550	
C	15A0	
Calibrate	0000	
Call	15E4	
Cancel	1628	
Carpenter	1690	
Check	006A	
Clock	1710	
Close	00A0	
Control	010A	
Crane	016E	
Crease	0B0A	
Cycle	0B40	
D	1740	
Danger	01D8	
Days	178A	
Degrees	17E4	
Delay 1	38BE	(80 ms pause)
Delay 2	38B6	(160 ms pause)
Delay 3	38AE	(240 ms pause)
Delay 4	38A6	(320 ms pause)
Device	0246	
Direction	1844	
Display	18C2	
Door	0B9E	
Down	0BCC	
E	1910	
East	02DC	
Eight	02AC	
Electrician	193A	
Eleven	19AC	
Enter	1A12	
Equal	0C26	
Exit	1A5C	
F	1A9C	
Fail	1AD0	
Farad	1B24	
Fast	0C7C	
Feet	0CD6	
Fire	0316	
Five	03C0	

Word	Address (Hexadecimal)
Flow	1B92
Foreman	1BC6
Four	0372
Frequency	1C1E
From	1C8A
G	1CE4
Gage	2628
Gate	040E
Get	0D20
Go	0D5C
Green	2EF2
H	1D14
Henry	2696
Hertz	2F30
High	0DB0
Hold	1D4E
Hours	26E0
Hundred	2F7E
I	1DC0
Inch	2736
Initialize	0E0C
Inspector	044A
Intruder	2FF8
Is	04B4
J	1DF6
Jog	276E
K	305A
L	1E42
Left	27EE
Light	309E
Line	1E8E
Low	281C
M	30F0
Machine	04F0
Maker	1F0C
Manual	0576
Measure	0EA0
Mega	2870
Meter	312E
Micro	1F5C
Mill	317A
Milli	28CE
Minus	0F02
Minutes	1FD0
Motor	0F6A
Move	2906
N	05EA
Nine	31C2
North	202A
Number	0FC2

Word	Address (Hexadecimal)
0	297C
Of	0622
Off	320C
Ohms	2078
On	0666
One	06FC
Open	1032
Operator	29A4
Out	06C0
Over	3250
P	20CC
Pass	2A0C
Passed	328C
Percent	20F6
Pico	2A4C
Plus	32CC
Point	2146
Position	108E
Pound	2A82
Power	3300
Press	2188
Pressure	1116
Priority	2AC8
Probe	3346
Processing	21EA
Pull	2B42
Push	33A0
Put	1168
Q	2264
R	2B72
Range	074E
Ready	33D6
Red	229C
Repair	2B9E
Repeat	341A
Replace	07BC
Right	22D4
S	2C16
Safe	2CCA
Seconds	346C
Set	1192
Seven	23AE
Shut	11C2
Six	355E
Slow	2326
Smoke	2C3E
South	359C
Speed	34D8
Start	236E
Stop	2C90
Switch	082C

Word	Address (Hexadecimal)
T	2404
Temperature	1202
Ten	2D06
Test	127A
The	087A
Thousand	35E4
Three	249A
Time	08B6
Timer	2454
Tool	2D3A
Turn	12D0
Twelve	3662
Two	0900
U	2D74
Under	36B4
Unit	130C
Up	24DE
V	2DA2
Valve	093A
Volts	36FA
W	2512
Wait	2DD4
Ward	3758
Watts	255A
Welder	2E04
West	37A8
X	25AA
Y	2E52
Yellow	37EC
Z	25D0
Zero	2E94

APPENDIX D

SPEECH SET, NUMERICAL LISTING

Word	Address (Hexadecimal)
Calibrate	0000
Check	006A
Close	00A0
Control	010A
Crane	016E
Danger	01D8
Device	0246
Eight	02AC
East	02DC
Fire	0316
Four	0372
Five	03C0
Gate	040E
Inspector	044A
Is	04B4
Machine	04F0
Manual	0576
N	05EA
Of	0622
On	0666
Out	06C0
One	06FC
Range	074E
Replace	07BC
Switch	082C
The	087A
Time	08B6
Two	0900
Valve	093A
A	09CC
Adjust	0A0A
All	0A6A
And	0AA0
Crease	0B0A
Cycle	0B40
Door	0B9E
Down	0BCC
Equal	0C26
Fast	0C7C
Feet	0CD6
Get	0D20
Go	0D5C
High	0DB0
Initialize	0E0C
Measure	0EA0
Minus	0F02
Motor	0F6A
Number	0FC2

Word	Address (Hexadecimal)
Open	1032
Position	108E
Pressure	1116
Put	1168
Set	1192
Shut	11C2
Temperature	1202
Test	127A
Turn	12D0
Unit	130C
Abort	135E
Alert	13C6
Amps	142E
Automatic	146E
B	14D0
Back	150C
Button	1550
C	15A0
Call	15E4
Cancel	1628
Carpenter	1690
Clock	1710
D	1740
Days	178A
Degrees	17E4
Direction	1844
Display	18C2
E	1910
Electrician	193A
Eleven	19AC
Enter	1A12
Exit	1A5C
F	1A9C
Fail	1AD0
Farad	1B24
Flow	1B92
Foreman	1BC6
Frequency	1C1E
From	1C8A
G	1CE4
H	1D14
Hold	1D4E
I	1DC0
J	1DF6
L	1E42
Line	1E8E
Maker	1F0C
Micro	1F5C
Minutes	1FD0

Word	Address (Hexadecimal)
North	202A
Ohms	2078
P	20CC
Percent	20F6
Point	2146
Press	2188
Processing	21EA
Q	2264
Red	229C
Right	22D4
Slow	2326
Start	236E
Seven	23AE
T	2404
Timer	2454
Three	249A
Up	24DE
W	2512
Watts	255A
X	25AA
Z	25D0
Gage	2628
Henry	2696
Hours	26E0
Inch	2736
Jog	276E
Left	27EE
Low	281C
Mega	2870
Milli	28CE
Move	2906
O	297C
Operator	29A4
Pass	2A0C
Pico	2A4C
Pound	2A82
Priority	2AC8
Pull	2B42
R	2B72
Repair	2B9E
S	2C16
Smoke	2C3E
Stop	2C90
Safe	2CCA
Ten	2D06
Tool	2D3A
U	2D74
V	2DA2
Wait	2DD4
Welder	2E04
Y	2E52
Zero	2E94

Word	Address (Hexadecimal)	
Green	2EF2	
Hertz	2F30	
Hundred	2F7E	
Intruder	2FF8	
K	305A	
Light	309E	
M	30F0	
Meter	312E	
Mill	317A	
Nine	31C2	
Off	320C	
Over	3250	
Passed	328C	
Plus	32CC	
Power	3300	
Probe	3346	
Push	33A0	
Ready	33D6	
Repeat	341A	
Seconds	346C	
Speed	34D8	
Six	355E	
South	359C	
Thousand	35E4	
Twelve	3662	
Under	36B4	
Volts	36FA	
Ward	3758	
West	37A8	
Yellow	37EC	
Delay 4	38A6	(320 ms pause)
Delay 3	38AE	(240 ms pause)
Delay 2	38B6	(160 ms pause)
Delay 1	38BE	(80 ms pause)

APPENDIX E

PARTS LIST

Symbol	Description	Qty
C01-C06,C22	Capacitor, tantalum, 68 uFd, 15 V, 10%	7
C07	Capacitor, fixed, 10 uFd, 200 V, 10%	1
C08,C09	Capacitor, fixed, ceramic, 0.0047 uFd, 200 V,10%	2
C10	Capacitor, fixed, ceramic, 0.047 uFd, 100 V, 10%	1
C11,C22	Capacitor, fixed, ceramic, 0.1 uFd, 100 V, 10%	2
C13	Capacitor, fixed, tantalum, 1.0 uFd, 50 V, 10%	1
C14-C21,C23-C36	Capacitor fixed, axial lead, 0.047 uFd, 25 V, +80-20%	22
C37	Capacitor, fixed, 0.10 uFd, 50 V, 10%	1
C38	Capacitor, fixed, axial lead, 330 pFd, 25 V, 2%	1
CR1-CR16	Diode, IN914B, switching, 75 PIV	16
E1-E18,TP1,TP2	Pin, 0.25 in. square	20
Q01,Q02	IC, voltage regulator, neg. 3-term, 5 V	2
Q03-Q07	Transistor, AST 2907	5
R01,R37,R43	Resistor, fixed, 1 kilohm, ¼ watt, 5%	3
R02,R03,R45,R47	Resistor Network, 3.3 kilohm, 2%	4
R04,R55,R56	Resistor, fixed, 470 ohms, ¼ watt, 5%	3
R05	Resistor, 2.7 ohms, ¼ watt, 5%	1
R06,R48	Resistor, fixed, 10 kilohm, 2%	2
R07,R35,R36,R51,R52	Resistor, fixed, 10 kilohm, ¼ watt, 5%	5
R18-R21,R30	Resistor, fixed, 47 kilohm, ¼ watt, 5%	5
R22,R24,R26,R28,R44,R54,R57	Resistor, fixed, 2.2 kilohm, ¼ watt, 5%	7
R23,R25,R27,R29	Resistor, fixed, 3.9 kilohm, ¼ watt, 5%	4
R31	Resistor, variable, 100 kilohm, 20 turn	1
R32,R33	Resistor, fixed, 47 ohms, ¼ watt, 5%	2
R34,R38	Resistor, fixed, 33 kilolm, ¼ watt, 5%	2
R39-R42,R53	Resistor, fixed, 100 kilohm, ¼ watt, 5%	5
R46	Resistor Network, 3.3 kilohm, 6 Pin	1
R49	Resistor Network, 10 kilohm, 2 Pin	1
R50	Resistor, fixed, 15 kilohm, ¼ watt, 5%	1
R58	Resistor, fixed, 75 kilohm, ¼ watt, 5%	1
R59	Resistor, variable, 20 kilohm, 1 turn	1

Symbol	Description	Qty
U01	EPROM, speech data 6	1
U02	EPROM, speech data 8	1
U03	IC, TMS 0281A speech synthesizer	1
U04	EPROM, speech data 5	1
U05	EPROM, speech data 7	1
U06	IC, 0A747CN, op amp	1
U08	EPROM, speech data 4	1
U09,U17,U19,U26,U34	Network, SN74LS163	1
U10	Network, SN74LS201	1
U12-U15	IC, CD4016BE, analog switch	4
U16	EPROM, speech data 3	1
U18	Network, SN74LS20	1
U20	Network, SN74LS109	1
U21	Network, SN 74LS04	1
U22	IC, TMS 990	1
U23,U40	Network, SN74LS00	2
U24,U11,U41	Network, SN74LS08	3
U25	EPROM, speech data 2	1
U27,U35	Network, SN74LS138	2
U28	Network, SN74LS194	1
U29	Network, SN74LS74	1
U30	Network, SN74LS09N	1
U31	PROM, SE2 logic array	1
U32	EPROM, SE2 program	1
U33	EPROM, speech data 1	1
U36	PROM, CRU decode	1
U37	Network, SN74LS175	1
U38	IC, TMS 1099 CPU	4
U39	IC, LM380, audio amplifier	1
XU01,XU02,XU04,XU05,XU08,XU16,XU25 XU32,XU33	Socket, 24 Pin	9
XU03,XU07	Socket, 28 Pin	2
XU22	Socket, 40 Pin	1
XU31,XU36	Socket, 16 Pin	2
Jumper Plugs	These plugs are available from: Berg Electronics, Inc. Rt. 3 New Cumberland, Penn. 17070 Berg part number 65474-005	2

APPENDIX F

PROGRAMMING 74LS287 PROM FOR UNIQUE CRU ADDRESSES

F.1 INTRODUCTION

This appendix contains detailed information on coding the speech board's 74LS287 PROM (U36); this allows the user to choose his own CRU address for the TMS 9901 on the TM 990/306 speech module. The user can program the PROM for one to four jumper-selectable addresses similar to that described in paragraph 2.4. Address bus lines are interpreted by the PROM, with a correct address causing the PROM to output an enabling zero to the chip enable (CE-) of the TMS 9901 on the speech module. When enabled, the TMS 9901 is used to interpret word addresses and control signals.

F.2 CRU ADDRESS NOMENCLATURE

Refer to Figure F-1 for the definitions below:

- CRU hardware base address: contents of bits 3-14 of register 12.
- CRU bit address: CRU hardware base address plus a signed displacement contained in the CRU instruction.
- CRU software base address: contents of bits 0 to 15 of register 12.

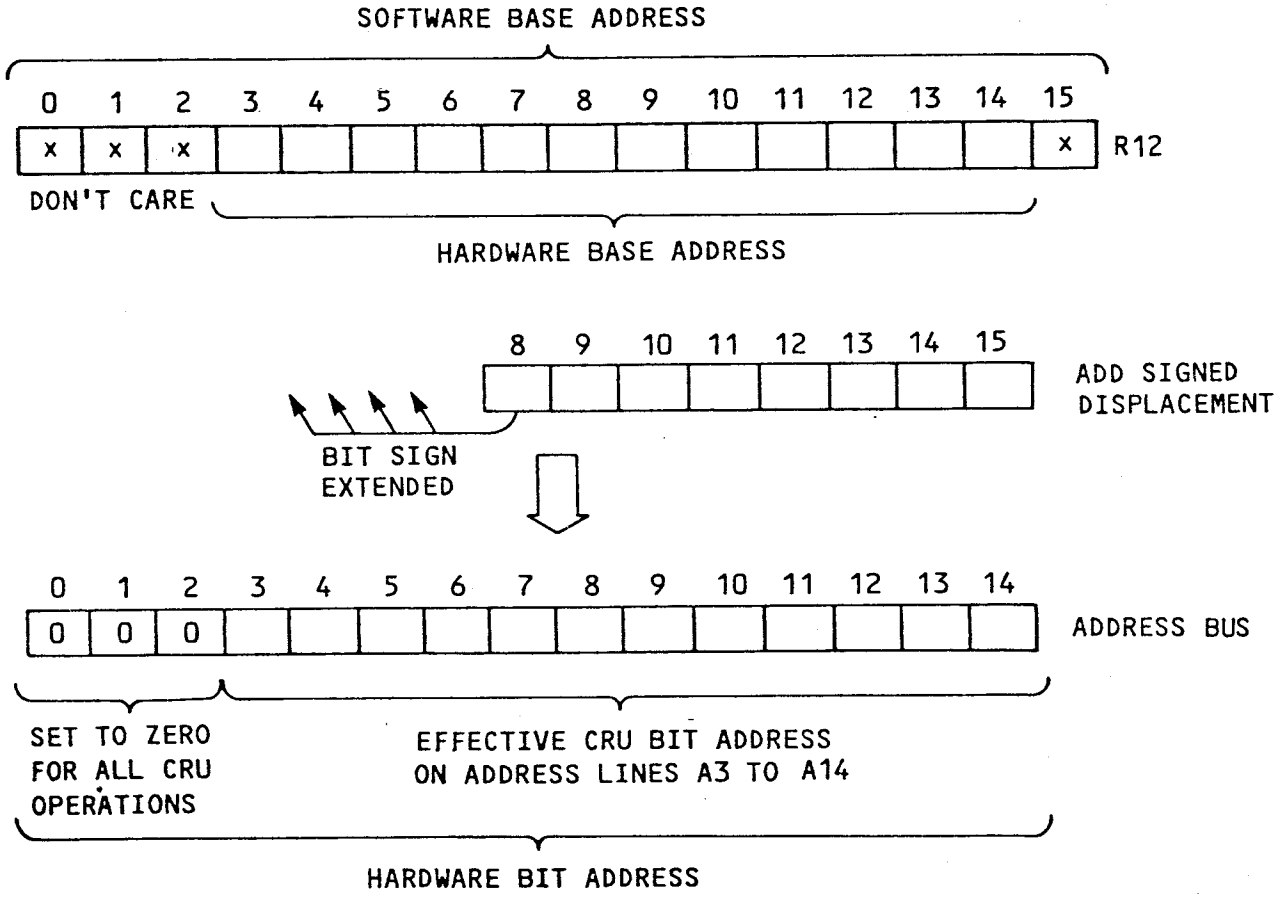


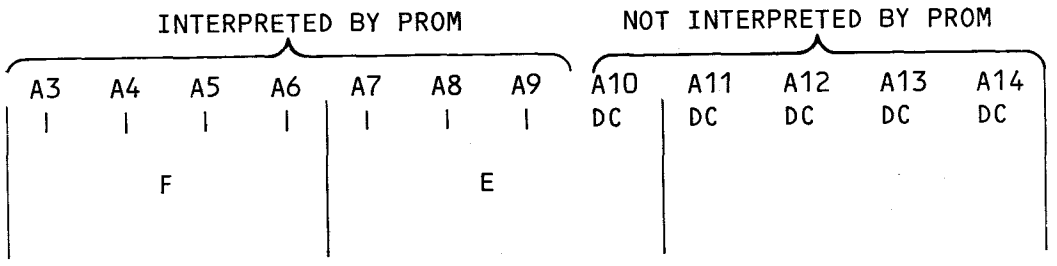
FIGURE F-1. CRU ADDRESS NOMENCLATURE

F.3 CRU MAP SELECTION AND PROGRAMMING CONSIDERATIONS

The following considerations must be taken into account when setting up the CRU map and programming the PROM:

- The TM 990/100 and /101 microcomputer boards reserve the first 256 bits (sixteen 16-bit words) of the total 4096-bit CRU space for I/O interrupt lines and on-card expansion. Therefore, CRU hardware base addresses $0000\text{-}00FF_{16}$ (contained on A3-A14 or software base address $0000\text{-}01FE_{16}$) are reserved, and CRU hardware base addresses $0100_{16}\text{-}0FFF_{16}$ are available for expansion.

- Only the seven address lines A3-A9 are decoded by the PROM, and the A10 input to the PROM is held high (a one); thus, the values on address lines A10 to A14 are "don't cares" that do not affect the selecting of the PROM output. The 32 CRU bit addresses (0 to $1F_{16}$) of address lines A10-A14 do not affect the address translation. For example, if the seven address lines A3-A9 contain address FE_{16}; CRU bit addresses $FE0_{16}$ to FFF_{16} would all enable the same PROM nibble located at CRU bit address $7C0_{16}$ as shown below:



- The PROM outputs are wired to the CE- input to the TMS 9901 on the speech module. A zero to CE- will enable the address inputs to the TMS 9901 so that it can be used to interpret the incoming word address and control signals.

- Pin H of the PROM's address-input pins is held high (a one) as shown in Figure F-2. This pin would normally have been the input from address line A10; however, the address line has no effect on pin H. Do not jumper E9-E10 (See Figure F-2) as this will disable the speech module TMS 9901 address inputs. This feature is reserved for future speech module applications.

- Each of the four jumper pairs (E1-E2 to E7-E8) can be used to select a unique CRU address by programming four addresses (of the 256 addresses) so that four different patterns of three ones and one zero occur at D01 to D04. The zero, when jumpered to the CE- of the TMS 9901, will enable the TMS 9901. Thus, select four PROM addresses and program the PROM nibbles so that a zero appears at the desired output pin (74LS287 PROMs are normally a one and programmed to a zero). See Figure F-2.

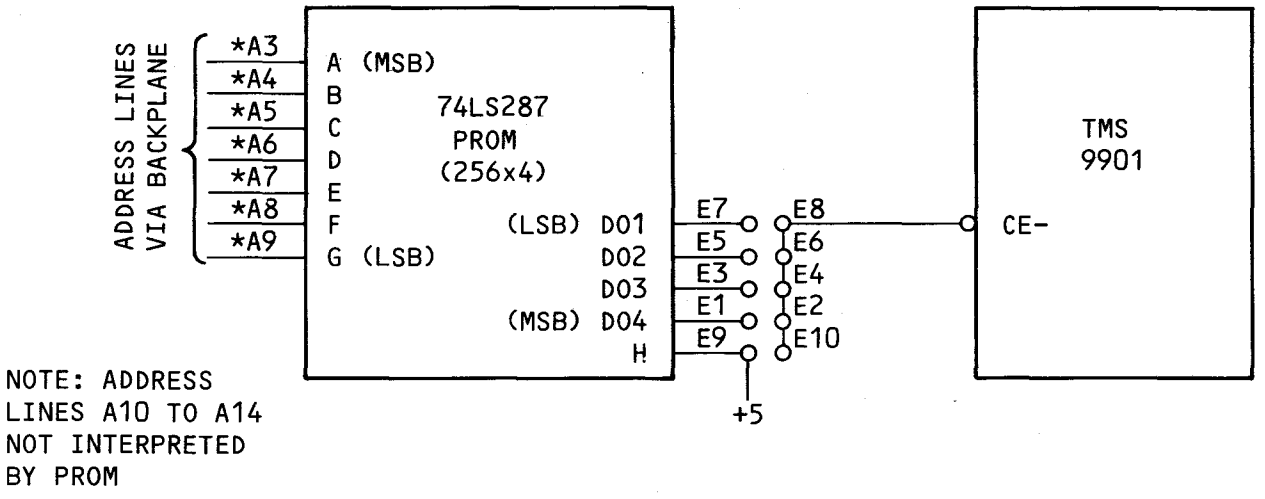


FIGURE F-2. PROM U36 (74LS287) ADDRESS AND DATA PINS

F.4 DETERMINE PROM ADDRESS TO BE CODED WITH ZEROES

Use the following steps to code the PROM correctly:

1) Determine the desired hardware base address (bits 3 to 14 of R12) of the PROM.

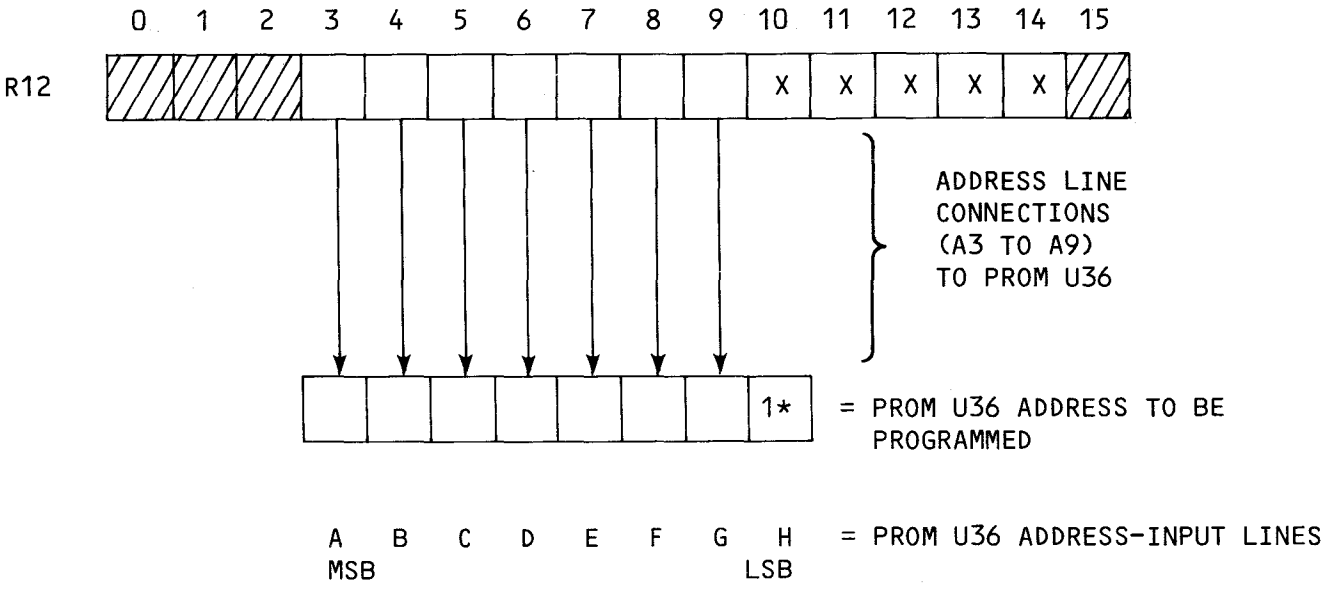
2) Insert the binary value of the hardware base address in the 12 blanks in the top of Figure F-3. Note that only bits 3-9 are actually used to address the PROM. Address bits A10-A14 are not decoded by the PROM and thus are treated as "don't cares" (See Figure F-2).

3) Transfer the binary value of bits 3 to 9 in the top of Figure F-3 to the corresponding numbered boxes in the bottom of the figure. Note that the H address-line input will always be a one (tied high); this must be included in determining the PROM address to be programmed. The resulting 8-bit value in the bottom boxes will be the binary PROM address (inputs to PROM pins A to H) of the nibble to be programmed.

4) The 4-bit value to be programmed at the derived PROM address depends upon which jumper pair will be jumpered for the address. The following table shows the required bit pattern (the 74LS287 is naturally all ones that have to be programmed to zeroes).

JUMPER PAIR	FOUR-BIT VALUE PROGRAMMED AT PROM ADDRESS
E7-E8	1110
E5-E6	1101
E3-E4	1011
E1-E2	0111

D01 OUTPUT
D04 OUTPUT (MSB)



*NOTE: Line H is tied to +5V (HIGH)

FIGURE F-3. INTERPRETING R12 CONTENTS INTO ADDRESS OF PROM NIBBLE

F.5 PROGRAMMING EXAMPLES

This subsection illustrates the development of a PROM address from the desired CRU hardware base address and corresponding jumper pins.

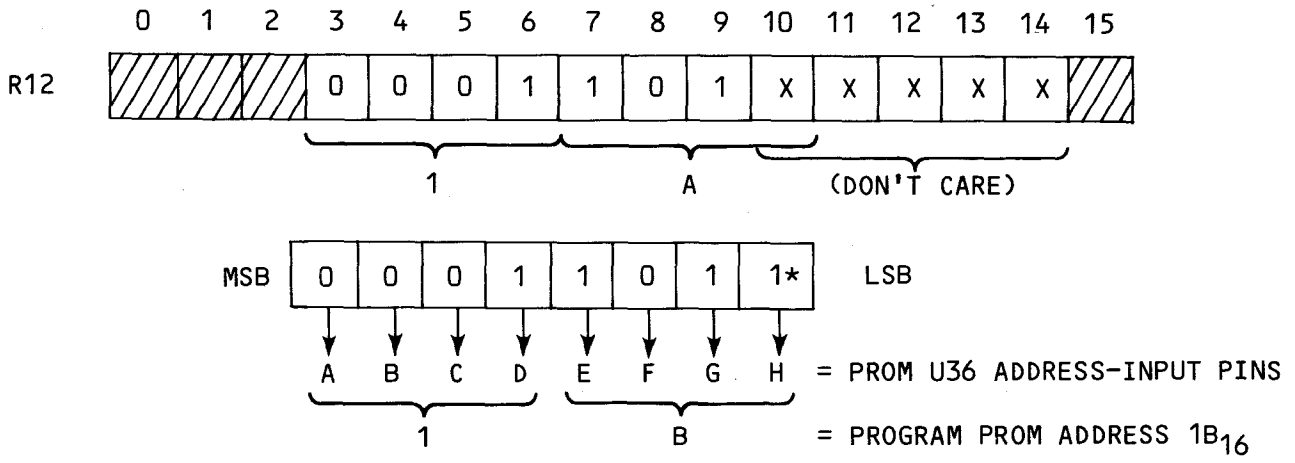
F.5.1 Example 1

In this example, the desired hardware base address is $01A0_{16}$, and the jumper pair for this address is E7-E8.

1) Step 1: CRU hardware base address = $01A0_{16}$
CRU software base address = 0340_{16}

2) Step 2: Interpret nibble address in PROM.

3) Step 3: Determine nibble contents for a low to jumpered pair E7-E8.



*Input to line H is always a one.

Therefore, the PROM nibble at address $1B_{16}$ should be programmed to 1110_2 to enable the TMS 9901 when E7-E8 is jumpered (a low on output pin DO1 with ones to the other jumper positions). Jumpers and PROM address contents are explained in paragraph F.4 (step 4).

Note that the nibble can be enabled by using CRU hardware base addresses $01A0_{16}$ to $01BF_{16}$ (software base addresses 0340_{16} to $037E_{16}$) because only bits 3 to 9 of R12 are used to select the PROM address.

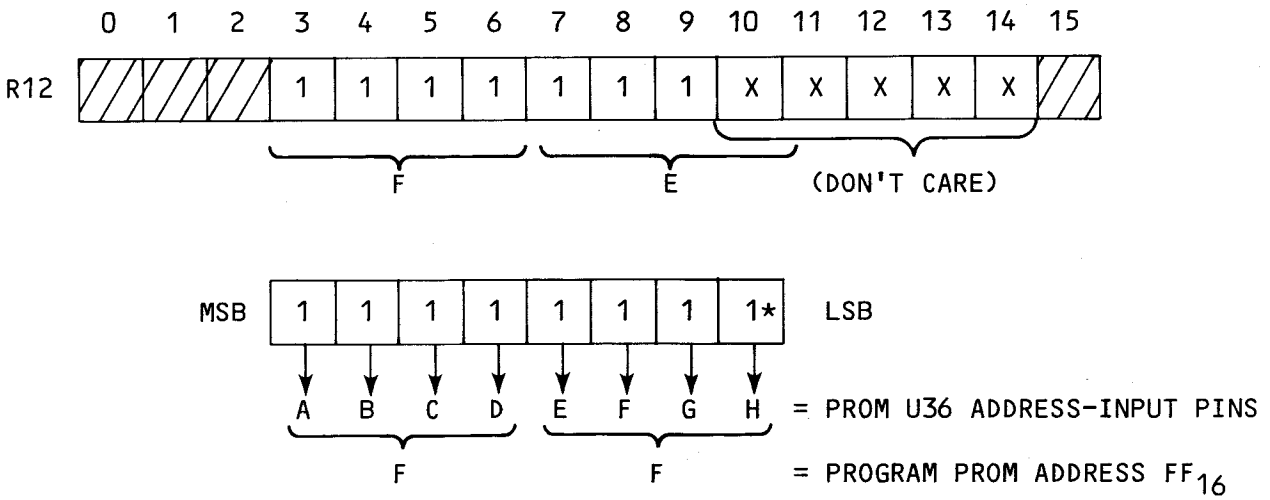
F.5.2 Example 2

In this example, the desired CRU software base address is $1FC0_{16}$ to enable the TMS 9901 via jumper pins E1-E2.

1) Step 1: CRU software base address = $1FC0_{16}$
   CRU hardware base address = $0FE0_{16}$

2) Step 2: Interpret nibble address in PROM.

3) Step 3: Determine nibble contents for a low through jumpered pair E1-E2.



*Input to line H is always a one.

Therefore, the PROM nibble at address FF_{16} should be programmed to 0111_2 to output a low at output pin DO4. This will enable the TMS 9901 via jumper pins E1-E2.

APPENDIX G

TM 990/306 P2 AND P3 PINOUTS

PIN	P2	P3	PIN	P2	P3
1	GND	GND	21	-5V	NC
2	*R7	NC	22	-5V	AMP OUT
3	*R0	GND	23	NC	*R7
4	*R1	NC	24	NC	NC
5	*R2	NC	25	NC	S13
6	*R3	NC	26	NC	S12
7	K1	GND	27	NC	S11
8	K2	NC	28	NC	S10
9	K4	S1(LSB)	29	FILTER OUT	S9
10	K8	NC	30	NC	GND
11	PDC	S2	31	NC	S3
12	SP1	FILTER OUT	32	NC	NC
13	NC	NC	33	NC	S8
14	SP2	GND	34	NC	NC
15	NC	S16(MSB)	35	NC	S4
16	AMP OUT	NC	36	NC	GND
17	NC	S15	37	-10V	S7
18	NC	NC	38	-10V	S5
19	NC	S14	39	GND	S6
20	NC	GND	40	GND	GND

INDEX

In this index, page numbers preceded by (F) indicate a figure on the page page numbers preceded by a (T) indicate a table on the page.