GENERAL INSTRUMENT APPLICATION NOTE

Microchip™

Formerly
GENERAL
INSTRUMENT
MICROELECTRONICS

CTS 256A-AL2 CODE-TO-SPEECH I.C.

FEATURES:

* Translates an unlimited vocabulary of English words into speech output.

* 20 byte input buffer, expandable to 1792 bytes.

* Input buffer full handshake singnal.

* Serial port built-in.

* Serial port parameters selectable.

* Parallel port option.

* Power-on system reset.

DESCRIPTION

The CTS256A-ALs is an 8-bit microcomputer programmed to create text to allophone-address sequences, in a flexible and cost effective manner. It sources the SP0256A-AL2 which is a speech synthesizer whose output drives an audio amplifier to produce speech output.

Input to the CTS256A-AL2 is standard English ASCII characters, which makes connecting to any stand-alone terminal or any personal computer simple.

Upon power-up or use of the reset switch, the system initializes itself and then speaks "O.K." to demonstrate that it is ready for input.

SCHEMATIC I:

Schematic 1 shows the minimum component configuration. It uses serial input whose serial parameters are 7 bits per character, 2 stop bits, no parity; and uses Internal-RAM input buffer. This buffer accomodates words that are no greater than 19 characters in length followed by a delimeter; with an output buffer that accomodates an allophone translation of that word this is no greater than 26 bytes. Since the translation more often than not results in the output buffer contents consisting of two times that of the input buffer, words no longer than 13 characters in length should be used as a rule of thumb. If a translation results in an over-flowed output buffer, the system reset switch may have to be used to clear the system.

SCHEMATIC 2:

Schematic 2 shows the configuration necessary to incorporate a parallel port, a 1792 byte input buffer (External-RAM), and selectable serial parameters. These three options are mutually exclusive.

The parallel port accepts standard English ASCII characters. The parallel data strobe signifies that the data is valid, and latches the data.

External-RAM mode can be used to extend the size of the imput buffer to 1792 bytes. In this mode, two pages of text can be loaded into the system. The remainder of the 2K x 8 byte RAM is 256 bytes which is used for the output buffer. (The output buffer is the area where the strings of allophone addresses are held prior to transfer into the SP0256A-AL2.)

The serial parameters option may be selected if the host terminal or computer can not be set to accomodate the default parameters.

INTERNAL-RAM/EXTERNAL-RAM OPTION:

In either Internal-RAM or External-RAM mode, the input buffer is protected from overflow by hysteresis which signals the most when the input buffer is full, and when ready for additional input. BUSY (pin 03) toggles LO when he input buffer becomes 100% full, the parallel and serial port interrupts and disabled to prevent input buffer overwrite. The interrupts are not re-enabled until the BUSY condition has disapated. BUSY condition has disapated. BUSY will toggle HI when the input buffer becomes 50% empty.

CONTROL CHARACTERS:

"ESC" The Escape key will dump the current input buffer contents. It may also be used to silence speech output that is in progress.
"---" The Backspace key erases the input buffer one character at a time, starting with the latest entry.

ANY-DELIMETER / CARRIAGE-RETURN-ONLY OPTION:

In the any-delimiter mode, the code-to-speech algorithm will process and speak words or phrases as soon as they are followed by any delimeter. These include . , ; : ? space, carriage return, etc. In the carriage-return-only mode, the algorithm will process and speak only after a carriage-return is received. The latter mode is meant for use with a slow input device such as a terminal, where the user wishes to buffer-up a complete phrase so that it is spoken with fluency.

If the carriage-return-only option is chosen (with External-RAM), limit to 160 characters the length of the phrase which is entered before the carriage return is entered. (This rule of thumb allows for a two line phrase to be spoken with fluency, yet insures that the 256 byte output buffer does not overflow.)

TABLE 1

```
NOTE:  These pin-outs refer to U1, the CTS256A-ALs intergrated circuit.
       <---- represents an input
       ----> represents an output

PIN 61718
    0|0|0   ----PARALLEL INPUT MODE
    0|0|1   BAUD 50        +
    0|1|0   BAUD 110       1
    0|1|1   BAUD 300       1 SERIAL INPUT MODE
    1|0|0   BAUD 1200      1
    1|0|1   BAUD 2400      1
    1|1|0   BAUD 4800      1
    1|1|1   BAUD 9600      +

PIN 09
     0 ---PROGRAMMED DEFAULT SERIAL PARAMETERS (7 bits/character, 2 stop
                                                bits, no parity).
     1 ---SELECTABLE SERIAL PARAMETERS.  (Refer table 2.)

PIN 10
     0 ---INTERNAL RAM BUFFERS.  (20 byte input/26 byte output.)
     1 ---EXTERNAL RAM BUFFERS.  (1792 byte input/256 byte output.)

PIN 11
     0 ---CARRIAGE-RETURN-ONLY DELIMITER.
     1 ---ANY DELIMITER.

PIN 03
     0 ---INPUT BUFFER IS "BUSY".
     1 ---INPUT BUFFER IS "NOT BUSY".
```

TABLE 2 SELECTABLE SERIAL PARAMETERS:

U10 PIN 13|14|
0|1 16 BITS/CHARACTER
1|0 17 BITS/CHARACTER
1|1 18 BITS/CHARACTER

U10 PIN 8
1=PARITY ENABLED
0=PARITY DISABLED

U10 PIN 7
1=EVEN PARITY
0=ODD PARITY

U10 PIN 3
0=ONE STOP BIT
1=TWO STOP BIT

```
+------------------------------------+          +------------------------------------+
!   GENERAL      APPLICATION NOTE    !          !             SP0256A-AL2            !
! INSTRUMENT     AN-0505 REVISION D  !          !             CTS256A-AL2            !
+------------------------------------+          +------------------------------------+
```

CODE-TO-SPEECH CHIP SET

INTRODUCTION FOR REVISION D OF THE AN0505

This application note has been updated in order to cover recent customer claims regarding problems with the downloading of entire text files from a computer to the CTS256A-AL2 via the serial or parallel interface.
There is a problem within the CTS256A-AL2 regarding the buffer management and the handshaking signal (BUSY#). The BUSY# signal covers only the input buffer, but not the output buffer which normally overflows much earlier than the input buffer as one single character in the input buffer can result in a string of allophones which have to be addressed by the contents of the output buffer. Due to this, it can happen that some characters will be skipped or lost and in some cases even the entire system gets lost and has to be reset.
The changes with respect to revision C are all covered on this first page. All other sections have been left unchanged regarding their contents but have been reformatted.

FILE GENERATION AND DATA TRANSMISSION HINTS

In order to translate computer text files successfully into spoken words the following system configuration and set up is recommended:

- Baud rate 110 baud (or equivalent character/sec rate for the parallel interface)
- Any-delimiter-mode
- External RAM buffer

Due to the above mentioned CTS256A-AL2 problem, the BUSY# signal can be disregarded.
The maximum file length should not exceed about 4000 characters (this value depends on the text to be spoken and should be even less if a text contains a lot of numbers as they result in a high " # of allophones per character" rate and the output buffer will be filled more quickly than with pure text files).
If more than one file has to be transmitted it is recommended to send an "ESC" character after each file has been spoken completely. This resets the input and output buffer.
Caution: When the CTS256A-AL2 receives an "ESC" character the speech output is stopped immediately without completion of an already commenced phrase.

EXCEPTION WORD EPROM AND USER EPROM

The implemetation of exception word and user EPROM's as descibed in this application note requires a high level of system knowledge as well as PIC7000 development tools. Therefore General Instrument suggests that the private user does not implement these features.

SYSTEM PERFORMANCE

The user can get a satisfying system performance with a configuration as described above. In fact, as the average duration of a spoken word is in the range of 250 to 1000 msec, a fluent speech output is still possible even with that slow data transmission rates.
If the text is generated online (e.g. using a terminal) the time needed to type the characters is the real limiting factor to the system speed. So there is almost no influence of the data transmission speed itself on the system performance. However, baudrates up to 9600 baud can be used in that mode without problems if the restrictions as described in the sections "INTERNAL RAM" and "ANY DELIMITER/CARRIAGE RETURN ONLY" are regarded.

FEATURES:

-Unlimited vocabulary
-Utilizes letter-to-sound rules
-Serial or parallel interface
-Microprocessor available for user code

DESCRIPTION

The Code-to-Speech chip set consists of two chips: the SP0256A-AL2, an allophone-based single chip speech synthesizer, and the CTS256A-AL2, an 8-bit microcomputer programmed with a letter-to-sound based algorithm. This chip set translates English characters into LPC synthesized speech sounds.

SP0256A-AL2

The SP0256A-AL2 is General Instrument's standard allophone chip and is based on the SP0256A speech synthesizer. This synthesizer consists of a 10 or 12 pole second-order cascaded LPC filter, a controller, and a 16-Kbit ROM in which 59 allophones (speech sounds) and five pauses are stored.

CTS256A-AL2

The CTS256A-AL2 is a PIC7041 whose on-board ROM is masked with our code-to-speech algorithm. This algorithm converts English text (in the form of standard ASCII characters) into SP0256A-AL2 compatible allophone addresses, using letter-to-sound rules.

General Instrument's PIC7041 is a licensed second source of the TMS7041.

TWO-CHIP SOLUTION

This chip set delivers highly recognizable speech output from any peripheral device or computer in a flexible and cost effective manner. It can be configured as a dedicated code-to-speech system, as well as add speech output to a user's program running in this CTS256A-AL2 from off-chip Rom. Such user programs are written in PIC7001 assembly language which is 100% compatible with TMS7001 assembly language. (Refer to PIC7001 Data Sheet, DS33001A).

Eproms can be added to improve the pronunciation of certain proper names, acronyms and technical words [see "EXCEPTION-WORD" eprom]; as well as to store user programs [see "USER" eprom].

PIN SELECTABLE CODE-TO-SPEECH OPTIONS:
Refer to TABLE 1.

INPUT INTERFACE	-Serial port & baud rate vs. Parallel port
INPUT BUFFER	-Internal RAM vs. External RAM
DELIMITER	-Any-delimeter vs. Carriage-return-only
UART PARAMETERS	-Program defaults vs. 74LS373 selectable (or eprom definable)

FIRMWARE (EXCEPTION-WORD/USER EPROM) CONTROLLED CODE-TO-SPEECH OPTIONS: (optional)
Refer to TABLE 2.

-Parallel port decode relocatable
-UART parameters 74LS373 decode relocatable
-UART parameters selectable
-Start & end address of External-Ram relocatable

CODE-TO-SPEECH ALGORITHM FEATURES:

-ESCAPE: "ESC",(1B Hex) THE ESCAPE-KEY CODE WILL DUMP THE CONTENTS OF THE INPUT AND OUTPUT BUFFERS, AND WILL ALSO SILENCE SPEECH OUTPUT WHICH IS IN PROGRESS.
-BACKSPACE:"<--",(08 Hex) THE BACKSPACE-KEY CODE ERASES THE INPUT BUFFER ONE CHARACTER AT A TIME, BEGINNING WITH THE LATEST ENTRY.

NOTE: The R/C combination indirectly connected to PIN 14 of the CTS256A-AL2 and to PIN 2,25 of the SP0256A-AL2 acts as a power-on reset. The requirement to reset the chip-set is a negative-going pulse which remains Lo for a minimum of 500 uS.

NOTE: A signal (input or output) that is Active-LO is designated by its signal name followed by an asterik (*).

NOTE: The program default address decode of the SP0256A-AL2's ALD* input is 2000H. It is re-definable via the EXCEPTION-WORD or USER eprom. Refer Table 2.

NOTE: MSnibble means most significant nibble, where a nibble is half a byte. MSB means most significant byte; LSB means least significant byte. 'X' stands for the MSnibble of the MSB of the two byte address, and can be 1,2,3,4,5,6,7,8,9,A,B,C,D,or E because an eprom may reside from 1000H to E000H.

NOTE: The term 'delimeter' refers to any punctuation following a word or numerical sequence. These include: , . ; : ! ? spaces and carriage-returns.

CODE-TO-SPEECH ALGORITHM

Upon power-up (or hardware reset) the CTS256A-AL2 determines the system configuration with respect to the following five options:

1- INTERNAL / EXTERNAL RAM SELECTION: (Refer to TABLE 1.)

INTERNAL-RAM mode has an input buffer which accomodates words or phrases that are no greater than 19 characters in length followed by a delimeter; and an output buffer that accomodates an allophone translation of that word or phrase that is no greater than 26 allophone addresses.

Since the translation more often than not results in the output buffer contents consisting of two times that of the input buffer, words no longer than 13 characters in length and numerical sequences no longer than 4 numbers in length should be used as a rule of thumb.
If the output buffer overflows, what has not been spoken yet from the output buffer might be lost,

and the BUSY* flag will not necessarily show an input buffer empty status even though the input buffer might be empty.
If a translation results in an output buffer overflow, the system reset may have to be used to clear the system.

EXTERNAL-RAM mode can be used to extend the size of the input and output buffers. If no EXCEPTION-WORD or USER eproms are present, the start address default is 3000H. Static RAM can be added in 256 byte contiguous block increments, beginning with a minimum of 512 bytes.
The algorithm will find the end address by searching for the first non-RAM location at 256 byte intervals. The search for the end address will not progress beyond 2K bytes.

If an eprom is present, the start and end addresses are re-definable there.
Requirements are: minimum start address is 0200H; the start address must begin on a boundary where the LSByte of the address =00; and without the end address specified in eprom, the maximum valid start address is EE00H.

In any case, 256 bytes are taken for the output buffer; the remainder is the input buffer. (External-Ram used must have an access time of 250 nS or less.)

2- ROM: A search is made from 1000H to E000H in 4K increments for the 5 byte sequence (80H,48H,28H,58H,85H) which uniquely identifies the presents of an EXCEPTION-WORD or USER eprom. If neither are present, the system options are set to algorithm default values or can be chosen by the Pin selectable options. If only a USER eprom is present, the system options may be re-defined from the USER eprom; refer to APPENDIX-D.
If both USER and EXCEPTION-WORD eproms are present or if only an EXCEPTION-WORD eprom is present, the system options may be re-defined from the EXCEPTION-WORD eprom;refer to APPENDIX-A,B.
(External-Rom used must have an access time of 300 nS or less.)

EXCEPTION-WORD EPROM(s): (optional)

Exception-word eprom(s) may reside anywhere within the decodeable address space of the CTS256A-AL2 from 1000H to E000H, providing its start address falls on a 4K boundary. The code-to-speech initialization routine will search for its existence which is denoted by a unique 5-byte sequence of numbers (80H,48H,28H,58H,85H). A few other locations in the primary exception-word eprom are reserved, and must contain specific sequences of numbers; the remainder are user-defined. Additional exception-word eprom(s) contiguous to the primary exception-word eprom contain no reserved locations. Refer to APPENDIX-A,B for the applicable EXCEPTION-WORD EPROM MEMORY MAP.

USER EPROM(s): (optional)

If a USER eprom is accompanied by an EXCEPTION-WORD eprom, it may reside anywhere. If no EXCEPTION-WORD eprom accompanies it then it may reside anywhere from 1000H to E000H providing its start address falls on a 4K boundary; and it must then begin with the sequence 80H,48H,28H,58H,85H; and also contain other reserved locations.
If an EXCEPTION-WORD eprom is present, the USER's program can even reside in an unused portion of the EXCEPTION-WORD eprom. Refer to APPENDIX-D,E for the applicable USER EPROM MEMORY MAP.

Interaction between a USER program and the code-to-speech algorithm must be controlled in an orderly manner, ie: the user must save the processor status before taking control of the processor for execution of any USER code (except for character string loading operations, which is described next:)

To prepare the code-to-speech algorithm to process and speak, the USER program passes the character string it wants spoken into the Accumulator one character at a time, then calls the routine @SAVE which transfers it into the input buffer. After the character string loading has been completed, the USER code can initiate the speech by calling the @SPEAK routine; assuming that a delimeter followed that character string. After the loaded character string is processed and spoken, program control resumes in the hands of the USER program by the Branch @USERCODE instruction.

No registers used by the code-to-speech algorithm may be disturbed by the USER code during character string loading, (except for the Accumulator).
Prior to the USER code executing anything other than character string loading, all registers used by the code-to-speech algorithm as well as the Stack Pointer and STATUS register are to be saved. These registers must be recovered prior to future character string loading operations; or prior to initiating speech.

Because of masked code-to-speech restrictions within the CTS256A-AL2, Interrupt-1* and Interrupt-3* are not USER accessible. Also, input from the serial port into the USER code can be obtained, but restrictions apply.

Refer to APPENDIX-F for a discussion of the sequence of events and subroutines necessary for USER / CODE-TO-SPEECH interactions as described above.

3- SERIAL / PARALLEL INPUT INTERFACE SELECTION: (Refer to TABLE 1.)

In the parallel mode, ASCII data is latched by an 74LS374, upon receipt of an Active-LO data-valid strobe. This strobe also vectors the algorithm to accept the data via Interrupt-3*, PIN 12 of the CTS256A-AL2. The latch's address default is 200H. It is re-definable from EXCEPTION-WORD or USER eprom. (Refer to TABLE 9 for timing requirements of the parallel port.)

In the serial mode, ASCII data is accepted via the CTS256A-AL2 PIN 16, which is a built-in UART that requires a TTL level signal input.
The baud rate is selectable at 50,110,300,1200,2400,4800, and 9600. The other UART parameters are set to algorithm default values, or are hardware selectable via an 74LS373 buffer. The buffer address default is 1000H. The UART parameters as well as the baud rate is re-definable from EXCEPTION-WORD or USER eprom. The algorithm default UART values are: Asynchronous, 7 bits/character, 2 stop bits, and no parity.

In either serial or parallel mode, the input buffer is protected from overflow by a hysteresis subroutine which signals the host when the input buffer is full, and when the input buffer is ready for additional input. Hardware handshaking (BUSY*) is provided to accomplish this signaling of input buffer status.

BUSY* is Active-LO. It toggles LO when the input buffer becomes 87.5% full. In this way the host system may use its discretion to complete that transmission or a part thereof. If the input buffer becomes 100% full, the parallel and serial port interrupts are disabled to prevent input buffer overwrite; and the interrupts are not re-enabled until the input buffer full condition has disapated. BUSY* will toggle HI when the input buffer becomes 50% empty; at which time the interrupts are enabled if they had been disabled by a 100% full condition. (BUSY* is PIN-3 of the CTS256A-AL2 which is a TTL level output capable of sinking 10 mA maximum.)

4- SOFTWARE / HARDWARE (or FIRMWARE) UART PARAMETERS SELECTION: (Refer to TABLE 1.)

This hardware option tells the code-to-speech algorithm to use the default UART values, or to find the parameters at the 74LS373 buffer. The buffer address default is 1000H. The UART parameters are re-definable from eprom, but only if the hardware mode is selected via Pin 9 of the CTS256A-AL2.

5- ANY-DELIMETER / CARRIAGE-RETURN-ONLY SELECTION: (Refer to TABLE 1.)

In the any-delimiter mode, the code-to-speech algorithm will process and speak words or phrases as soon as they are followed by any delimiter. In the carriage-return-only mode, the algorithm will process and speak words or phrases only after a carriage return is received as a delimeter. The carriage-return-only mode is meant for use with a slow input device such as a terminal, where the user wishes to buffer-up a complete phrase so that it is spoken with fluency.
If the carriage-return-only mode is chosen in conjunction with EXTERNAL-RAM, limit to 160 characters the length of the phrase which is entered before the carriage-return is entered.
This allows for a two line phrase to be spoken with fluency while insuring that the 256 byte output buffer should not overflow.

After completion of the initialization the phrase "O.K." is spoken to demonstrate that the system is ready for input, then one of the following two paths is taken dependent upon the system configuration:

1: In a 'dedicated code-to-speech system' (ie; USER eprom is not present), the algorithm idles as long as the input buffer remains empty. Input is via standard ASCII characters. Processing begins with an alphabetical search of the EXCEPTION-WORD eprom, if it is present. If no exact match for the character string is found, or if an EXCEPTION-WORD eprom is not present, the algorithm employs a letter-to-sound rule table against which main, right, and left context matches are performed. This results in the translation of a particular word into the proper string of allophone addresses necessary for its pronounciation. This list of allophone addresses is sent to the SP0256A-AL2 after a carriage-return, or after any delimiter depending upon the mode selected.

2: In the 'add speech to USER's program' mode (ie; USER eprom is present), control of the processor is relinquished to the USER code immediately after the initialization is complete. The USER code may then execute its own code, may pass character strings into the input buffer memory, or may hand-off processor control to the code-to-speech algorithm to speak any previously loaded character strings. If speech is initiated, control returns to the USER code after the last delimited character string in the input buffer has been processed. Refer to APPENDIX-F.

TABLE 1.

Hardware selectable option pin-outs of CTS256A-AL2:

```
PIN 6 7 8
    -----
    0 0 0<---PARALLEL INPUT MODE
    0 0 1 BAUD 50      <+
    0 1 0 BAUD 110      |
    0 1 1 BAUD 300      |
    1 0 0 BAUD 1200     |SERIAL INPUT MODE
    1 0 1 BAUD 2400     |
    1 1 0 BAUD 4800     |
    1 1 1 BAUD 9600    <+
```

```
PIN 9
    0<---PROGRAM DEFAULT UART VALUES (Asynchronous, 7 bits/character, 2 stop bits, no parity).
    1<---HARDWARE (or FIRMWARE) SELECTED UART VALUES.

PIN 10
     0<---INTERNAL-RAM BUFFERS, (20 BYTE INPUT/26 BYTE OUTPUT).
     1<---EXTERNAL-RAM BUFFERS, (1792 BYTE INPUT/256 BYTE OUTPUT WITH A 2-KBYTE RAM),(EX: TMS4016-25).

PIN 11
     0<---CARRIAGE-RETURN-ONLY DELIMITER.
     1<---ANY DELIMITER.

PIN 03  "BUSY*" (Input buffer flag is a TTL level output); for RS232 compatibility use MC1488 Line Driver or
     0--->INPUT BUFFER IS >=87.5% FULL.                                                          equivalent.
     1--->INPUT BUFFER IS <=50.0% EMPTY.

PIN 16-->UART RECEIVER  (Serial input is a TTL level input); for RS232 compatibility use MC1489 Line Receiver
                                                                                               or equivalent.


NOTE:  0 implies TLL LO level; 1 implies TTL HI level.
      <--- implies input; ---> implies output.
```

A typical connection to a computer with an RS232 interface:

```
        COMPUTER                      CODE-TO-SPEECH CHIP-SET
        protective GND     <------> signal GND (Circuit ground).
            signal GND     <------> signal GND (Circuit ground).
   Clear To Send (CTS)     <------- Request To Send (RTS) = CTS256A-AL2's PIN 3 (BUSY*).
Transmitter's Line Driver -------> CTS256A-AL2 UART's Line Receiver.
```

TABLE 2. NEW PARAMETERS.

X009	FF	NUMBER OF BYTES OF 50% OF EXTERNAL INPUT BUFFER (MSB)
X00A	FF	NUMBER OF BYTES OF 50% OF EXTERNAL INPUT BUFFER (LSB)
X00B	FF	NUMBER OF BYTES OF 12.5% OF EXTERNAL INPUT BUFFER (MSB)
X00C	FF	NUMBER OF BYTES OF 12.5% OF EXTERNAL INPUT BUFFER (LSB)
X00D	FF	EXTERNAL RAM START ADDRESS (MSB) see note 2.3
X00E	FF	EXTERNAL RAM START ADDRESS (LSB) see note 2.3
X00F	FF	EXTERNAL RAM END ADDRESS-100H (MSB) see note 2.3
X010	FF	EXTERNAL RAM END ADDRESS-100H (LSB) see note 2.3
X011	FF	EXTERNAL RAM START ADDRESS-1 (MSB) see note 2.3
X012	FF	EXTERNAL RAM START ADDRESS-1 (LSB) see note 2.3
X013	FF	EXTERNAL RAM END ADDRESS-FFH (MSB) see note 2.3
X014	FF	EXTERNAL RAM END ADDRESS-FFH (LSB) see note 2.3
X015	FF	EXTERNAL RAM END ADDRESS+1 (MSB) see note 2.3
X016	FF	EXTERNAL RAM END ADDRESS+1 (LSB) see note 2.3
X017	FF	ADDRESS DECODE OF SP0256A-AL2's ALD* (MSB) see note 2.4
X018	FF	ADDRESS DECODE OF SP0256A-AL2's ALD* (LSB) see note 2.4
X019	FF	ADDRESS DECODE OF 74LS374 PARALLEL PORT LATCH (MSB)
X01A	FF	ADDRESS DECODE OF 74LS374 PARALLEL PORT LATCH (LSB)
X01B	FF	see note 2.1
X01C	FF	TOTAL NUMBER OF BYTES IN INPUT BUFFER (MSB)
X01D	FF	TOTAL NUMBER OF BYTES IN INPUT BUFFER (LSB)
X01E	FF	see note 2.1
X01F	FF	see note 2.1
X020	FF	SERIAL PORT MODE REGISTER (see table 5) see note 2.5
X021	FF	SERIAL PORT CONTROL REGISTER (see table 6) see note 2.5
X022	FF	SERIAL PORT TIMER DATA REGISTER (see table 6) see note 2.5

^
¦
+----- YOUR EXCEPTION-WORD OR USER EPROM CAN RESIDE ANYWHERE FROM 1000H TO E000H PROVIDING IT BEGINS ON A 4K BOUNDARY WHERE X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E. (The least significant 3 nibbles of the address must remain as shown.)

NOTE 2.1 THESE LOCATIONS MUST BE FF, (THEY ARE NOT USER DEFINABLE).

NOTE 2.2 TO MAINTAIN ANY PARAMETER AT ITS DEFAULT VALUE, LOAD THAT LOCATION WITH FFH.

NOTE 2.3 IF ANY OF THE EXTERNAL RAM BUFFER PARAMETERS ARE REDEFINED HERE, ALL OF THEM MUST BE REDEFINED HERE.

NOTE 2.4 NO MATTER WHAT ADDRESS IS CHOSEN FOR ALD*, THAT ADDRESS THRU THAT ADDRESS + 3FH IS RESERVED FOR SP0256A-AL2 ADDRESSING.

NOTE 2.5 IF ANY OF THE SERIAL PORT PARAMETERS ARE REDEFINED HERE, ALL OF THEM MUST BE REDEFINED HERE.

NOTE 2.6 H, AS IN 100H REFERS TO HEXADECIMAL NOTATION.

NOTE 2.7 A NIBBLE IS HALF OF A BYTE, OR 4 BITS.

TABLE 3. SAMPLE OF ASSEMBLED ALPHABETIZED EXCEPTION-WORD INDEX.

X0A3	X1	MSB	OF POINTER TO START OF EXCEPTION-WORD BEGINNING WITH "A"
X0A4	93	LSB	"A"
X0A5	X1	MSB	"B"
X0A6	A8	LSB	"B"
X0A7	X1	MSB	"C"
X0A8	A9	LSB	"C"
X0A9	X1	MSB	"D"
X0AA	B1	LSB	"D"
X0AB	X1	MSB	"E"
X0AC	B2	LSB	"E"
X0AD	X1	MSB	"F"
X0AE	B3	LSB	"F"
X0AF	X1	MSB	"G"
X0B0	B4	LSB	"G"
X0B1	X1	MSB	"H"
X0B2	E1	LSB	"H"
X0B3	X1	MSB	"I"
X0B4	E2	LSB	"I"
X0B5	X2	MSB	"J"
X0B6	0D	LSB	"J"
X0B7	X2	MSB	"K"
X0B8	0E	LSB	"K"
X0B9	X2	MSB	"L"
X0BA	0F	LSB	"L"
X0BB	X2	MSB	"M"
X0BC	1B	LSB	"M"
X0BD	X2	MSB	"N"
X0BE	1C	LSB	"N"
X0BF	X2	MSB	"O"
X0C0	1D	LSB	"O"
X0C1	X2	MSB	"P"
X0C2	1E	LSB	"P"
X0C3	X2	MSB	"Q"
X0C4	2D	LSB	"Q"
X0C5	X2	MSB	"R"
X0C6	2E	LSB	"R"
X0C7	X2	MSB	"S"
X0C8	2F	LSB	"S"
X0C9	X2	MSB	"T"
X0CA	30	LSB	"T"
X0CB	X2	MSB	"U"
X0CC	3D	LSB	"U"
X0CD	X2	MSB	"V"
X0CE	5A	LSB	"V"
X0CF	X2	MSB	"W"
X0D0	5B	LSB	"W"
X0D1	X2	MSB	"X"
X0D2	64	LSB	"X"
X0D3	X2	MSB	"Y"
X0D4	65	LSB	"Y"
X0D5	X2	MSB	"Z"
X0D6	6F	LSB	"Z"
X0D7	X2	MSB	"NUMBER OR PUNCTUATION"

```
X0D8    70      LSB                                  "NUMBER OR PUNCTUATION"
^       ^
|       +---- The least significant nibble of the MSB and the entire LSB address locations will vary
|             with a different set of exception words; X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
|
+----- YOUR EXCEPTION-WORD EPROM CAN RESIDE ANYWHERE FROM 1000H TO E000H PROVIDING IT BEGINS ON
       A 4K BOUNDARY WHERE X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.  (The least significant 3 nibbles
       of the address must remain as shown.)
```

TABLE 4.   SAMPLE OF ASSEMBLED ENCODED EXCEPTION-WORDS

```
X193 13 6E 24  AA:DB 19,110,36,185,19,90,11,1,33,19,0,18,15,0,1,65,34,39,20,141
X196 B9 13 5A 0B 01 21 13 00 12 0F
X1A0 00 01 41 22 27 14 8D
               ;<[ANDY]<=[AE NN1 PA2 DD2 IY PA1 DH1 AX PA1 PA2 GG3 RR2 EY TT2]  ANDY-THE-GREAT
X1A7 FF           DB 255
               ;******************************************************************************
X1A8 FF        BB:DB 255
               ;******************************************************************************
X1A9 13 61 B0  C:DB 19,97,176,33,106,20,137;<[CAP]A=[KK1 EY PP]                      CAPABILITY
X1AC 21 6A 14 89
X1B0 FF           DB 255
               ;******************************************************************************
X1B1 FF        D: DB 255
               ;******************************************************************************
X1B2 FF        E: DB 255
               ;******************************************************************************
X1B3 FF        F: DB 255
               ;******************************************************************************
X1B4 13 E9 13  G: DB 19,233,19,74,7,11,51,62,0,12,11,55,13,39,31,16,7,11,2,141
X1B7 4A 07 0B 33 3E 00 0C 0B 37 0D
X1C1 27 1F 10 07 0B 02 8D
               ;<[GI]<=[JH EH NN1 ER1 EL PA1 IH NN1 SS TT2 RR2 UW2 MM EH NN1 PA3 TT2]
               ;GENERAL INSTRUMENT
X1C8 13 E9 2D     DB 19,233,45,33,41,44,19,74,7,11,51,62,0,12,11,55,13,39,31,16,7,11,2,141
X1CB 21 29 2C 13 4A 07 0B 33 3E 00
X1D5 0C 0B 37 0D 27 1F 10 07 0B 02
X1DF 8D
               ;<[GI]MAIL<=[JH EH NN1 ER1 EL PA1 IH NN1 SS TT2 RR2 UW2 MM EH NN1 PA3 TT2]
X1E0 FF           DB 255
               ;******************************************************************************
X1E1 FF        H: DB 255
               ;******************************************************************************
X1E2 13 E4 13  I: DB 19,228,19,70,0,33,7,11,2,13,12,40,12,2,42,20,37,15,139;<[ID]<=[AY PA1
X1E5 46 00 21 07 0B 02 0D 0C 28 0C
X1EF 02 2A 14 25 0F 8B
               ;DD2 EH NN1 PA3 TT2 IH FF IH PA3 KK1 EY SH AX NN1]                 IDENTIFICATION
X1F5 13 73 2C     DB  19,115,44,165,19,70,1,190;<[ISLE]<=[AY PA2 EL]                         ISLE
X1F8 A5 13 46 01 BE
X1FD 13 73 2C     DB 19,115,44,33,46,164,19,70,0,45,26,11,1,21,1;<[ISLAND]<=[AYPA2ELAENN1DD1]
X200 21 2E A4 13 46 00 2D 1A 0B 01
X20A 15 01
X20C FF           DB 255
               ;******************************************************************************
X20D FF        J: DB 255
               ;******************************************************************************
X20E FF        K: DB 255
               ;******************************************************************************
X20F 13 69 36  L: DB 19,105,54,37,164,19,109,12,35,3,149;<[LIVED]<=[LL IH VV PA4 DD1]    LIVED
X212 25 A4 13 6D 0C 23 03 95
X21A FF           DB 255
               ;******************************************************************************
X21B FF        M: DB 255
               ;******************************************************************************
```

```
X21C FF         N: DB 255
                ;***************************************************************************
X21D FF         O: DB 255
                ;***************************************************************************
X21E 13 75 32  P: DB 19,117,50,48,47,51,165,19,73,51,9,15,55,183;<[PURPOSE]<=[PPER1PPAXSSSS]
X221 30 2F 33 A5 13 49 33 09 0F 37
X22B B7
X22C FF            DB 255
                ;***************************************************************************
X22D FF         Q: DB 255
                ;***************************************************************************
X22E FF         R: DB 255
                ;***************************************************************************
X22F FF         S: DB 255
                ;***************************************************************************
X230 13 6F 34  T: DB 19,111,52,33,172,19,77,53,13,0,15,190;<[TOTAL]<=[TT2 OW TT1 PA1 AX EL]
X233 21 AC 13 4D 35 0D 00 0F BE
X23C FF            DB 255
                ;***************************************************************************
X23D 13 73 25  U:DB 19,115,37,50,41,164,19,113,22,43,51,1,6,0,33,7,11,2,13,12,40,12,2,42,20,37
X240 32 29 A4 13 71 16 2B 33 01 06
X24A 00 21 07 0B 02 0D 0C 28 0C 02
X254 2A 14 25
X257 0F 8B         DB 15,139;<[USERID]<=[YY1 UW1 ZZ ER1 PA1 AY PA1 DD2 EH NN1 PA3 TT2 IH FF IH
                          ;PA3 KK1 EY SH AX NN1]
X259 FF            DB 255
                ;***************************************************************************
X25A FF         V: DB 255
                ;***************************************************************************
X25B 13 65 07  W: DB 19,101,7,50,165,110,19,180        ;<[WE'RE]=[WW IY ER2]             WE'RE
X25E 32 A5 6E 13 B4
X263 FF            DB 255
                ;***************************************************************************
X264 FF         X: DB 255
                ;***************************************************************************
X265 13 6F 35  Y: DB 19,111,53,7,50,165,19,89,186      ;<[YOU'RE]<=[YY2 OR]             YOU'RE
X268 07 32 A5 13 59 BA
X26E FF            DB 255
                ;***************************************************************************
X26F FF         Z: DB 255
                ;***************************************************************************
X270 13 CF 13  NUMORPUN: DB 19,207,19,89,58,1,16,7,55,55,12,1,10,0,2,42,26
X273 59 3A 01 10 07 37 37 0C 01 0A
X27D 00 02 2A 1A
X281 0B 01 3F   DB 11,1,63,19,0,55,55,2,9,53,2,42,7,11,0,46,12,29,0,18,15,0,2,13,15,2,50
X284 13 00 37 37 02 09 35 02 2A 07
X28E 0B 00 2E 0C 1D 00 12 0F 00 02
X298 0D 0F 02 32
X29C 00 0F 23   DB 0,15,35,0,20,0,2,42,19,128
X29F 00 14 00 02 2A 13 80
                ;<[/]<=[YY1 OR PA2 MM EH SS SS IH PA2 JH PA1 PA3 KK1 AE NN1 PA1 PA2 BB2 IY
                ;PA1 SS SS PA3 PP OW PA3 KK1 EH NN1 PA1 WW IH TH PA1 DH1 AX PA1 PA3
                ;TT2 AX PA3 CH PA1 AX VV PA1 EY PA1 PA2 KK1 EY PA1]YOU'RE MESSAGE CAN BE SPOKEN
                ;WITH THE TOUCH OF A KEY
                ;---------------------------------------------------------------------------
```

```
X2A6 C6 5A 0B     DB 198,90,11,21,128                   ;[&]=[AE NN1 DD1 PA1]              AND
X2A9 15 80
X2AB FF           DB 255                                ;MUST END EACH CATEGORY WITH [ ].
^
¦
+---- WHERE X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.  (The least significant 3 nibbles of the address
      will vary with a different set of exception words.)
```

TABLE 5. SERIAL PORT MODE REGISTER

```
+-MSB---------------------------------------LSB-+
|  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
+-----------------------------------------------+
|STOP | SIO |PEVEN| PEN |CHAR1|CHAR0| COMM|MULTI|
+-----------------------------------------------+
   |     |     |     |     |     |     |     |
   |     |     |     |     +-----+     |     | 0=MOTOROLA PROTOCOL
   |     |     |     |        |        |     +-1=INTEL PROTOCOL (see note 5.1)
   |     |     |     |        |        |
   |     |     |     |        |        | 0=ISOSYNCHRONOUS COMMUNICATION
   |     |     |     |        |        +-1=ASYNCHRONOUS COMMUNICATION
   |     |     |     |        |            (see note 5.1)
   |     |     |     |        |
   |     |     |     |        | 00=5 BITS/CHARACTER
   |     |     |     |        | 01=6 BITS/CHARACTER  (see note 5.1)
   |     |     |     |        | 10=7 BITS/CHARACTER
   |     |     |     |        +-11=8 BITS/CHARACTER
   |     |     |     |
   |     |     |     | 1=PARITY ENABLED
   |     |     |     +-0=PARITY DISABLED
   |     |     |
   |     |     | 1=EVEN PARITY
   |     |     +-0=ODD PARITY
   |     |
   |     | 0=SERIAL I/O MODE
   |     +-1= COMMUNICATION MODE  (see note 5.1)
   |
   | 0=ONE STOP BIT
   +-1=TWO STOP BITS
```

NOTE 5.1 FOR A COMPLETE DESCRIPTION REFER TO PIC7001 Data Sheet, DS33001A.

FOR TYPICAL APPLICATIONS USE: MOTOROLA PROTOCOL, ASYNCHRONOUS COMMUNICATION, 7 BITS/CHARACTER, and COMMUNICATION MODE; THE NUMBER OF STOP BITS AND PARITY MODE REMAIN UP TO THE USER.

TABLE 6. SERIAL PORT CONTROL REGISTER / TIMER REGISTER

$$\text{Asynchronous Baud Rate} = \frac{2.5\ \text{MHz}}{64(PL + 1)(TL + 1)}$$

$$\text{Isosynchronous Baud Rate} = \frac{2.5\ \text{MHz}}{4(PL + 1)(TL + 1)}$$

where: PL = prescale latch value
TL = timer latch value

Example: To program the serial port to operate at 300 baud in the asynchronous mode, the prescaler value is set to 0, and the timer latch value to 81H.

```
+-MSB---------------------------------------LSB-+
¦  7  ¦  6  ¦  5  ¦  4  ¦  3  ¦  2  ¦  1  ¦  0  ¦
+-----------------------------------------------+
¦  X  ¦  1  ¦  0  ¦  0  ¦  0  ¦  0  ¦ PRE1¦ PRE0¦
+-----------------------------------------------+
                                       ¦     ¦
                                       +-----+
                                          ¦
                                          +--2 BIT PRESCALE LATCH VALUE (PL)


+-MSB---------------------------------------LSB-+
¦  7  ¦  6  ¦  5  ¦  4  ¦  3  ¦  2  ¦  1  ¦  0  ¦
+-----------------------------------------------+
¦             TIMER LATCH VALUE (TL)            ¦
+-----------------------------------------------+
```

TABLE 7. ASCII CHARACTER SET ENCODED VALUES

LETTER	ENCODED VALUE (shown in Hexadecimal).
A	21
B	22
C	23
D	24
E	25
F	26
G	27
H	28
I	29
J	2A
K	2B
L	2C
M	2D
N	2E
O	2F
P	30
Q	31
R	32
S	33
T	34
U	35
V	36
W	37
X	38
Y	39
Z	3A

NUMBER	ENCODED VALUE
0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19

PUNCTUATION	ENCODED VALUE
SPACE	00
!	01
"	02
#	03
$	04
%	05
&	06
'	07
(	08
)	09
*	0A
+	0B
,	0C
-	0D
.	0E
/	0F
:	1A
;	1B
<	1C
=	1D
>	1E
?	1F
@	20
[	3B
\	3C
]	3D
^	3E
_	3F
`	40
{	5B
|	5C
}	5D
~	5E

TABLE 8. ALLOPHONE ADDRESS ENCODED VALUES (shown in Hexadecimal).

ENCODED VALUE	ALLOPHONE	SAMPLE WORD	DURATION (ms)	ENCODED VALUE	ALLOPHONE	SAMPLE WORD	DURATION (ms)
00	PA1	PAUSE	10	20	AW	OUt	250
01	PA2	PAUSE	30	21	DD2	Do	80
02	PA3	PAUSE	50	22	GG3	wiG	120
03	PA4	PAUSE	100	23	VV	Vest	130
04	PA5	PAUSE	200	24	GG1	Guest	80
05	OY	bOY	290	25	SH	SHip	120
06	AY	skY	170	26	ZH	aZUre	130
07	EH	End	50	27	RR2	bRain	80
08	KK3	Comb	80	28	FF	Food	110
09	PP	Pow	150	29	KK2	sKy	140
0A	JH	dodGe	400	2A	KK1	Can't	120
0B	NN1	thiN	170	2B	ZZ	Zoo	150
0C	IH	sIt	50	2C	NG	aNchor	200
0D	TT2	To	100	2D	LL	Lake	80
0E	RR1	Rural	130	2E	WW	Wool	140
0F	AX	sUcceed	50	2F	XR	repaIR	250
10	MM	Milk	180	30	WH	WHig	150
11	TT1	parT	80	31	YY1	Yes	90
12	DH1	THey	140	32	CH	CHurch	150
13	IY	sEE	170	33	ER1	fIR	110
14	EY	bEIge	200	34	ER2	fIR	210
15	DD1	coulD	50	35	OW	bEAU	170
16	UW1	tO	60	36	DH2	THey	180
17	AO	OUght	70	37	SS	veSt	60
18	AA	hOt	60	38	NN2	No	140
19	YY2	Yes	130	39	HH2	Hoe	130
1A	AE	hAt	80	3A	OR	stORe	240
1B	HH1	He	90	3B	AR	alARm	200
1C	BB1	Business	40	3C	YR	clEAR	250
1D	TH	THin	130	3D	GG2	Got	80
1E	UH	bOOk	70	3E	EL	saddLE	140
1F	UW2	fOOd	170	3F	BB2	Business	60

TABLE 9. PARALLEL PORT TIMING REQUIREMENTS:

SETUP TIME, BEFORE DATA CLOCK LO TO HI TRANSITION: MIN. 20 nS.
HOLD TIME, BEFORE DATA CLOCK LO TO HI TRANSITION: MIN. 10 nS.
WIDTH OF CLOCK LO: MIN. 500 nS.

HOLD OFF TIME, FROM DATA STROBE HI TO LO TO HI, UNTIL NEXT
DATA STROBE HI TO LO: MIN. 450 uS.

NOTE: The addition of an 74LS74 Flip-Flop as shown on the schematic can be used for parallel port latch handshaking using the Active-LO LATCH-BUSY* output. LATCH-BUSY* is LO when the latch is full, and it is HI when the latch is empty and available for the next character to be strobed in.

```
APPENDIX-A
*************************************************************************
*      EXCEPTION-WORD EPROM MAP     (For use without USER eprom present)      *
*************************************************************************
  NOTE:  ENCAPSULATED SEQUENCES ARE USER-DEFINED, REFER TABLES 2,3, AND 4.
        ¦0 ¦1 ¦2 ¦3 ¦4 ¦5 ¦6 ¦7 ¦8 ¦9 ¦A ¦B ¦C ¦D ¦E ¦F
                                   +--------------------+
   X000  80 48 28 58 85 E0 35 E0 31¦FF FF FF FF FF FF FF¦<--sample
        +--------------------------+                    ¦   NEW PARAMETERS.
   X010 ¦FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF¦   (see table 2).
        ¦        +--------------------------------------+
   X020 ¦FF FF FF¦1E 1F 20 21 28 29 24 25 22 23 2A 2B 26 <--NEW PARAMETER
        +--------+                                          INITIALIZATION
   X030  27 2C 2D 2E 2F 32 33 34 35 36 E0 65 78 02 31 8E    ROUTINE.
   X040  F1 43 C5 AA X0 09 2D FF E2 1E B8 AA X0 23 D5 12
   X050  D0 13 B9 9B 13 C3 AA X0 09 2D FF E2 0B B8 AA X0    The MSnibble of the following locations
   X060  23 D5 12 D0 13 B9 9B 13 5D 16 E6 E9 C3 AA X0 09    from the NEW PARAMETER INITIALIZATION
   X070  2D FF E2 14 A2 40 11 82 11 A2 15 11 C3 AA X0 09    ROUTINE are user defined also:
   X080  82 15 C3 AA X0 09 82 14 98 29 03 98 2B 07 22 20    X044,X04C,X057,X05F,X06E,X07E,and X084;
   X090  9B 03 8E F7 2B 98 03 05 98 07 09 98 03 19 8C F1    where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
                 +--------------------------------------+
   X0A0  00 E0 36¦X1 93 X1 A8 X1 A9 X1 B1 X1 B2 X1 B3 X1¦<--sample
        +--------+                                      ¦   ALPHABETIZED
   X0B0 ¦B4 X1 E1 X1 E2 X2 0D X2 0E X2 0F X2 1B X2 1C X2¦   EXCEPTION-WORD
   X0C0 ¦1D X2 1E X2 2D X2 2E X2 2F X2 30 X2 3D X2 5A X2¦   INDEX, where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
        ¦                          +--------------------+   (see table 3).
   X0D0 ¦5B X2 64 X2 65 X2 6F X2 70¦D8 02 D8 03 98 03 11 <--EXCEPTION-WORD
        +--------------------------+                        ROUTINE.
   X0E0  8E F7 4B 8E F7 0F 77 01 0A 05 74 80 0B E0 03 73
   X0F0  7F 0B 8E F3 AF 76 20 0A 0E 52 34 AA X0 A3 D0 14    The MSnibble of the following locations
   X100  AA X0 A4 D0 15 E0 0F C5 2A 41 2C 02 AA X0 A3 D0    from the EXCEPTION-WORD ROUTINE are user
   X110  14 AA X0 A4 D0 15 52 01 8E F4 88 8E F4 C2 76 10    defined also:  X0FC,X101,X10D,and X112;
   X120  0A 4D 2D FF E2 60 98 11 1D 73 BF 0A 8E F5 64 76    where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
   X130  10 0A 3C 8E F4 7E 74 40 0A 8E F5 64 76 10 0A 42
   X140  48 37 34 79 00 33 D5 37 73 FD 0B 52 02 8E F4 88
   X150  8E F4 9E 98 0F 03 98 03 11 8E F7 4B 77 80 0B 0A
   X160  DB 39 8E F3 47 C9 C9 8C F1 36 C9 C9 8C F3 F4 D3
   X170  15 E7 02 D3 14 52 02 8E F4 8B 72 01 37 73 FD 0B
   X180  E0 99 52 03 E0 F1 D9 03 D9 02 D5 37 73 FD 0B 8C
                 +--------------------------------------+
   X190  F3 EE FF¦13 6E 24 B9 13 5A 0B 01 21 13 00 12 0F¦<--sample
        +--------+                                      ¦   ENCODED
   X1A0 ¦00 01 41 22 27 14 8D FF FF 13 61 B0 21 6A 14 89¦   EXCEPTION-WORDS.
   X1B0 ¦FF FF FF FF 13 E9 13 4A 07 0B 33 3E 00 0C 0B 37¦   (see table 4).
   X1C0 ¦0D 27 1F 10 07 0B 02 8D 13 E9 2D 21 29 2C 13 4A¦
   X1D0 ¦07 0B 33 3E 00 0C 0B 37 0D 27 1F 10 07 0B 02 8D¦
   X1E0 ¦FF FF 13 E4 13 46 00 21 07 0B 02 0D 0C 2B 0C 02¦
   X1F0 ¦2A 14 25 0F 8B 13 73 2C A5 13 46 01 BE 13 73 2C¦ (see APPENDIX-C
   X200 ¦21 2E A4 13 46 00 2D 1A 0B 01 15 01 FF FF FF 13¦  for discussion
   X210 ¦69 36 25 A4 13 6D 0C 23 03 95 FF FF FF FF 13 75¦  of encoding scheme.)
   X220 ¦32 30 2F 33 A5 13 49 33 09 0F 37 B7 FF FF FF FF¦
   X230 ¦13 6F 34 21 AC 13 4D 35 0D 00 0F BE FF 13 73 25¦
   X240 ¦32 29 A4 13 71 16 2B 33 01 06 00 21 07 0B 02 0D¦
   X250 ¦0C 28 0C 02 2A 14 25 0F 8B FF FF 13 65 07 32 A5¦
   X260 ¦6E 13 B4 FF FF 13 6F 35 07 32 A5 13 59 BA FF FF¦
```

```
X270 ¦13 CF 13 59 3A 01 10 07 37 37 0C 01 0A 00 02 2A¦
X280 ¦1A 0B 01 3F 13 00 37 37 02 09 35 02 2A 07 0B 00¦
X290 ¦2E 0C 1D 00 12 0F 00 02 0D 0F 02 32 00 0F 23 00¦
X2A0 ¦14 00 02 2A 13 80 C6 5A 0B 15 80 FF            ¦
^    ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
^    ¦0 ¦1 ¦2 ¦3 ¦4 ¦5 ¦6 ¦7 ¦8 ¦9 ¦A ¦B ¦C ¦D ¦E ¦F
+----YOUR EXCEPTION-WORD EPROM CAN RESIDE ANYWHERE FROM
     1000H TO E000H, PROVIDING IT BEGINS ON A 4K BOUNDARY WHERE X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
```

```
APPENDIX-B
*******************************************************************************
*      EXCEPTION-WORD EPROM MAP        (For use with USER eprom present)      *
*******************************************************************************
   NOTE:  ENCAPSULATED SEQUENCES ARE USER-DEFINED, REFER TABLES 2,3, AND 4.
        ¦0 ¦1 ¦2 ¦3 ¦4 ¦5 ¦6 ¦7 ¦8 ¦9 ¦A ¦B ¦C ¦D ¦E ¦F
                                   +--------------------+
   X000  80 48 28 58 85 E0 35 E0 31¦FF FF FF FF FF FF FF¦<--sample
        +--------------------------+                    ¦  NEW PARAMETERS.
   X010 ¦FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF¦  (see table 2).
        ¦        +--------------------------------------+
   X020 ¦FF FF FF¦1E 1F 20 21 28 29 24 25 22 23 2A 2B 26 <--NEW PARAMETER
        +--------+                                         INITIALIZATION
   X030  27 2C 2D 2E 2F 32 33 34 35 36 E0 65 78 02 31 8E   ROUTINE.
   X040  F1 43 C5 AA X0 09 2D FF E2 1E B8 AA X0 23 D5 12   The MSnibble of the following locations
   X050  D0 13 B9 9B 13 C3 AA X0 09 2D FF E2 0B B8 AA X0   from the NEW PARAMETER INITIALIZATION
   X060  23 D5 12 D0 13 B9 9B 13 5D 16 E6 E9 C3 AA X0 09   ROUTINE are user defined also:
   X070  2D FF E2 14 A2 40 11 82 11 A2 15 11 C3 AA X0 09   X044,X04C,X057,X05F,X06E,X07E,and X084;
   X080  82 15 C3 AA X0 09 82 14 98 29 03 98 2B 07 22 20   where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
   X090  9B 03 8E F7 2B 98 03 05 98 07 09 98 03 19 8C MS<--(see note 1 below).
                 +--------------------------------------+
   X0A0  LS E0 36¦X1 93 X1 A8 X1 A9 X1 B1 X1 B2 X1 B3 X1¦<--sample
        +--------+                                      ¦  ALPHABETIZED
   X0B0 ¦B4 X1 E1 X1 E2 X2 0D X2 0E X2 0F X2 1B X2 1C X2¦  EXCEPTION-WORD
   X0C0 ¦1D X2 1E X2 2D X2 2E X2 2F X2 30 X2 3D X2 5A X2¦  INDEX, where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
        ¦                          +--------------------+  (see table 3).
   X0D0 ¦5B 82 64 B2 65 82 6F 82 70¦D8 02 D8 03 98 03 11 <--EXCEPTION-WORD
        +--------------------------+                       ROUTINE.
   X0E0  8E F7 4B 8E F7 0F 77 01 0A 05 74 80 0B E0 03 73
   X0F0  7F 0B 8E F3 AF 76 20 0A 0E 52 34 AA 80 A3 D0 14   The MSnibble of the following locations
   X100  AA 80 A4 D0 15 E0 0F C5 2A 41 2C 02 AA 80 A3 D0   from the EXCEPTION-WORD ROUTINE are user
   X110  14 AA 80 A4 D0 15 52 01 8E F4 88 8E F4 C2 76 10   defined also:  X0FC,X101,X10D,and X112;
   X120  0A 4D 2D FF E2 60 98 11 1D 73 BF 0A 8E F5 64 76   where X=1,2,3,4,5,6,7,8,9,A,B,C,D,or E.
   X130  10 0A 3C 8E F4 7E 74 40 0A 8E F5 64 76 10 0A 42
   X140  48 37 34 79 00 33 D5 37 73 FD 0B 52 02 8E F4 88
   X150  8E F4 9E 98 0F 03 98 03 11 8E F7 4B 77 80 0B 0A
   X160  DB 39 8E F3 47 C9 C9 8C F1 36 C9 C9 8C F3 F4 D3
   X170  15 E7 02 D3 14 52 02 8E F4 88 72 01 37 73 FD 0B
   X180  E0 99 52 03 E0 F1 D9 03 D9 02 D5 37 73 FD 0B 8C
                 +--------------------------------------+
   X190  F3 EE FF¦13 6E 24 B9 13 5A 0B 01 21 13 00 12 0F¦<--sample
        +--------+                                      ¦  ENCODED
   X1A0 ¦00 01 41 22 27 14 8D FF FF 13 61 B0 21 6A 14 89¦  EXCEPTION-WORDS
   X1B0 ¦FF FF FF FF 13 E9 13 4A 07 0B 33 3E 00 0C 0B 37¦  (see table 4). .
   X1C0 ¦0D 27 1F 10 07 0B 02 8D 13 E9 2D 21 29 2C 13 4A¦
   X1D0 ¦07 0B 33 3E 00 0C 0B 37 0D 27 1F 10 07 0B 02 8D¦
   X1E0 ¦FF FF 13 E4 13 46 00 21 07 0B 02 0D 0C 28 0C 02¦
   X1F0 ¦2A 14 25 0F 8B 13 73 2C A5 13 46 01 BE 13 73 2C¦
   X200 ¦21 2E A4 13 46 00 2D 1A 0B 01 15 01 FF FF FF 13¦  (see APPENDIX-C
   X210 ¦69 36 25 A4 13 6D 0C 23 03 95 FF FF FF FF 13 75¦  for discussion
   X220 ¦32 30 2F 33 A5 13 49 33 09 0F 37 B7 FF FF FF FF¦  of encoding scheme).
   X230 ¦13 6F 34 21 AC 13 4D 35 0D 00 0F BE FF 13 73 25¦
   X240 ¦32 29 A4 13 71 16 2B 33 01 06 00 21 07 0B 02 0D¦
   X250 ¦0C 28 0C 02 2A 14 25 0F 8B FF FF 13 65 07 32 A5¦
   X260 ¦6E 13 B4 FF FF 13 6F 35 07 32 A5 13 59 BA FF FF¦
   X270 |13 CF 13 59 3A 01 10 07 37 37 0C 01 0A 00 02 2A|
   X280 |1A 0B 01 3F 13 00 37 37 02 09 35 02 2A 07 0B 00|
   X290 |2E 0C 1D 00 12 0F 00 02 0D 0F 02 32 00 0F 23 00|
   X2A0 |14 00 02 2A 13 80 C6 5A 0B 15 80 FF            |
```

NOTE 1. APPENDIX-B is the same as APPENDIX-A, except for two address. These are X09F and X0A0 (MSB and LSB respectively, labeled MS and LS above). Place the origin of the MAIN-CONTROL-PROGRAM (see APPENDIX-F) in these locations so that program control will transfer to the user's code at the appropriate time.

APPENDIX-C

```
********************************************************************************
*                       EXCEPTION-WORD ENCODING SCHEME                         *
********************************************************************************
```

To store a unique word or symbol and its corresponding allophone address string in an efficient and flexible manner, the following encoding format was derived:

<[encoded word or symbol]< = [encoded allophone address(es)]

```
where:  < equals 13H.
        [ equals 40H.
        ] equals 80H.
```

The first and last byte is 13H. This informs the code-to-speech algorithm that the word or symbol is not a prefix or suffix.
If the word or symbol is an individual letter, then the representation of it between the brackets is an FFH; this includes the value of the left and right brackets. (If it is a number or punctuation, then it is represented by its value from Table-7 plus the value of the left and right brackets.)

Otherwise:
(1) The first letter in the word or symbol is always to be ignored; this does not apply to numbers or punctuations.
(2) The next letter in the word is represented by the value of the letter from TABLE-7, plus the value of the left bracket "[" which is 40H.
(3) The following letter(s), if and only if it is not the last letter in the word or symbol, is represented solely by its value from TABLE-7.
(4) The last letter in the word or symbol is represented by the value of the letter from TABLE-7, plus the value of the right bracket "]" which is 80H.

The allophone address string is encoded in a similar manner:

If only one allophone is used for the pronunciation, it is represented by its value from TABLE-8, plus the value of the right "[" and left "]" brackets which are 40H and 80H respectively.

Otherwise:

(1) The first allophone is represented by its value from TABLE-8, plus the value of the left bracket "[" which is 40H.
(2) The following allophone(s), if and only if it is not the last allophone in the string is represented by its value from TABLE-8.
(3) The last allophone is represented by its value from TABLE-8 plus the value of the right bracket "]" which is 80H.

```
Example:  To encode "Au" to pronounce as "GOLD"
          <[Au]< = [GG2 OW LL DD1]
         13,F5,13,  7D, 35,2D,95 <--This line is ready to store in EXCEPTION-WORD
             ^                      eprom under the "A" category.
             :                      (The encoded string is shown in Hexadecimal notation.)
             :
             +--Remember, throw away the first letter (in this case an "A"),
                then find the value of the next letter in TABLE-7 and add
                40H plus 80H to it so as to represent the left "[" and right "]" brackets.
```

APPENDIX-D

```
******************************************************************************
*      USER EPROM MAP        (For use without EXCEPTION-WORD eprom)          *
******************************************************************************
   NOTE:  ENCAPSULATED SEQUENCES ARE USER-DEFINED, REFER TABLES 2,3, AND 4.
        ¦0 ¦1 ¦2 ¦3 ¦4 ¦5 ¦6 ¦7 ¦8 ¦9 ¦A ¦B ¦C ¦D ¦E ¦F
        ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦
                                   +--------------------+
   X000  B0 48 28 58 85 E0 35 E0 31¦FF FF FF FF FF FF FF¦<--sample
        +--------------------------+                    ¦   NEW PARAMETERS.
   X010 ¦FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF¦   (see table 1).
        ¦        +--------------------------------------+
   X020 ¦FF FF FF¦1E 1F 20 21 28 29 24 25 22 23 2A 2B 26 <--NEW PARAMETER
        +--------+                                          INITIALIZATION
   X030  27 2C 2D 2E 2F 32 33 34 35 36 E0 65 78 02 31 8E    ROUTINE.
   X040  F1 43 C5 AA X0 09 2D FF E2 1E B8 AA X0 23 D5 12    The MSnibble of the following locations
   X050  D0 13 B9 9B 13 C3 AA X0 09 2D FF E2 0B B8 AA X0    from the NEW PARAMETER INITIALIZATION
   X060  23 D5 12 D0 13 B9 9B 13 5D 16 E6 E9 C3 AA X0 09    ROUTINE are user defined also:
   X070  2D FF E2 14 A2 40 11 82 11 A2 15 11 C3 AA X0 09    X044,X04C,X057,X05F,X06E,X07E,and X084;
   X080  82 15 C3 AA X0 09 82 14 98 29 03 98 2B 07 22 20    where X=1,2,3,4,5,6,7,8,9,A,B,C,D, or E.
   X090  9B 03 8E F7 2B 98 03 05 98 07 09 98 03 19 8C MS<--(see note 1 below).
   X0A0  LS 8C F3 F4 +----------------------------------+
         +-----------+ User code may start at X0A4,     ¦
         ¦ but must contain the MAIN-CONTROL-PROGRAM    ¦
         ¦ somewhere within, refer to APPENDIX-F.       ¦
```

NOTE 1.Place the immediate address of the origin of the MAIN-CONTROL-PROGRAM (see APPENDIX-F) in these locations; so that program control will transfer to the user's code at the appropriate time.

APPENDIX-E

```
*******************************************************************************
*      USER EPROM MAP           (For use with EXCEPTION-WORD eprom)           *
*******************************************************************************
```

NOTE 1. Contains no reserved locations, except for the MAIN-CONTROL-PROGRAM.
(See APPENDIX-F).

NOTE 2. A user's code does not have to reside in a second eprom (USER eprom).
It may reside in an unused portion of an EXCEPTION-WORD eprom which
is for use where "USER eprom is present". Refer APPENDIX-B.

APPENDIX-F

```
**************************************************************************
*  USER's MAIN CONTROL PROGRAM (for residency anywhere within USER eprom).  *
**************************************************************************
```

NOTE: ENCAPSULATED AREAS ARE USER DEFINED UNLESS OTHERWISE NOTED.

```
 F1AC =                   AUDIBLE  EQU 0F1ACH     ;<--+
 F3E7 =                   GISPEECH EQU 0F3E7H     ;<--+ THESE ARE ADDRESS VECTORS WITHIN
 F1E2 =                   SAVE     EQU 0F1E2H     ;<--+ THE MASKED CODE-TO-SPEECH ALGORITHM.
 F1F0 =                   ESCAPE   EQU 0F1F0H     ;<--+
 000B =                   F2       EQU R11
 0002 =                   F1HI     EQU R2
 0003 =                   F1LO     EQU R3
 0004 =                   R1HI     EQU R4
 0005 =                   R1LO     EQU R5
 0007 =                   F2LO     EQU R7
 0009 =                   R2LO     EQU R9
 0038 =                   WORDCNTH EQU R56
 0039 =                   WORDCNTL EQU R57
 0032 =                   BUFBVALU EQU R50
 0000 =                   IOCNT0   EQU P0
          +------------+         This is the origin of the Main Control P rogram which is defined by the
 9000     | ORG >9000  |         user.  Here  it is arbitrarily chosen to be 9000H.   Remember to  place
          +------------+         this immediate address in the "MS","LS" locations of the EXCEPTION-WORD
                                 eprom ("for use with USER eprom"),  see APPENDIX-B.  (MS=MSB=90 and the
                                 LS=LSB=00 in this example.)
 9000 8EF1AC    MESSAGE:  CALL @AUDIBLE
+--------------------THIS ENCAPSULATED AREA IS NOT USER DEFINED--------------------+
|  The following two lines are placed here only if the USER code wishes to gain    |(The  XXXX  XXXX
|  access to the serial port.                                                       |here does not have
|XXXX XXXX                ANDP %>FE,IOCNT1      ;DISABLE INTERRUPT-4 (SERIAL PORT).|the same meaning as
|XXXX XXXX                ANDP %>FE,PORTB       ;SET BUSY* LO.                      |the X from the
+-----------------------------------------------------------------------------------+previous appendices
 9003 E00E                JMP ANYSTART                                               and tables.)
                        +------------+ ;THE BRANCH ADDRESS BELOW IS USER DEFINED.
 9005 8C9046    CRSTART: |BR @USERCODE| ;AFTER INITIALIZATION OR AFTER PROCESSING AND SPEAKING
                        +------------+ ;WHAT HAS BEEN LOADED INTO THE INPUT BUFFER, CONTROL
                                       ;TRANSFERS TO THE USER CODE VIA THIS BRANCH INSTRUCTION.
 9008 76010B07  SPEAK:    BTJO %>01,F2,ANYSTART
 900C 73EF0B              AND %>EF,F2
 900F 77100BFC  CRWAIT:   BTJZ %>10,F2,CRWAIT
 9013 4D0305    ANYSTART: CMP F1LO,R1LO
 9016 E607                JNE HOLEWORD
 9018 4D0204              CMP F1HI,R1HI
 901B E602                JNE HOLEWORD
 901D E0E6                JMP CRSTART
 901F 7D0038    HOLEWORD: CMP %>00,WORDCNTH
 9022 E605                JNE BFULTEST
 9024 7D0039              CMP %>00,WORDCNTL
 9027 E2F6                JEQ HOLEWORD
 9029 77080B09  BFULTEST: BTJZ %>08,F2,PROCESS
 902D 7D0132    LOCKUP:   CMP %>01,BUFBVALU
 9030 E211                JEQ ESC
 9032 76080BFC  BFULHOLD: BTJO %>08,F2,BFULHOLD
```

```
9036 8EF3E7    PROCESS:  CALL @GISPEECH
9039 4D0709    MAINROUT: CMP F2LO,R2LO
903C E2D5                JEQ ANYSTART
903E A40100              ORP %>01,IOCNTO
9041 E0D0                JMP ANYSTART
9043 8CF1F0     ESC:     BR @ESCAPE
              +---------------+
9046 00       | USERCODE: NOP |         ;FROM THIS POINT IT IS THE USER CODES RESPONSIBILITY
              +---------------+         ;TO EXECUTE ITS OWN CODE OR TO LOAD A CHARACTER STRING
                                        ;INTO THE INPUT BUFFER.
                                        ;THE TWO EXAMPLES SHOWN BELOW DEMONSTRATE THE
                                        ;RECOMMENDED SEQUENCE OF EVENTS FOR EACH MODE.
                                        ;MODE 1 IS USED WHEN THE USER CODE HAS PREVIOUSLY
                                        ;PREPARED THE CHARACTER STRING IT WISHES TO HAVE
                                        ;SPOKEN; MODE 2 IS USED WHEN THE USER CODE WISHES
                                        ;TO EXECUTE ANYTHING ELSE.
               +--------+               ;
               | MODE1: |               ;LOADING INPUT BUFFER OF CODE-TO-SPEECH ALGORITHM:
               +--------+               ;
                                        ;ACCUMULATOR AND STATUS REGISTER ARE TO BE SAVED.
                                        ;NO OTHER REGISTER IS TO BE MODIFIED.
                                        ;Loading a character string is accomplished
                                        ;by placing each character into the Accumulator and
                                        ;then using CALL @SAVE to load it into the input
                                        ;buffer.  Remember to end each word or phrase with a
                                        ;delimeter.  Restore the Accumulator and the Status Registers.
                                        ;Call @SPEAK to process and speak the word(s) or phrase(s)
                                        ;that were loaded.
                                        ;
```

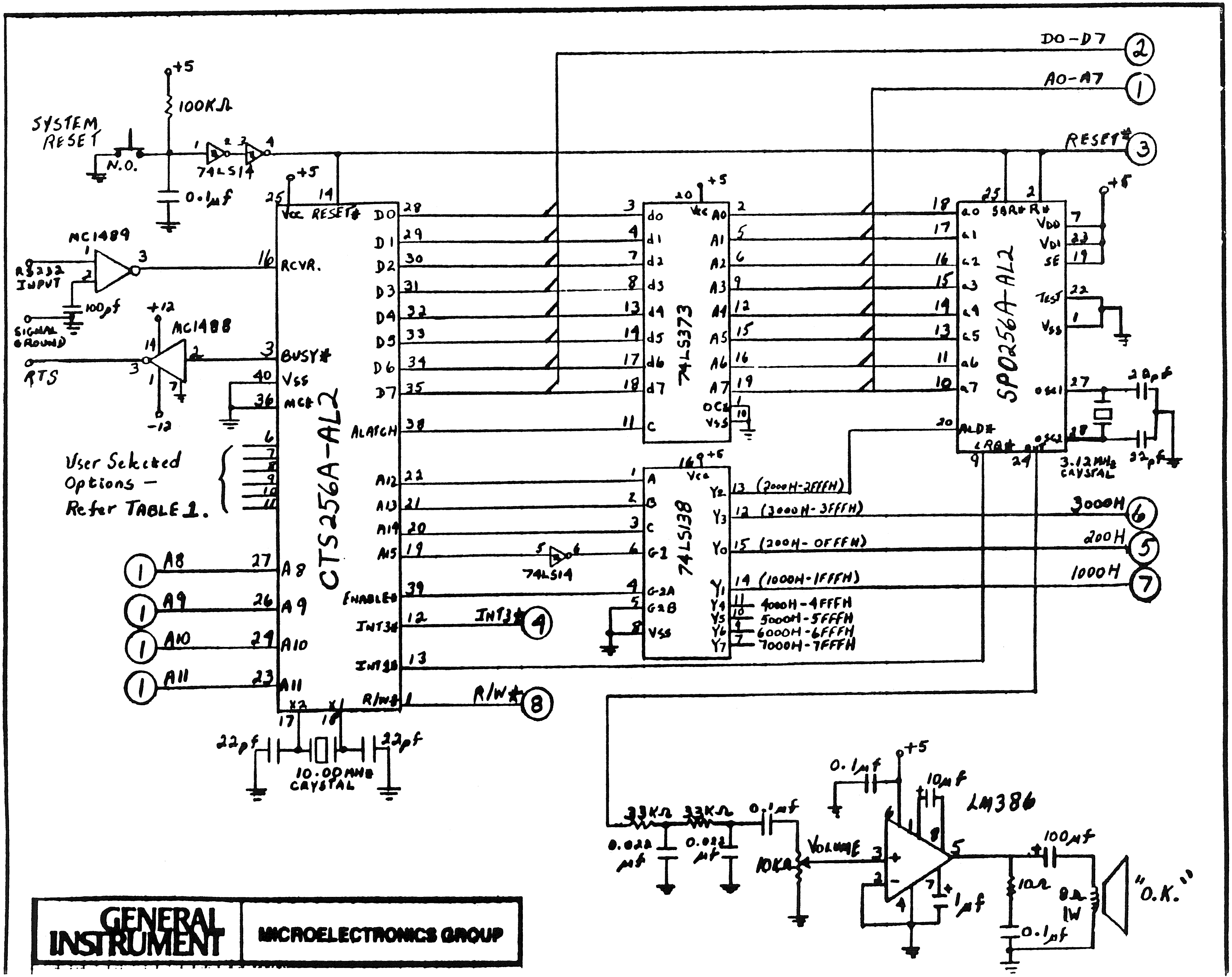
NOTE: Once "SPEAK" is initiated, control does not return to the USERCODE until the last word or phrase that is in the input buffer has been processed by the code-to-speech algorithm.

NOTE: Because of masked code-to-speech restrictions, the USER can not intercept input from the serial port while speech processing is in progress. During this interval, handshaking (BUSY*) shall hold off additional serial communication. This is accomplished by the two encapsulated lines shown above.

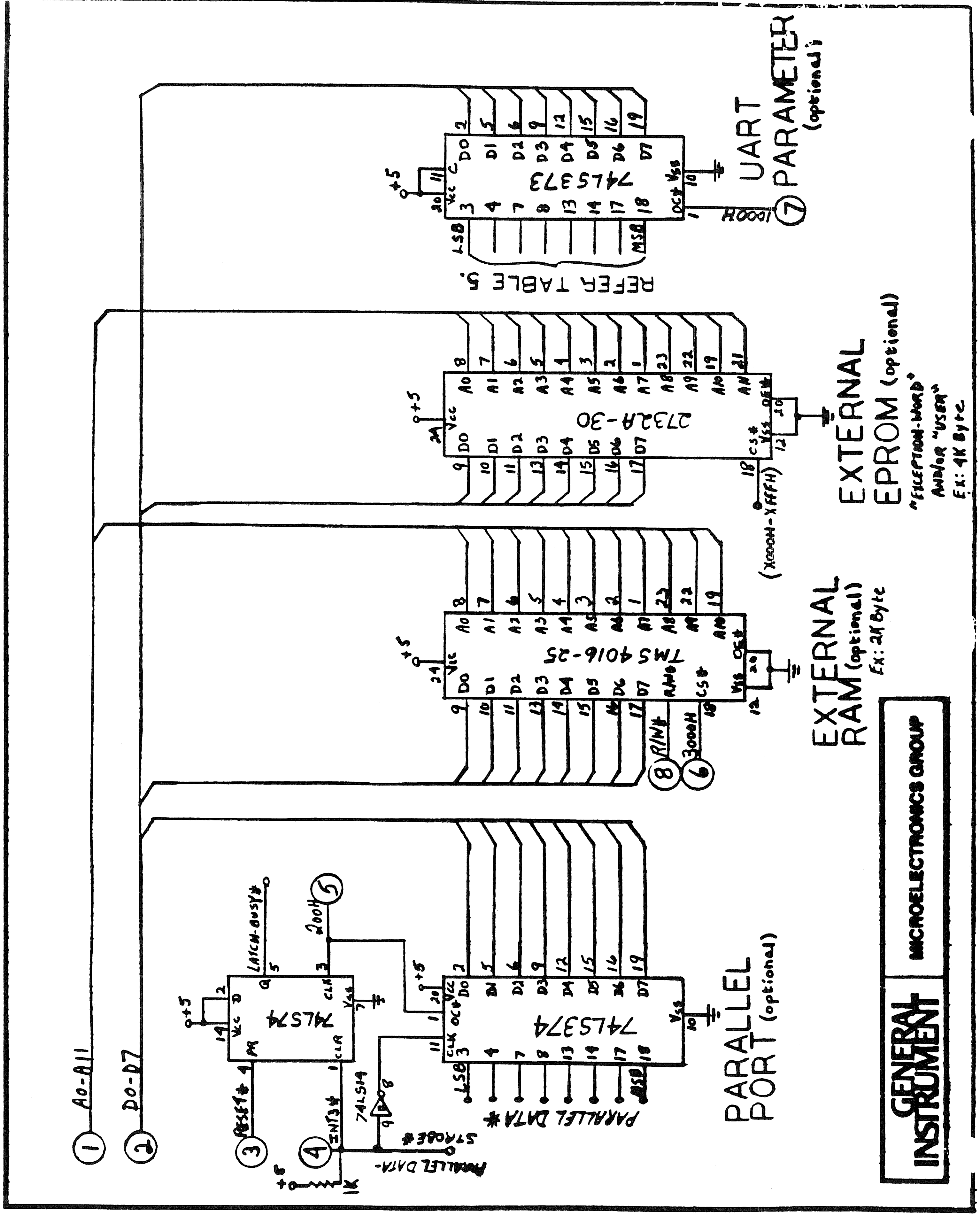
```
                                        ;
                                        ;THE FOLLOWING EXAMPLE WILL LOAD THE LETTER "A" AND
                                        ;SPEAK IT:
               +---------------------+  ;
9047 0E        |        PUSH ST      |  ;SAVE CONTENTS OF STATUS REGISTER.
9048 B8        |        PUSH A       |  ;SAVE CONTENTS OF ACCUMULATOR.
9049 2241      |        MOV %>41,A   |  ;MOVE 41H (which is ASCII "A") into the ACCUMULATOR.
904B 8EF1E2    |        CALL @SAVE   |  ;LOAD THE ASCII "A" INTO THE INPUT BUFFER.
904E 222D      |        MOV %>0D,A   |  ;MOVE 0DH (which is a carriage return).
9050 8EF1E2    |        CALL @SAVE   |  ;LOAD THE DELIMETER INTO THE INPUT BUFFER.
9053 B9        |        POP A        |  ;RECOVER CONTENTS OF ACCUMULATOR.
9054 08        |        POP ST       |  ;RECOVER CONTENTS OF STATUS REGISTER.
9055 8C9008    |        BR @SPEAK    |  ;TRANSFER CONTROL TO THE MAIN-CONTROL-PROGRAM WHICH
               +---------------------+  ;WILL ACCESS THE CODE-TO-SPEECH ALGORITHM; AFTER WHICH THE
                                        ;CONTROL WILL RETURN TO THE "BR @USERCODE" INSTRUCTION LOCATION.
               +-----------+            ;
9058 00        |MODE2:  NOP|            ;The following is the recommended
               +-----------+            ;sequence of events necessary for the user's code
                                        ;to do anything else (except for loading the input
                                        ;buffer as described under MODE 1.)
                                        ;
                                        ;SAVE STATUS REGISTER
                                        ;SAVE REGISTER 0 THRU 39H (EXTERNAL-RAM MODE), along with 3AH
                                        ;thru current Stack Pointer.
                                        ;OR, SAVE REGISTER 0 THRU 7FH (INTERNAL-RAM MODE).
                                        ;(DO NOT USE PUSH INSTRUCTIONS TO SAVE THE REGISTERS BECAUSE
                                        ;THE STACK IS NOT LARGE ENOUGH, INSTEAD
                                        ;BLOCK MOVE THE RESPECTIVE REGISTER CONTENTS INTO EXTERNAL-USER-
                                        ;-RAM.
                                        ;USER DEFINED CODE GOES HERE NEXT.
                                        ;
                                        ;(TO READ THE SERIAL PORT, SEE THE EXAMPLE SEQUENCE BELOW).
                                        ;
                                        ;THEN RECOVER RESPECTIVE REGISTERS.
                                        ;RECOVER STATUS REGISTER.
                                        ;BRANCH TO MODE 1, OR BRANCH TO OTHER USER CODE such as the
                                        ;example shown below for reading the serial port.

                                        ;The following is the recommended sequence of events necessary
                                        ;for the user's code to obtain input from the serial port:
                                        ;
LOOP:   ORP %>01,IOCNT1                 ;ENABLE INTERRUPT-4 (SERIAL PORT) BECAUSE WANT TO RECEIVE SERIAL
        ORP %>01,PORTB                  ;SET BUSY* HI.                                          INPUT.
        IDLE                            ;WAIT HERE FOR SERIAL INTERRUPT TO OCCUR AND TO BE SERVICED.
        ANDP %>FE,IOCNT1                ;DISABLE INTERRUPT-4 (SERIAL PORT).
       +---+                            ;
       ¦NOP¦                            ;THE CHARACTER RECEIVED BY SERIAL PORT IS IN THE ACCUMULATOR,
       +---+                            ;SO USER MAY EVALUATE IT HERE.
                                        ;
        MOV %>08,A                      ;LOAD A "BACKSPACE" INTO ACCUMULATOR IN ORDER TO TELL
        CALL @SAVE                      ;THE CODE-TO-SPEECH INPUT BUFFER TO IGNORE THE CHARACTER
                                        ;WHICH ARRIVED VIA THE SERIAL PORT.
       +---+                            ;
       ¦NOP¦                            ;IF USER WANTS ADDITIONAL CHARACTERS FROM THE SERIAL PORT TO EVALUATE:
       +---+                            ;JUMP TO LOOP TO WAIT FOR NEXT SERIAL PORT INTERRUPT (JMP LOOP).
                                        ;
                                        ;OTHERWISE:  ENABLE INTERRUPT-4 (ORP %>01,IOCNT1), SET BUSY* LO
                                        ;(ANDP %>FE,PORTB), THEN FALL THRU TO REST OF USER CODE.
```

```
                *******************************************************************************************
        NOTE:   * To successfully incorporate a USER program with the code-to-speech algorithm requires a *
                * thorough understanding of the concepts described in this application note, and an      *
                * in-depth working knowledge of PIC7001 assembly language.                                *
                *******************************************************************************************
```

END   OF   APPLICATION NOTE   AN-0505 REVISION D



SYSTEM RESET
N.O.
+5
100KΩ
0.1μf
74LS14
MC1489
RS232 INPUT
100pf
SIGNAL GROUND
+12
MC1488
RTS
-12
User Selected Options - Refer TABLE 1.
A8
A9
A10
A11
CTS256A-AL2
Vcc
RESET#
RCVR.
BUSY#
Vss
MC#
D0
D1
D2
D3
D4
D5
D6
D7
ALATCH
A12
A13
A14
A15
ENABLE#
INT3#
INT1#
R/W#
X1
X2
22pf
10.00MHz CRYSTAL
74LS373
74LS138
(2000H-2FFFH)
(3000H-3FFFH)
(200H-0FFFH)
(1000H-1FFFH)
4000H-4FFFH
5000H-5FFFH
6000H-6FFFH
7000H-7FFFH
INT3#
R/W#
SP0256A-AL2
SBY
RST#
VDD
VD1
SE
TEST
Vss
OSC1
OSC2
ALD#
LRQ#
SER OUT
3.12MHz CRYSTAL
22pf
D0-D7
A0-A7
RESET#
3000H
200H
1000H
33KΩ
0.022μf
0.1μf
VOLUME
10KΩ
LM386
10μf
1μf
100μf
10Ω
8Ω 1W
"O.K."
GENERAL INSTRUMENT
MICROELECTRONICS GROUP


1 A0-A11
2 D0-D7
3 RESET#
4 INT3#
+5
1K
PARALLEL DATA-STROBE#
74LS14
74LS74
LATCH-BUSY#
5 200H
74LS374
LSB
MSB
PARALLEL DATA#
PARALLEL PORT (optional)
TMS 4016-25
8 R/W#
6 3000H
EXTERNAL RAM (optional)
Ex: 2K Byte
2732A-30
(X000H-XFFFH)
EXTERNAL EPROM (optional)
"EXCEPTION-WORD" AND/OR "USER"
Ex: 4K Byte
74LS373
7 1000H
REFER TABLE 5.
UART PARAMETER (optional)
GENERAL INSTRUMENT
MICROELECTRONICS GROUP

NOTES:

GENERAL INSTRUMENT MICROELECTRONICS AN0505D
Worldwide Sales Offices (c)1986

Sales Offices USA: CA, 818-789-0952 or 408-496-0844; FL, 813-894--3273; IL, 312-981-0040;
IN, 317-841-9328; MA, 617-038-5890/91; NC, 919-828-0317; NJ, 201-254-5024; TX, 214-934-1654;
EUROPE: London, Ruislip (08956)36141; Milano, (2)498 7362; Muenchen, (089)95997-35;
Paris, (1)43-74-9134; ASIA: Hong Kong, (3)722-6577; Osaka, (06)375-0606; Seoul, (2)739-8543;
Singapore, (65)344-4711; Taipei, (2)914-6234; Tokyo, (03)437-0281

"The information in this publication, including schematics, is suggestive only. General Instrument Corporation does not warrant, nor will be responsible or liable for, (a) the accuracy of such information, (b) its use or (c) any infringement of patents or other rights of third parties."

GENERAL
INSTRUMENT
MICROELECTRONICS
