ARFON MICROELECTRONICS SPEECH SYNTHESIS CARD

Functional and Software Documentation

Arfon Microelectronics Ltd.
Cibyn Industrial Estate,
Caenarfon,
Gwynedd, North Wales.

Telephone (0286) 5005/6/7

Telex: AMICRO 61126

Issue 3 For Part No AML SS1 07-07-81

Introduction

The design of the AML speech synthesis card is based on the National Semiconductors "DIGITALKER" system chip set. The card contains a speech processor chip (SPC) and two 64k speech ROMs, which contain a vocabulary of 143 words. Other speech ROMs will be made available in due course. The system produces high quality speech including emphasis of the original speech and natural inflection.

Also included on the card are a 700Hz and 200Hz filter, power amplifier, 2.5in loudspeaker and Nasbus/RS232c interface circuitry. Designed to Nasbus 3 specification, the standard unit can be used with a Nascom as supplied. Interface adaptors are available for PET, Apple and TRS 80 computer systems. An RS232c cased system is also available.

Installation

(1) Unpack the speech card and check that the card has not been damaged in transit. The bag in which the card is supplied has been anti-static treated. Keep the bag for card storage when not installed in a system.

(2) I/O EXT and N2 links. The I/O EXT link has been factory fitted for testing purposes only. If no other card in a Nascom system is providing an I/O EXT. signal then leave this link in place. Also change the appropriate link or switch on the Nascom 1 or 2 respectively (Nascom 1, LK1 from Int. to Ext and Nascom 2, LSW2. switch 8, move switch towards the edge of the card). The N2 link should remain if the system is running at 4Mhz (Nasbus line 5). The N2 link has ben provided so that an external 4Mhz TTL clock may be sent to the card. For most applications this link can be omitted, but if used remove XTL1,R1,R3 and C1. These links only apply to the Nascom range.

(3) Since the onboard loudspeaker has a limited frequency range it may be desirable to feed the audio output to a larger loudspeaker or hi-fi system. When a connection is made to the jack socket the onboard loudspeaker is disconnected. A 3.5mm mono jack socket would be used for this. If the maximum volume is insufficient then the gain of the power-amp can be increased (see circuit description on page 1-2).

(4) Set VR1 fully clockwise. Plug the card into the system. Power-up the system and if a multimeter or 'scope is available check the +12v and +5v lines. When power is applied to the card about 1 second of random word sounds will be heard. This is the SPC internal power-up circuit. Check the system for normal operation. The card has been thoroughly tested and burned in at the factory. If a Nascom is being used then the simple instruction below can be entered, if not then refer to the documentation supplied with the interface adaptor.

```
Newline
O F6 00
```

If all is correct you will hear the card say "This is Digitalker". Repeat the "out" instruction whilst adjusting VR1 to a comfortable listening level. Turn to page 1-7 for further programming information.

Circuit Description.

Before covering the Nasbus interface and analogue circuitry, a detailed examination of the National Semiconductors "DIGITALKER" system would be useful . The system comprises a speech processor chip and speech ROM(s). The system uses speech compression synthesis techniques. This technique dramatically reduces the amount of memory required to store speech when compared to other systems such as digitisation, PCM (Pulse Code Modulation) & ADPCM (Adaptive Code Pulse Code Modulation).

National Semiconductors use a computer program to analyse the tape recording and produce a ROM pattern that will synthesize the original recordings. During this process the speech waveform is sampled,digitised and compressesed by eliminating symmetrical redundancy and silence periods. During the compression algorithm, the voiced and unvoiced sounds are separated. The signal is adaptive delta modulated and the phase information is adjusted. By using this method speech elements can be synthesized as phonemes or even complete phrases, this data can then be stored on tape, disc or transferred to ROM or EPROM.

In the English language there are between 36 to 40 phonemes (comprising of 14 to 16 vowel sounds and 24 consonants), together with emphasis, inflection, and volume these produce the fundamental building blocks of speech. A phoneme is made up of either voiced (eye) or unvoiced sounds (sh). Unvoiced sounds are usually less frequent and less varied than voiced sounds. A speech synthesizer can exploit this difference. Including silence periods, speech rates are about 10 to 15 phonemes per second. The normal bit rate for phoneme speech is approximately 60 to 90 bits per second. The synthesis model has two driving functions, a grey noise source, which is a hissing sound for unvoiced sounds, and a tone source providing pitch for voiced sounds. The sounds created by the two sources are filtered by time-varying formant filters. We can easily relate the sythesizer to the human vocal tract. The lungs are the energy source, When air is passed through the vocal chords a pitch (voiced sound) is produced. Unvoiced sounds are produced as air is passed through the vocal chambers and not through the vocal chords. The formants are created by the throat, mouth and nasal cavities. By controlling these chambers, tongue and throat size, a phoneme can be generated.

Fricative sounds like "ch" or "sh" are created by pulses of noise normally around 2.5Khz to 8Khz. A typical English male voice would require three formant filters and the fricative formant. The "DIGITALKER" system can maintain the orginal attributes of the speaker ie, data can include inflection.

To obtain speech from the card, an eight bit word chosen from the master word list on page 1-5 is then written to Z80 port F6. The word start address is then loaded into the SPC (U1) word address register. When the WR line (U1 pin 4) goes high, the start address code is loaded into the control word register. The SPC then fetches the control word for the first block of speech data. The control word contains repeat and waveform information and the address of the speech data. This address is loaded into the phoneme register and is used to recreate the speech waveform. Further decoding takes place, voiced or unvoiced, half-period zeroed or not, male or female and silence. When the decoding is complete,synthesis then takes place. When using SSR1 (U2) and SSR2 (U3) ROMs, access of both chips during a single word can occur, therefor they should be used together. The computer can scan bit

0 (LSB) of port F6 to check for end of speech sequence. This bit goes high on completion of the speech sequence.

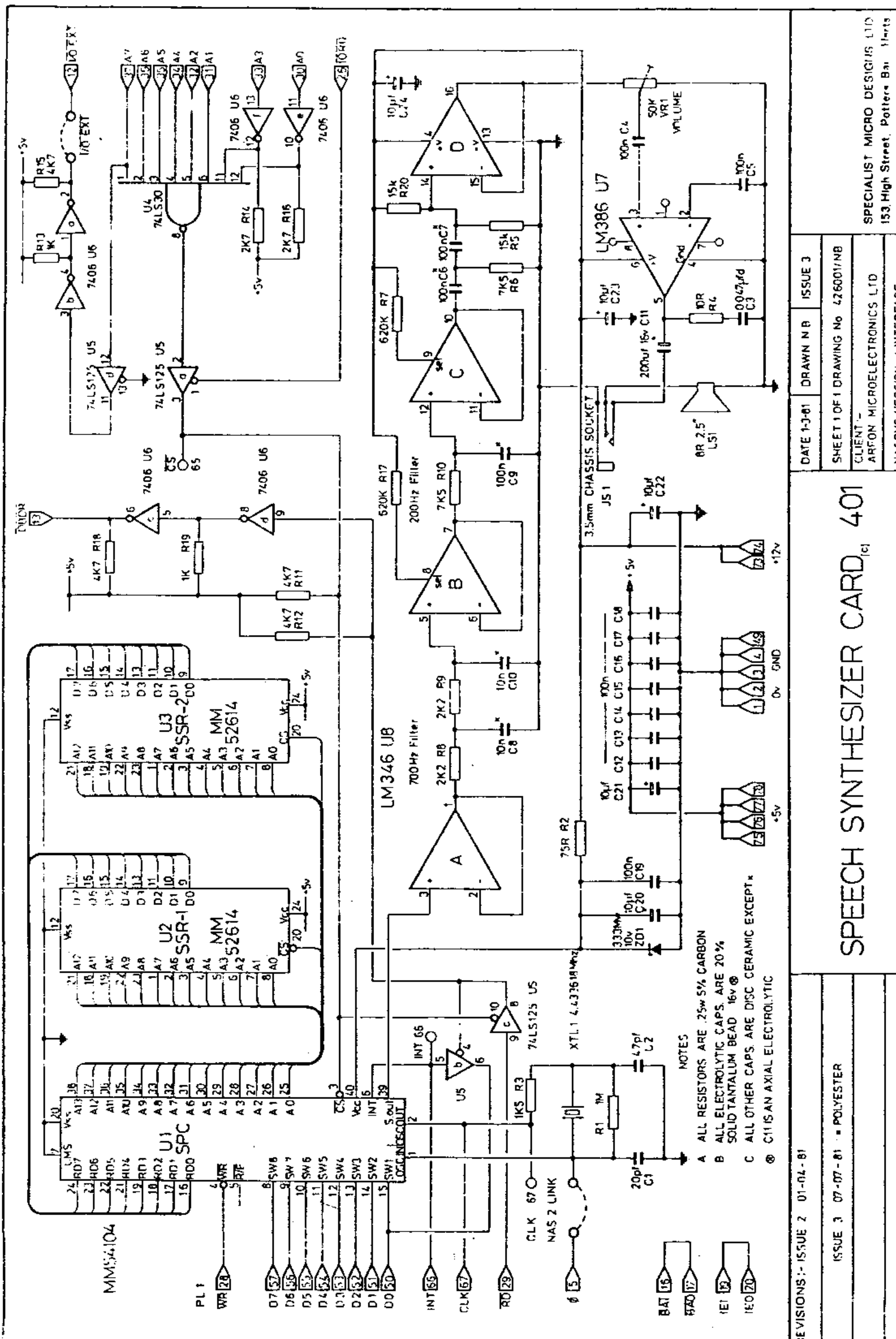
The sound output of the SPC (pin 39) is fed to 2 filters and a buffer amplifier. The first stage (U8-a) is a 7Khz filter. This filter is used to reduce the sampling noise. The second stage (U8-b&c) is a 200Hz low-pass filter with an attenuation characteristic of 20db per decade. This filter is used to compensate for the high frequency pre-emphasis used in this technique. The conditioned signal is then fed to a buffer amplifier, whose output is fed to VR1. VR1 controls the input gain to the power amplifier LM386N (U7). The amplifier has an internal 1k35 resistor between pins 1 and 8. With no connection to these pins the amplifier will have a gain of 20 (26 dB). If a 10Mfd capacitor is wired to pins 1 and 8 to by-pass the resistor, the gain will be increased to 200 (46 dB). If a resistor is wired in series with the capacitor the gain can be set from 20 to 200. Three pads on the PCB are connected to pins 1,7 and 8 to facilitate a change of gain. A 3.5mm jack plug can be connected to JS1. This will turn off the on-board 2.5inch loud-speaker. Better sound quality can be accomplished if the synthesiser card is connected to a Hi-Fi system or larger loudspeaker.

There are 2 links on the AML SS1 card. The link marked "N2" should only be used if the system bus is carrying a 4Mhz clock signal as per Nascom 2. The other link marked "I/O EXT" should only be connected if no other card in the system is providing this signal. When an "I/O EXT" link is used on any card the appropriate link or switch on the Nascom 1 and 2 cards respectively, should be changed. If the system is not Nascom then refer to the interface documentation provided.

U4 74LS30 decodes the port F6. Its output is taken to a tri-state (U5,a) buffer, which is enabled when IORQ becomes active. When port F6 is called the output of this buffer is forced low (in tri-state mode pulled high by R11) and enables the SPC chip select pin 3.

Buffer U5c is also enabled. Its output goes to logic low if the Z80 RD signal is active. The output of U5c is also fed to two open-collector inverters (U6c and U6d), which maintains the original logic level. The output of U6c provides the Nasbus signal DBDR (PL1 pin 13), which is used to determine the direction of bi-directional data bus buffers on the Nascom buffer card. This signal is not used by Nascom 2.

Fig. 2 Circuit diagram



National Semiconductor DT1050 master word list (SSR1 & 2)

Word	Hex	Dec	Word	Hex	Dec
This is Digitalker	00	0	Q	30	48
One	01	1	R	31	49
Two	02	2	S	32	50
Three	03	3	T	33	51
Four	04	4	U	34	52
Five	05	5	V	35	53
Six	06	6	W	36	54
Seven	07	7	X	37	55
Eight	08	8	Y	38	56
Nine	09	9	Z	39	57
Ten	0A	10	Again	3A	58
Eleven	0B	11	Ampere	3B	59
Twelve	0C	12	And	3C	60
Thirteen	0D	13	At	3D	61
Fourteen	0E	14	Cancel	3E	62
Fifteen	0F	15	Case	3F	63
Sixteen	10	16	~~Can~~ CENT	40	64
Seventen	11	17	400Hz tone	41	65
Eighteen	12	18	80 Hz tone	42	66
Nineteen	13	19	20 mS. silence	43	67
Twenty	14	20	40 mS. silence	44	68
Thirty	15	21	80 mS. silence	45	69
Forty	16	22	160mS. silence	46	70
Fifty	13	23	320mS. silence	47	71
Sixty	18	24	Centi	48	72
Seventy	19	25	Check	49	73
Eighty	1A	26	Comma	4A	74
Ninety	1B	27	Control	4B	75
Hundred	1C	28	Danger	6C	76
Thousand	1D	29	Degree	4D	77
Million	1E	30	Dollar	4E	78
Zero	1F	31	Down	4F	79
A	20	32	Equal	50	80
B	21	33	Error	51	81
C	22	34	Feet	52	82
D	23	35	Flow	53	83
E	24	36	Fuel	54	84
F	25	37	Gallon	55	85
G	26	38	Go	56	86
H	27	39	Gram	57	87
I	28	40	Great	58	88
J	29	41	Greater	59	89
K	2A	42	Have	5A	90
L	2B	43	High	5B	91
M	2C	44	Higher	7C	92
N	2D	45	Hour	5D	93
O	2E	46	In	5E	94
P	2F	47	Inches	5F	95

Word list continued,

Word	Hex	Dec	Word	Hex	Dec
Is	60	96	Please	78	120
It	61	97	Plus	79	121
Kilo	62	98	Point	7A	122
Left	63	99	Pound	7B	123
Less	64	100	Pulses	7C	124
Lesser	65	101	Rate	7D	125
Limit	66	102	Re	7E	126
Low	67	103	Ready	7F	127
Lower	68	104	Right	80	128
Mark	69	105	Ss (See below)	81	129
Meter	6A	106	Second	82	130
Mile	6B	107	Set	83	131
Milli	6C	108	Space	84	132
Minus	6D	109	Speed	85	133
Minute	6E	110	Star	86	134
Near	6F	111	Start	87	135
Number	70	112	Stop	88	136
Of	71	113	Than	89	137
Off	72	114	The	8A	138
On	73	115	Time	8B	139
Out	74	116	Try	8C	140
Over	75	117	Up	8D	141
Parenthesis	76	118	Volt	8E	142
Percent	77	119	Weight	8F	143

Notes

(1) "Ss" (Hex 81, Dec 129) makes any singular word plural.

(2) Silence periods (Hex 43 to 47), have been included to improve the quality of speech phrasing. As a rough guide to their use, words beginning with the letters B,D,G,K,P and T insert 80mS silence prior to the word. For words ending in these letters insert 40mS.

(3) If a call is made to the speech card higher than decimal 143, unintelligible invalid speech will be output. Other speech ROMs may allow calls higher than 143.

Programming the AML SS1 speech synthesizer

This Basic program causes the synthesizer to recite its entire vocabulary. If other speech ROM sets are used instead of the DT1050 set, then change the highest number in line 100, eg the DT1057 set would require 256 to be entered.

```
100 FOR A=0 TO 143            : 0 is the first word & 143 is the last word
110 GOSUB 200                 : Call silence routine
120 GOSUB 300                 : Call word output routine
130 NEXT                      : Loops to line 100
140 END                       :

200 B=A                       : Save the contents of A
210 A=71                      : 71 is the code for 320mS silence period
220 GOSUB 300                 :
230 A=B                       : Get A back
240 RETURN                    : Return to program

300 WAIT 246,1                : Wait till not busy
310 OUT 246,A                 : Output word to synthesizer
320 RETURN                    : Return to program
```

This second Basic program allows words not contained in the speech ROMs to be generated. The parameters provided by the data statements determine the particular word spoken and the length of time that word is used. The message contained in the statements is "The AML speech unit".

```
100 READ A                    : Get word number
110 IF A=0 THEN 500           : Finished?
120 READ B                    : Get delay
130 OUT 246,A                 : Output word to synthesizer
140 IF B<>0 GOTO 170          : Early interrupt?
150 WAIT 246,1                : No - wait till done
160 GOTO 100                  : Get next
170 FOR C=0 TO B              : Yes do delay
180 NEXT C                    :
190 GOTO 100                  : Get next

400 DATA 138,0,70,0,32,0,69,0,44,0,69,0
410 DATA 43,0,70,0,133,200,67,0,73,36,70,0
420 DATA 52,100,111,100,69,0,51,10,67,0
430 DATA 0

500 END
```

The following BASIC program is a small editor, which allows construction of a series of codes for presentation to the speech card, using the same data format as the previous BASIC routines.

ie. Digitalker code, delay

The phrases and sentences constructed, may be saved on tape and read back either into the editor again, or into the program which is actually going to use the sounds.

In response to the prompt "Command ? ", type one of the editor commands detailed below.

"A" Add/Alter

Allows alteration of existing data, or addition of new data to the end of the file.

If there is no data in the file the command must be 'Add' so the program drops through to the "Word number ?" prompt described below.

If data exists in the file, the line "Highest entry number is x" will be displayed, followed by a prompt for an entry number. A number in the range 0 to x+1 should now be entered. Typing 'enter' only, or the number x+1, will add an entry to the end of the file, but a number in the range 0 to x will 'Alter' an existing entry.

Requesting an 'Alter' will give a display of the current entry at that point and the prompt "Change delay only ?" will appear. Respond with 'Y' or 'N'. If 'N' the program will continue as for 'Add' described below. 'Y' will permit alteration to the delay parameter only.

'Add'ing an entry consists of answering three prompts. 1) "Word number ?" - enter a Digitalker word code number in decimal range 1 to 255. 2) "Delay ?" - enter positive decimal number. Default ('enter' key with no number entry) is 0, directing the output routine to wait until the complete sound is output, before starting the next sound. Any other number will cause a delay, proportional to the number, before starting the next sound, regardless of the state of the 'busy' line. 3) "Sound string ?" - enter a character string representative of the sound. This is to give a guide to assist in editing changes only, and has absolutely no effect on sound output. Default is a null string. This string is not saved on tape.

After this the entry and those surrounding it (if they exist) will be displayed before requesting the next command.

"O" Open

Opens a gap in the file to insert an additional entry.

Responds with "Highest entry is x", then the prompt "Open from where ?". Enter a number between 0 and x, and the entry number specified plus all those above it, will be moved up the buffer 1 place, and the 320ms silence code is inserted at x. Default is '0'.

The 'end play' position (see below) is modified in the expected fashion. After this the entry and those surrounding it (if they exist) will be displayed before requesting the next command.

"C" Close

Closes up the file to remove an unwanted entry.

Responds with "Highest entry is x", then the prompt "Close up where ?". Enter a number between 0 and x, and the entry number specified will be deleted and all those above it will be moved down the buffer 1 place. If either a number greater than x, or nothing (default) is entered, the command is aborted.

The 'end play' position (see below) is modified in the expected fashion. After this the entry and those surrounding it (if they exist) will be displayed before requesting the next command.

"D" Display

Display portions of the entry buffer.

Again the highest entry information is given followed by "Display from ?" prompt. A suitable response will display a screen-full of the buffer.

"E" End play

Sets the end of play pointer.

Sometimes it is required that only a small portion of the buffer be heard on 'Play' (see below). This command sets a pointer such that play will terminate at a specified entry. Default entry is the current end of buffer. The 'end play' pointer is moved by "O", "C" and "A" commands as would be expected. In the latter case the pointer is only modified if an 'Add'ition is made to the buffer, and the 'end play' pointer is there as well.

"P" Play

Outputs a section of the buffer to the Speech Card.

An indication of the current 'end play' pointer is given, followed by "Play from ?". Answer this prompt in the obvious manner, default entry is 0.

As the sounds are produced from the Speech card, the display shows the entry number of the current sound, and the 'sound string' if entered.

"W" Write to tape

Writes edit buffer to tape.

This command saves the numeric data in the edit buffer to a cassette recorder, using the CSAVE* command in BASIC, for later re'entry into the editor or target program. The string data is not saved on tape.

The edit buffer is not altered by this command.

"R" Read from tape

Reads an edit buffer from tape using CLOAD*.

The tape need not have been written by the edit program, as long as it was written by CSAVE* and is the correct length (see Program notes below).

The current edit buffer is overwritten by this command.

"H" Help

Gives a summary of the possible edit commands.

PROGRAM NOTES

Main variables used.

A(99,1) Holds all numeric data.
D$(99) Holds all string data.
NU is a count of the number of entries in the buffer.
EP is the current 'end play' pointer.

The size of the data arrays may, of course, be altered in line 110. If this is done the value 99 on the following lines must also be changed: 370, 1340, 1360.

NU is the entry count, but since the data array subscripts start from 0, it also serves as a pointer to the first available space in the buffer.

Output to the card is done in lines 990 to 1040 and could serve as a model for other programs.

Display of data during output to the speech card causes some extra delay, so for sounds or phrases which are very time dependent, it would be best to 'REM' out line 1000 if the data is to be used in another program with a tighter output loop. It would not be difficult to add a software switch to this program, to skip this display when required.

This program is written for a NASCOM with 8k BASIC, but should be easily alterable for other machines. Watch out for the default entries, different BASICs deal with these in different ways.

```
10 REM      Basic sound editor for the
20 REM                A.M.L.
30 REM              Speech Card
40 REM
50 REM   for NASCOM with 8K Microsoft Basic
60 REM
70 CLS
80 PRINT "Editor for AML speech card":PRINT
90 PRINT "Type 'H' for help when prompted."
100 PRINT:PRINT
```

```
110 DIM D(99,1),D$(99)
120 DT=246:           REM Port number
130 FOR I=0 TO 99:    REM clear buffer
140 D$(I)=""
150 FOR J=0 TO 1
160 D(I,J)=0
170 NEXT J,I
180 NU=0:             REM number of data entered
190 EP=0:             REM 'end play' pointer
200 REM ********    MAIN LOOP    ***********
210 INPUT "Command ";C$
220 IF C$="O" GOTO 330
230 IF C$="C" GOTO 490
240 IF C$="A" GOTO 610
250 IF C$="D" GOTO 850
260 IF C$="P" GOTO 950
270 IF C$="E" GOTO 1070
280 IF C$="H" GOTO 1160
290 IF C$="W" GOTO 1270
300 IF C$="R" GOTO 1320
310 GOTO 210
320 REM ********** open
330 IF NU=0 GOTO 1090
340 GOSUB 1400
350 A=0:INPUT "Open from where ";A:A=ABS(A)
360 IF A>=NU GOTO 1140
370 IF NU<>99 GOTO 390
380 PRINT "Buffer full !!"":goto 250
390 FOR I=NU TO A+1 STEP -1
400     D(I,0)=D(I-1,0)
410     D(I,1)=D(I-1,1)
420     D$(I)=D$(I-1)
430 NEXT I
440 NU=NU+1
450 D(A,0)=71:D(A,1)=0:D$(A)="320ms gap"
460 IF EP>=A THEN EP=EP+1
470 GOTO 820:          REM display changes
480 REM ********** close
490 IF NU=0 GOTO 1090
500 GOSUB 1400
510 A=100:INPUT "Close up where ";A:A=INT(A)
520 IF A>=NU GOTO 1140
530 FOR I=A TO NU-1
540     D(I,0)=D(I+1,0)
550     D(I,1)=D(I+1,1)
560     D$(I)=D$(I+1)
570 NEXT I
580 NU=NU-1:IF EP>=A THEN EP=EP-1
590 GOTO 820
600 REM ********** alter
610 GOSUB 1400:          REM display changes
620 IF NU=0 THEN NU=1:GOTO 740
630 A=NU:INPUT "Data number to Alter ";A
640 A=ABS(A):IF A>NU GOTO 1140
650 IF A<>NU GOTO 680
660 NU=NU+1:IF EP=0 OR EP=NU-2 THEN EP=NU-1
670 GOTO 740
680 PRINT A,D(A,0);",";D(A,1),D$(A)
690 INPUT "Change delay only ";A$
700 IF LEFT$(A$,1)<>"Y" GOTO 740
710 INPUT "New delay ";D(A,1)
```

```
720 D(A,1)=INT(ABS(D(A,1)))
730 GOTO 210
740 INPUT "Word number ";I:I=INT(ABS(I))
750 IF I=0 OR I>255 GOTO 740
760 D(A,0)=I:I=0
770 INPUT"Delay          ";I:D(A,1)=INT(ABS(I))
780 A$="":INPUT "Sound string";A$
790 IF LEN(A$)>12 THEN A$=LEFT$(A$,12)
800 D$(A)=A$
810 REM ** display changes
820 D=A-6:IF D<0 THEN D=0
830 GOTO 890
840 REM ********** display
850 IF NU=0 GOTO 1090
860 GOSUB 1400
870 D=0:INPUT"Display from ";D:D=ABS(D)
880 IF D>=NU GOTO 1140
890 FOR I=D TO D+13
900      IF I=NU THEN I=D+13:GOTO 920
910      PRINT I,D(I,0);",";D(I,1),D$(I)
920 NEXT I
930 GOTO 210
940 REM ********** play
950 IF NU=0 GOTO 1090
960 PRINT "Current play end is";EP;".  ";
970 P=0:INPUT "Play from ";P:P=ABS(P)
980 IF P>=NU GOTO 1140
990 FOR I=P TO EP
1000      PRINT I,D$(I)
1010      OUT DT,D(I,0)
1020      IF D(I,1)=0 THEN WAIT DT,1:GOTO 1040
1030      FOR J=1 TO D(I,1):NEXT J
1040 NEXT I
1050 GOTO 210
1060 REM ********** end play
1070 IF NU<>0 GOTO 1100
1080 GOSUB 1400
1090 PRINT "No data entered yet !!":GOTO 210
1100 PRINT "Current end play is";EP
1110 A=0:INPUT "End play where ";A:A=ABS(A)
1120 IF A=0 THEN EP=NU-1:GOTO 210
1130 IF A<NU THEN EP=A:GOTO 210
1140 PRINT "No data that far yet !!":GOTO 210
1150 REM ********** help
1160 PRINT:PRINT "Command list :"
1170 PRINT "'O' pen"
1180 PRINT "'C' lose"
1190 PRINT "'A' lter/Add"
1200 PRINT "'D' isplay"
1210 PRINT "'P' lay"
1220 PRINT "'E' nd play"
1230 PRINT "'W' rite to tape"
1240 PRINT "'R' ead from tape"
1250 PRINT:GOTO 210
1260 REM ********** write
1270 IF NU=0 GOTO 1090
1280 PRINT"Start tape, press 'ENTER' to Write";
1290 INPUT A$:PRINT "Writing .... ";:CSAVE*D
1300 PRINT "Complete. Stop tape.":GOTO 210
1310 REM ********** read
1320 PRINT"Start tape,press 'ENTER' to Read";
```

```
1330 INPUT A$:PRINT "Reading .... ";:CLOAD*D
1340 FOR I= 0 TO 99
1350    IF D(I,0)<>0 GOTO 1370
1360    NU=I:EP=I-1:I=99
1370 NEXT I
1380 PRINT "Complete. Stop tape .":GOTO 210
1390 REM ********** print highest No. entered
1400 IF NU=0 THEN PRINT "No data yet":RETURN
1410 PRINT"Highest entry number is";NU-1:RETURN
```

The following section presents a Z80 machine code routine to perform data output to the Speech Card. It uses a different data format from the BASIC programs, in the way that it deals with sound cut-off. It is more efficient in data storage if most of the sounds to be output are complete Digitalker words. The approach of the BASIC programs could, of course, be implemented in machine code if required.

```
********************************
* AML Speech Card output routine *
********************************
```

This is a fully relocatable routine to output a sequence of words/sounds from the AML Speech Card.

The routine is called at label "dtout" with HL pointing to a block of data which contains the series of desired sound codes.

Each sound code may take one of two possible forms:

```
        1) ss         [1 byte]
        2) ss 00 dd   [3 bytes]
```

Where ss is the byte code of one of the Digitalker words, 00 is a null byte and dd is a delay factor. The data block is terminated by '00 00'.

In case 1) the code ss is sent to the Speech card, and the next code is not sent until the card indicates 'not busy'.

In case 2) the code is sent to the card, the routine then delays for a time proportional to dd, then sends the next code to the card, regardless of the busy status - thus possibly cutting off the sound currently being output, and replacing it with a new sound.

An example of a legal data block:
ss ss ss 00 dd ss ss 00 dd ss 00 dd ss 00 00
Where ss and dd are in the range 01 - FF hex.

NOTE: If the data finishes 'ss 00 dd 00 00' then the final sound will not be cut off, since there is no further data to be sent. Sending one of the 'silence' codes as the final sound, would have the desired effect.

In short:
Call 'dtout' with data at (HL) terminated by '00 00'. Sequence '00 xx' is taken as delay count 'xx' to cut off previous sound.

The only register to be modified by this routine is HL, which is left pointing at the byte following the terminating '00 00'. This could be the start of the next sound sequence.

```
                                .z80

00F6                    dt      equ     0f6h    ; Speech Card port
0200                    dcon    equ     0200h   ; delay constant


                        ; ENTRY

0000'   F5              dtout:  push    af      ; save status
0001'   C5                      push    bc
0002'   0E 00           dtout1: ld      c,0     ; initialise delay flag
0004'   7E              dt1:    ld      a,(hl)  ; get data
0005'   23                      inc     hl
0006'   B7                      or      a       ; null ?
0007'   20 1D                   jr      nz,dt3  ; no - it is sound code
0009'   7E                      ld      a,(hl)  ; yes - get next byte
000A'   23                      inc     hl
000B'   B7                      or      a       ; delay or end ?
000C'   20 03                   jr      nz,dt2  ; delay - go to it
000E'   C1              dtx:    pop     bc      ; end - restore status
000F'   F1                      pop     af
0010'   C9                      ret

                        ; do delay. Count is in 'A'
0011'   0E FF           dt2:    ld      c,-1    ; set delay flag
0013'   C5                      push    bc      ; save it
0014'   01 0200         delay:  ld      bc,dcon ; preload constant
0017'   F5                      push    af
0018'   0B              del1:   dec     bc      ; delay loop
0019'   F1                      pop     af
001A'   F5                      push    af
001B'   78                      ld      a,b
001C'   B1                      or      c
001D'   20 F9                   jr      nz,del1
001F'   F1                      pop     af
0020'   3D                      dec     a       ; finished ?
0021'   20 F1                   jr      nz,delay
0023'   C1                      pop     bc
0024'   18 DE                   jr      dt1     ; get next data

                        ; output to Speech Card
0026'   0C              dt3:    inc     c       ; have we just finished
                                                ;   a delay ?
0027'   28 07                   jr      z,dt4   ; yes - do not wait
0029'   47                      ld      b,a     ; save code
002A'   DB F6           dt3a:   in      a,(dt)  ; wait for 'not busy'
002C'   1F                      rra
002D'   30 FB                   jr      nc,dt3a
002F'   78                      ld      a,b     ; get code back
0030'   D3 F6           dt4:    out     (dt),a  ; send to card
0032'   18 CE                   jr      dtout1  ; get next data

                                end
```

```
Macros:

Symbols:
DCON      0200     DEL1      0018'    DELAY     0014'    DT        00F6
DT1       0004'    DT2       0011'    DT3       0026'    DT3A      002A'
DT4       0030'    DTOUT     0000'    DTOUT1    0002'    DTX       000E'


No  Fatal error(s)
```

Component list

AML SS1

Issue 3

No.	Qty	Description	Circuit Ref.
INTEGRATED CIRCUITS			
01	1	MM54104 MOS Speech processor	U1
02	1	MM52614 DT1050 SSR-1 64K ROM	U2
03	1	MM52614 DT1050 SSR-2 64K ROM	U3
04	1	74LS30 8 input NAND gate	U4
05	1	74LS125 Quad Tri-state buffer	U5
06	1	7406 Hex Open-collector inverter	U6
07	1	LM386N Low voltage audio power Amp.	U7
08	1	LM346N Programmable Quad Op-amp	U8
RESISTORS			
09	1	1M .25w 10% Hystab Brn/Blk/Grn	R1
10	1	75R .25w 10% Hystab Voi/Grn/Blk	R2
11	1	1k5 .25w 10% Hystab Brn/Grn/Red	R3
12	1	10R .25w 10% Hystab Brn/Blk/Blk	R4
13	2	7k5 .25w 10% Hystab Mau/Grn/Red	R6 & R10
14	2	620k .25w 10% Hystab Blu/Red/Yel	R7 & R17
15	2	2k2 .25w 10% Hystab Red/Red/Red	R8 & R9
16	4	4k7 .25w 10% Hystab Yel/Mau/Red	R11,R12,R15 & R18
17	2	1k .25w 10% Hystab Brn/Blk/Red	R13 & R19
18	2	2k7 .25w 10% Hystab Red/Mau/Red	R14 & R16
19	2	15k .25w 10% Hystab Brn/Gry/Ora	R5 & R20
CAPACITORS			
20	1	20pf 30v Sub. Min. plate ceramic 10%	C1
21	1	47pf 30v Sub. Min. plate ceramic 10%	C2
22	10	100n 36v Monolithic ceramic 20%	C4,C5 & C12-C19
23	3	100n 25v Min. Polyester layer 10%	C6,C7 & C9
24	2	10n 25v Min. Polyester layer 10%	C8 & C10
25	1	47n 30v Disc ceramic 20%	C3
26	5	10Mfd 16v Min. Solid Tant. Bead 20%	C20 - C24
27	1	200Mfd 16v Axial Electrolytic 25%	C11
		(minimum operating voltages DC)	
IC SOCKETS			
28	1	40pin .6in Dual in-line	U1
29	2	24pin .6in Dual in-line	U2 & U3
MISCELLANEOUS			
30	1	10v 330Mw Zener diode	ZD1
31	1	2.5in 8 ohm Loudspeaker	LS1
32	1	3.5 mm insulated chasis socket	JS1
33	1	50k Preset resistor (horiz or vert.)	VR1
34	1	4.433618Mhz HC-18/U Holder crystal	XTL1
35	1	AML SS1 16 Swg Fibreglass, 5"x8", D/S,	PTH PCB
36	1	Set issue 3 documentation	

Alternate Speech ROM Set DT 1057. Available August 1981.

Fig. 1 Component Placement Plan

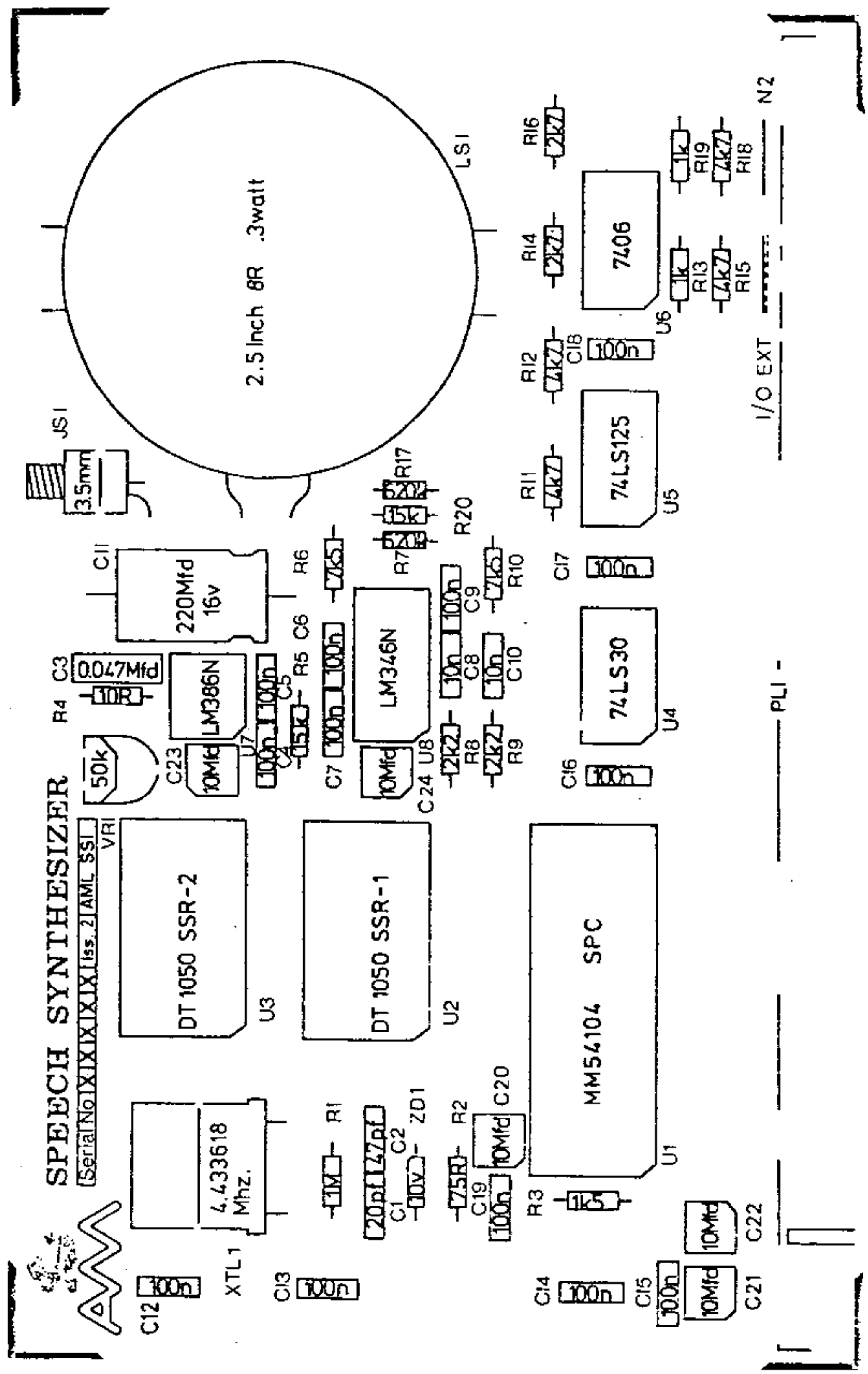
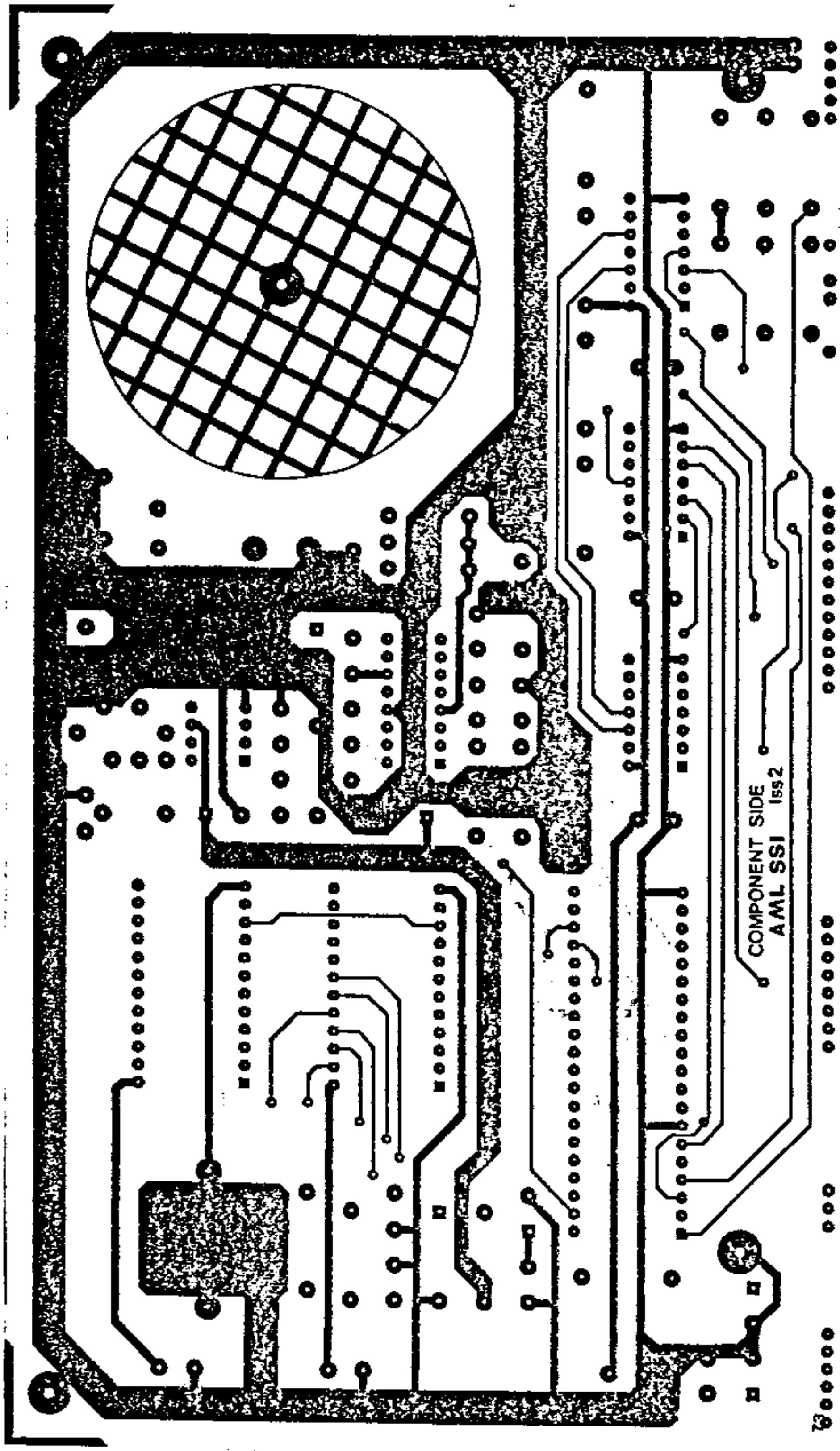


Fig. 3 PCB track plan (component side)



References

1. Smith, Jim., Speech Synthesis application note No. 252. National Semicondutors July 1980.

2. Weinrich, David W., A Speech Synthesis Chip using compression techniques, Electronics, April 1980.

3. Smith and Weinrich., Designers' guide to Speech Synthesis, Digital Design, February 1981.

Guarantee and service facilities

The AMLSS1 speech synthesis unit is guaranteed for a period of one year from the date of purchase.

If the card should require service, send the card to:-

Service Department.
Arfon Microelectronics Ltd.
Cibyn Industrial Estate,
Caernarfon,
Gwynedd, North Wales.

The card returned for service must be adequately packed, preferably in the carton in which it was supplied. Postal charges must be prepaid. We cannot accept responsibility for items arriving damaged. If the cause of failure is due to abuse or misuse during the guarantee period or if the guarantee has expired, the repair will be carried out immediately and charged for. If the repair is covered by the guarantee please supply proof of purchase date.

Nick Broome (Hardware), Dave Lewis and Howard Birket (Software) 3-3-81

Design and consultancy provided by Specialist Micro Designs Ltd.

DIGITALKER™ Speech Synthesis System

General Description

The DIGITALKER is a speech synthesis system consisting of multiple N-channel MOS integrated circuits. It contains a speech processor chip (SPC) and speech ROM and when used with external filter, amplifier, and speaker, produces a system which generates high quality speech including the natural inflection and emphasis of the original speech. Male, female, and children's voices can be synthesized.

The SPC communicates with the speech ROM, which contains the compressed speech data as well as the frequency and amplitude data required for speech output. Up to 128k bits of speech data can be directly accessed. This can be expanded with minimal external logic.

With the addition of an external resistor, on-chip debounce is provided for use with a switch interface.

An interrupt is generated at the end of each speech sequence so that several sequences or words can be cascaded to form different speech expressions.

Encoding (digitizing) of custom word or phrase lists must be done by National Semiconductor. Customers submit to the factory high quality recorded magnetic reel to reel tapes containing the words or phrases to be encoded. National Semiconductor will sell kits consisting of the SPC and ROM(s) containing the digitized word or phrases.

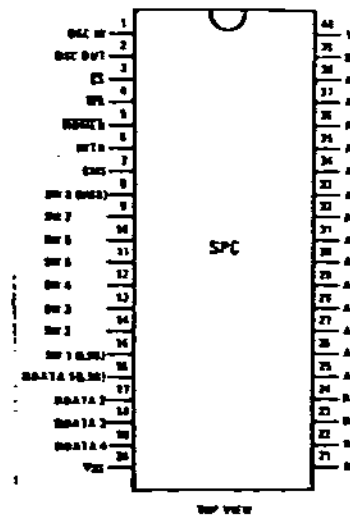
Features

- Designed to be easily interfaced to most popular microprocessors
- 256 possible addressable expressions
- Male, female, and children's voices
- Natural inflection and emphasis of original speech
- Addresses 128k of ROM directly
- Communicates with static or clocked dynamic ROMs
- TTL compatible
- MICROBUS™ compatible
- On-chip switch debounce for interfacing to manual switches independent of a microprocessor
- Easily expandable to greater than 128k ROM
- Interrupt capability for cascading words or phrases
- Crystal controlled or externally driven oscillator
- Ability to store silence durations for timing sequences

Applications

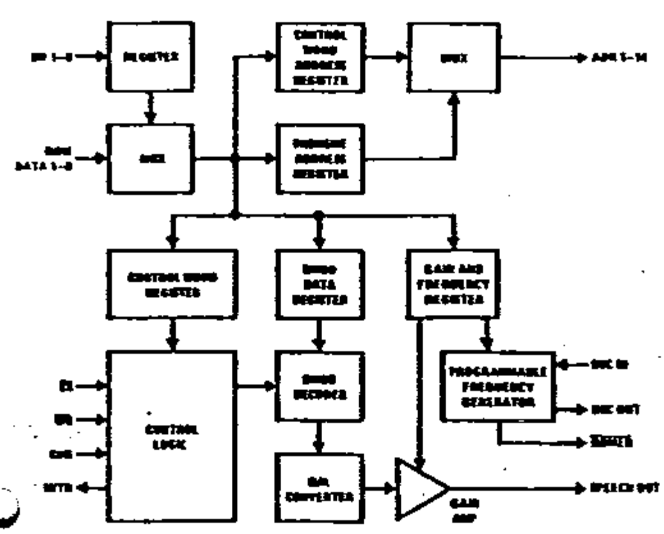
- Telecommunications
- Appliance
- Automotive
- Teaching aids
- Consumer products
- Clocks
- Language translation
- Annunciators

Dual-In-Line Package



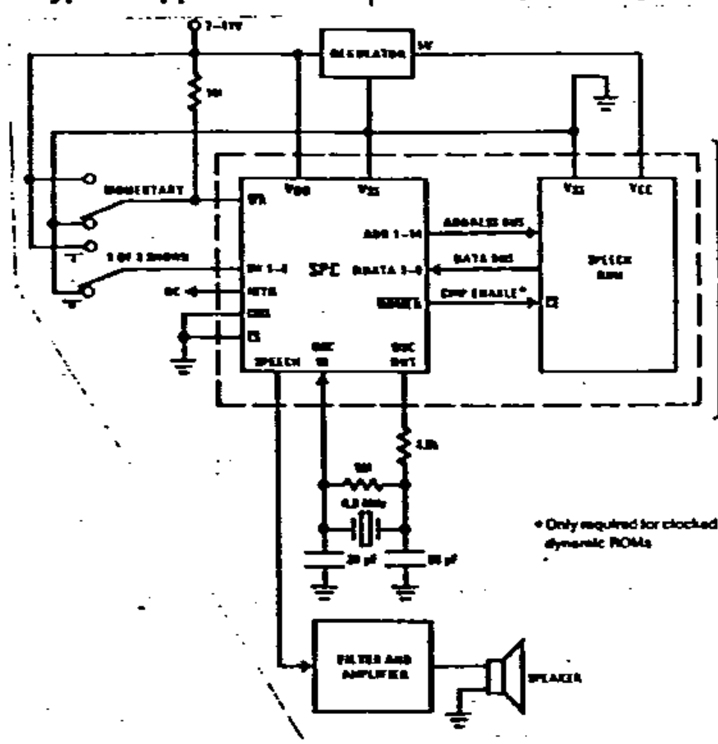
Top View

Block and Connection Diagrams



Typical Application

Minimum Configuration Using Switch



* Only required for clocked dynamic ROMs

Absolute Maximum Ratings

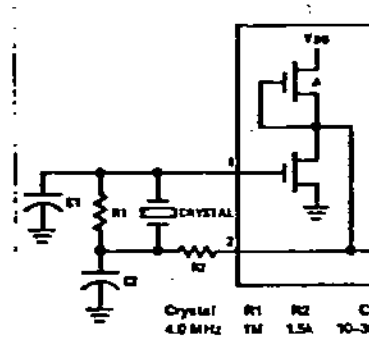
Storage Temperature Range	−65°C to +150°C
Operating Temperature Range	0°C to 70°C
V_{DD}-V_{SS}	12V
Voltage at Any Pin	12V
Operating Voltage Range, V_{DD}-V_{SS}	7V to 11V
Lead Temperature (Soldering, 10 seconds)	300°C

DC Electrical Characteristics $T_A = 0°C$ to $70°C$, $V_{DD} = 7V$–$11V$, $V_{SS} = 0V$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		−0.3		0.8	V
V_{IH}	Input High Voltage		2.0		V_{DD}	V
V_{OL}	Output Low Voltage	$I_{OL} = 1.6$ mA			0.4	V
V_{OH}	Output High Voltage	$I_{OH} = -100$ μA	2.4		5.0	V
V_{ILX}	Clock Input Low Voltage		−0.3		0.6	V
V_{IHX}	Clock Input High Voltage		4.0		V_{DD}	V
I_{DD}	Power Supply Current				50	mA
I_{IL}	Input Leakage				±10	μA
I_{ILX}	Clock Input Leakage				±10	μA
V_S	Silence Voltage			$0.45 V_{DD}$		V
V_{OUT}	Peak to Peak Speech Output	$V_{DD} = 11V$		2.0		V

Crystal Circuit Information

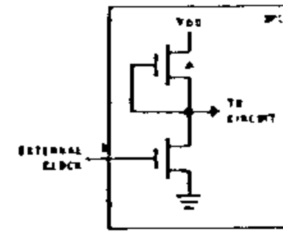
Typical Crystal Oscillator Netw

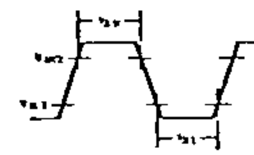


AC Electrical Characteristics $T_A = 0°C$ to $70°C$, $V_{DD} = 7V$–$11V$, $V_{SS} = 0V$, unless otherwise specified

Symbol	Parameter	Min	Max	Units
t_{sw}	CMS Valid to Write Strobe	350		ns
t_{csw}	Chip Select ON to Write Strobe	310		ns
t_{dw}	Data Bus Valid to Write Strobe	50		ns
t_{ws}	CMS Hold Time after Write Strobe	50		ns
t_{wd}	Data Bus Hold Time after Write Strobe	100		ns
t_{ww}	Write Strobe Width (50% Point)	430		ns
t_{rod}	$\overline{ROMEN}$ ON to Valid ROM Data		2	µs
t_{wsa}	Write Strobe to Speech Output Delay		410	µs
t_f	External Clock Frequency Tolerance		±2	%

External Clock Input (4.0

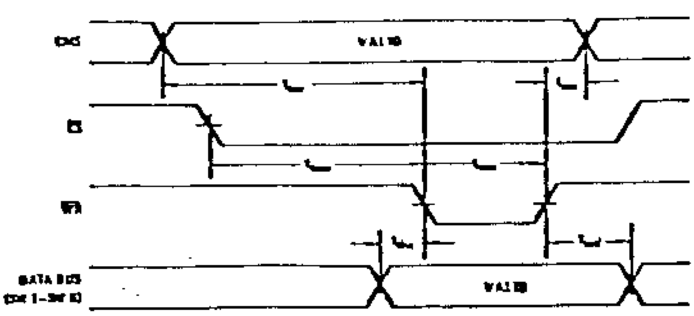




Timing	Min	Units
t_{XH}	300	ns
t_{XL}	300	ns

Timing Waveforms

Command Sequence



Functional Description

The following describes the function of all SPC input and output pins.

Note: In the following descriptions, a low represents a logic 0 (0.4V nominal), and a high represents a logic 1 (2.4V nominal).

INPUT SIGNALS

Chip Select ($\overline{CS}$): The SPC is selected when $\overline{CS}$ is low. It is only necessary to have $\overline{CS}$ low during a command to the SPC. It is not necessary to hold $\overline{CS}$ low for the duration of the speech data.

Data Bus (SW 1–8): This is an 8-bit parallel data bus which contains the starting address of the speech data.

Command Select (CMS): This line is used to define the t[illegible]ommands to the SPC.

CMS	Function
0	Reset interrupt and start speech sequence
1	Reset interrupt only

Write Strobe ($\overline{WR}$): This line latches the starting address (SW1–SW8) into a register. On the rising edge of the $\overline{WR}$, the SPC starts execution of the command specified by CMS. The command sequence is shown in the timing waveform section. If a command to start a new speech sequence is issued during a speech sequence, the new speech sequence will be started immediately.

ROM Data (RDATA 1–8): This is an 8-bit parallel data bus which contains the speech data from the speech ROM.

OUTPUT SIGNALS

Interrupt (INTR): This signal goes high at the completion of any speech sequence. It is reset by the next valid command. It is also reset at power up.

ROM Address (ADR1–ADR14): This is a 14-bit parallel bus that supplies the address of the speech data to the speech ROM.

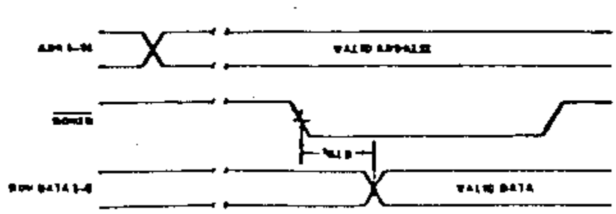
ROM Enable ($\overline{ROMEN}$): This line is for use with clocked dynamic ROMs. When used, the high to low transition must cause the speech ROM to generate a cycle and place the speech data on the RDATA lines. Data must remain on the RDATA lines while $\overline{ROMEN}$ is low. For low power applications, this line can be used to drive a transistor that switches the supply for static speech ROMs. See ROM data timing.

Speech Output (Speech Out): This is the analog output that represents the speech data. See frequency response section.

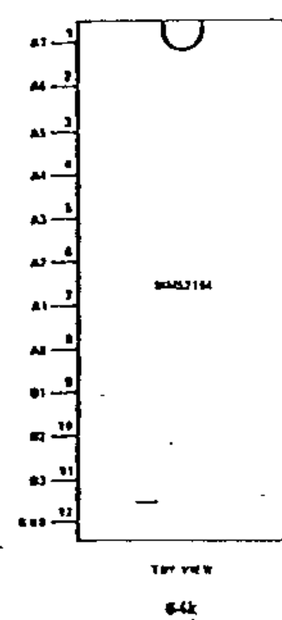
INPUT/OUTPUT SIGNALS

Clock Input/Output (OSCIN, OSCOUT): These two pins connect the main timing reference (crystal) to the SPC.

ROM Data Timing

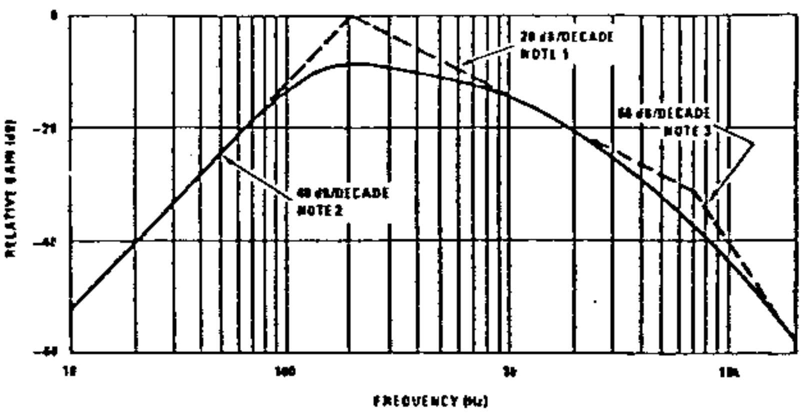


Dual-In-Line Package



64k

Frequency Response of Combined Amplifier and Speaker

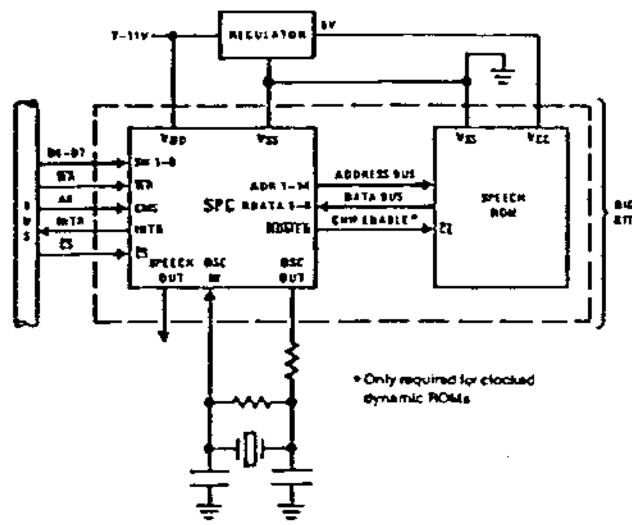


Note 1: This curve is the desired response of the entire audio system including speaker. Minimum response is a low pass filter with a cutoff frequency of 200 Hz. For an audio system with a natural cutoff frequency around 200 Hz, this filter can be eliminated. This cutoff frequency may be tuned for the particular voice being synthesized. For a low pitched male voice it may be 100 Hz, while for a high pitched female or child's voice it might be 300 Hz.

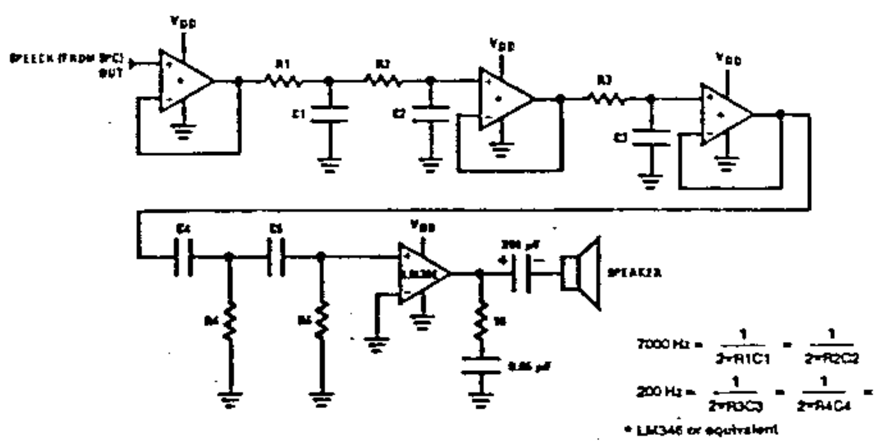
Note 2: This is optional filtering that can be eliminated by proper selection of the speaker. If this 2 pole response is electronically produced, it should be adjusted as described in Note 1.

Note 3: This is optional filtering that can be eliminated for simpler systems. The acceptable range for this cutoff frequency is 6000 Hz–8000 Hz.

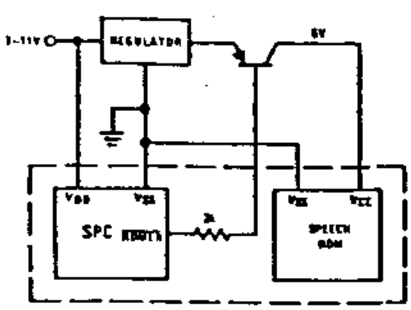
Complete Applications Schematic for High Quality Voice Reproduction



Filter Circuit to Produce Maximum Frequency Response



Low Power Configuration Using Static ROM



Minimum Filter Circuit