REVISIONS

REV	SYM	DATE	CN	SHT	DESCRIPTION	BY	CH	APP
					Rev. B December 30, 1980 - Rewrite by P. McLaughlin			
					Rev. C November 18, 1981 - Update by L. Davidoff			

SPEC. NO.

SP-0256 Design Objective Specification
(Internal Use Version not for Circulation)

SHEET													
LAST REV													

SHEET													
LAST REV													

DISTRIBUTION LIST	GENERAL INSTRUMENT MICROELECTRONICS GROUP	PLANT
SUPERSEDES		MODULE
SUPERSEDED BY	TITLE SP-0256 Design Objective Specification	OPERATION

	WRITTEN	APPROVED						
BY	DAVIDOFF						SHEET 1 OF	
DATE	11-18-81						SPEC. NO. SP-0256	REV C

FORM 1E104 (3/81)

PRODUCT PURPOSE

The SP 0256 speech processor is an N-Channel Metal Gate LSI device that is intended for product applications where synthetic speech or complex sounds are required.

The SP device serves as a fixed repertoire speech and sound synthesizer which is capable of reproducing up to 256 discrete sound sequences. Each sequence may be called by loading its 8 bit address into the command register of the device. The sequence data is located in mask programmable ROM, which enables the user to specify the speech or sound pattern desired. For applications requiring a larger yocabulary than can be accommodated in the 16K bits of on board ROM, external ROM may be used to extend the devices capabilities.

Through the addition of external ROMS the system may be expanded to directly address 491K bits of memory, and up to 3825 sequences (usually words or phrases) which may be called directly from the input port using a double byte load.

GENERAL DEVICE FUNCTION

The internal organization of SP, enables a large quantity of speech, or sound to be specified in a modest amount of pattern ROM. In addition, the flexible architecture of the on-board controller allows the user to partition the available storage space into as many sequences as desired.

From a functional standpoint, the device can be divided into two major sections, the controller, and the vocal tract model (VTM). The VTM is a parametric sound and voice snythesizer which produces complex waveforms under the control of 17 slowly time varying parameters.

The controller executes its internal ROM instructions and modifies the appropriate parameters of the VTM to create the desired sound sequence.

The interface between the controller and the VTM is accomplished through the parameter registers and related timing signals.

Since the number of bytes of data used per second of speech is variable, the user is able to trade off speech quality for vocabulary size when it is desirable to do so. High quality continuous speech requires about 2000 bits per second, while lower quality understandable speech can be coded at considerably lower bit rates.

DEVICE OPERATION

SP VTM Filter Structure

The SP device models speech (and other sounds) using a series of six variable 2nd order resonators excited by either a pseudo noise source, or a periodic impulse source.

The VTM is implemented using totally digital techniques. This approach allows one 2nd order section to serve as six sections through the use of multiplexing and information line pipelining. The section that is implemented is the 2nd order infinite impulse response (IIR) digital filter shown in Figure 1. This filter stage has the transfer function:

$$H(Z) = \frac{1}{1 - 2F_t Z^{-1} - B_t Z^{-2}}$$

Therefore it can be shown that the poles of the transfer function occur at:

$$\frac{-2F_t \pm \sqrt{4F_t^2 + 4B_t}}{-2}$$

and when,

$$-1 < B_t \leq 0$$

and,

$$|F_t| \leq \sqrt{-B}$$

the poles will be placed in a complex pair, forming a resonator with the bandwidth given by:

(1) $$B.W. = \frac{-Fs\ LN\ (B_t^2)}{\pi}$$

where Fs is the sampling frequency in HZ.

and the center frequency (Fk) given by:

(2) $$Fk = \frac{Fs\ COS^{-1}\left[\frac{2F_t}{\sqrt{-B_t}}\right]}{2\pi}$$

As can be seen from equations 1 and 2 above, the modification of the B coefficient changes both the frequency and the bandwidth of the resonator. The modification of the F coefficient, however changes only the center frequency, and has no effect on the corresponding bandwidth.

Since speech signals (in particular vowel sounds) convey information through the shifting of resonant peaks in the spectrum, it is desirable to be able to change center frequencies of the 2nd order stages independently of their respective bandwidth settings. In addition it is important that the parameters of the individual stages (corresponding to particular resonances) can be modified independently. The use of cascade 2nd order stages supports these features, giving this configuration a distinct advantage over other filter sections currently in use for speech synthesis, such as the Lattice section, and the direct form implementation. The instruction set of the SP controller section is designed to exploit the ability of the VTM parameters to be updated selectively to achieve a greater packing density in the ROM. In addition, this permits the user to trade-off between quality and quantity of speech samples possible within the ROM space available.

If it is desired to place resonances at a frequency of zero, these real axis poles can be accomodated directly. Each 2nd order stage may be used to place two real axis poles of variable bandwidth. If X_1 is the real axis location of the first pole, and X_2 the second:

$$F_t = \frac{X_1 + X_2}{2}$$

and:

$$B_t = (F - X_1)^2 - F^2$$

with the bandwidths of each given by:

$$Bw_1 = -2Fs \; LN \; X_1$$
$$BW_2 = -2Fs \; LN \; X_2$$

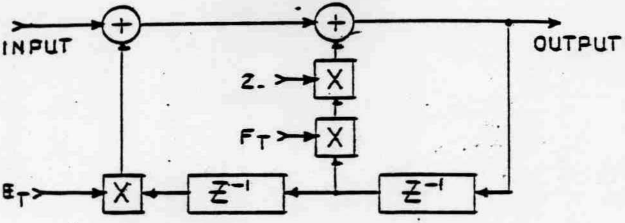
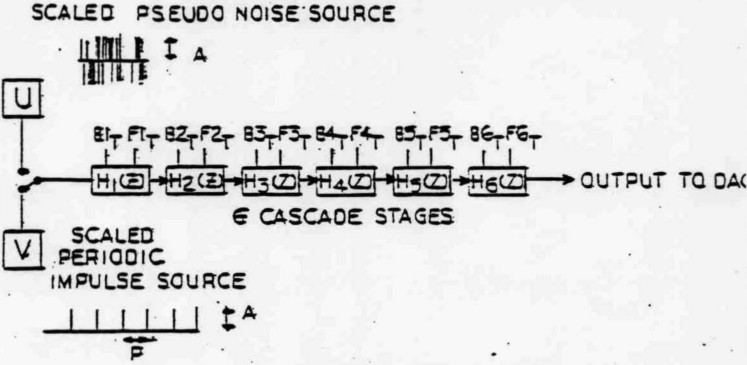


DIAGRAM OF SINGLE CASCADE STAGE

$$H_N(Z) = \frac{1}{1 - 2\,F_T\,Z^{-1} - B_T\,Z^{-2}}$$


SCALED PSEUDO NOISE SOURCE
A
U
$B1_T$ $F1_T$ $B2_T$ $F2_T$ $B3_T$ $F3_T$ $B4_T$ $F4_T$ $B5_T$ $F5_T$ $B6_T$ $F6_T$
$H_1(Z)$
$H_2(Z)$
$H_3(Z)$
$H_4(Z)$
$H_5(Z)$
$H_6(Z)$
OUTPUT TO DAC
6 CASCADE STAGES
V
SCALED PERIODIC IMPULSE SOURCE
A
P

$V(Z) = (H_1(Z))\,(H_2(Z))\,(H_3(Z))\,(H_4(Z))\,(H_5(Z))\,(H_6(Z))$

FOR IMPULSE SOURCES

OUTPUT = A V(Z) (VALID BETWEEN IMPULSES)

FIGURE 1

where F_t and B_t represent the coefficients in Figure 1.

Coefficient updates to the filter occur at the beginning of a pitch period. This timing results in the smallest possible distrubance to the output at update.

The information line precision is maintained at 16 bits throughout the VTM filter.

The multiply by 2 shown in Fig. 1 is implemented as a 1 bit binary shift following the F_t multiplier. The shift operation is performed separately from the multiplication to scale F_t to the same range of values as B_t.

The coefficients F_t and B_t are quantized non-linearly to minimize coefficient. sensitivity.

The two coefficients are processed by the same non-linear transformation hardware in the range.

$$-1 < C < 1$$

where C may be either F_t or B_t.

The non-linear transformation T(X) is implemented with a table lookup ROM.

The input coefficients of each stage, denoted as F and B, are expressed in sign magnitude form and used to generate the multiplier coefficients as follows:

$$F_t = T(F)$$
$$B_t = T(B)$$

B F	B_t X 512 F_t X 512	B_t F_t
0	0	0.000000
1	-9	-.017578
2	-17	-.033203
3	-25	-.048828
4	-33	-.064453
5	-41	-.080078
6	-49	-.095703
7	-57	-.111328
8	-65	-.126953
9	-73	-.142578
10	-81	-.158203
11	-89	-.173828
12	-97	-.189453
13	-105	-.205078
14	-113	-.220703
15	-121	- 236328
16	-129	-.251953
17	-137	-.267578
18	-145	-.283203
19	-153	-.298828
20	-161	-.314453
21	-169	-.330078
22	-177	-.345703
23	-185	-.361328
24	-193	-.376953
25	-201	-.392578
26	-209	-.408203
27	-217	-.423828
28	-225	-.439453
29	-233	-.455078
30	-241	-.470703
31	-249	-.486328
32	-257	-.501953
33	-265	-.517578
34	-273	-.533203
35	-281	-.548828
36	-289	-.564453
37	-297	-.580078
38	-301	-.587891
39	-305	-.595703
40	-309	-.603516
41	-313	-.611328
42	-317	-.619141.
43	-321	-.626953
44	-325	-.634766
45	-329	-.642578
46	-333	-.650391
47	-337	-.658203
48	-341	-.666016
49	-345	-.673828
50	-349	-.681641
51	-353	-.689453
52	-357	-.697266
53	-361	-.705078
54	-365	-.712891
55	-369	-.720703
56	-373	-.728516
57	-377	-.736328
58	-381	-.744141

B F	Bt X 512 Ft X 512	Bt Ft
59	-385	-.751953
60	-389	-.759766
61	-393	-.767578
62	-397	-.775391
63	-401	-.783203
64	-405	-.791016
65	-409	-.798828
66	-413	-.806641
67	-417	-.814453
68	-421	-.822266
69	-425	-.830078
70	-427	-.833984
71	-429	-.837891
72	-431	-.841797
73	-433	-.845703
74	-435	-.849609
75	-437	-.853516
76	-439	-.857422
77	-441	-.861328
78	-443	-.865234
79	-445	-.869141
80	-447	-.873047
81	-449	-.876953
82	-451	-.880859
83	-453	-.884766
84	-455	-.888672
85	-457	-.892578
86	-459	-.896484
87	-461	-.900391
88	-463	-.904297
89	-465	-.908203
90	-467	-.912109
91	-469	-.916016
92	-471	-.919922
93	-473	-.923828
94	-475	-.927734
95	-477	-.931641
96	-479	-.935547
97	-481	-.939453
98	-482	-.941406
99	-483	-.943359
100	-484	-.945313
101	-485	-.947266
102	-486	-.949219
103	-487	-.951172
104	-488	-.953125
105	-489	-.955078
106	-490	-.957031
107	-491	-.958984
108	-492	-.960938
109	-493	-.962891
110	-494	-.964844
111	-495	-.966797
112	-496	-.968750
113	-497	-.970703
114	-498	-.972656
115	-499	-.974609
116	-500	-.976563
117	-501	-.978516
118	-502	-.980469

B/F	Bt X 512 / Ft X 512	Bt/Ft
119	-503	-.982422
120	-504	-.984375
121	-505	-.986328
122	-506	-.988281
123	-507	-.990234
124	-508	-.992188
125	-509	-.994141
126	-510	-.996094
127	-511	-.998047
128	0	0.000000
129	9	.017578
130	17	.033203
131	25	.048828
132	33	.064453
133	41	.080078
134	49	.095703
135	57	.111328
136	65	.126953
137	73	.142578
138	81	.158203
139	89	.173828
140	97	.189453
141	105	.205078
142	113	.220703
143	121	.236328
144	129	.251953
145	137	.267578
146	145	.283203
147	153	.298828
148	161	.314453
149	169	.330078
150	177	.345703
151	185	.361328
152	193	.376953
153	201	.392578
154	209	.408203
155	217	.423828
156	225	.439453
157	233	.455078
158	241	.470703
159	249	.486328
160	257	.501953
161	265	.517578
162	273	.533203
163	281	.548828
164	289	.564453
165	297	.580078
166	301	.587891
167	305	.595703
168	309	.603516
169	313	.611328
170	317	.619141
171	321	.626953
172	325	.634766
173	329	.642578
174	333	.650391
175	337	.658203
176	341	.666016
177	345	.673828
178	349	.681641

$\frac{B}{F}$	$\frac{B_t \times 512}{F_t \times 512}$	$\frac{B_t}{F_t}$
179	353	.689453
180	357	.697266
181	361	.705078
182	365	.712891
183	369	.720703
184	373	.728516
185	377	.736328
186	381	.744141
187	385	.751953
188	389	.759766
189	393	.767578
190	397	.775391
191	401	.783203
192	405	.791016
193	409	.798828
194	413	.806641
195	417	.814453
196	421	.822266
197	425	.830078
198	427	.833984
199	429	.837891
200	431	.941797
201	433	.845703
202	435	.849609
203	437	.853516
204	439	.857422
205	441	.861328
206	443	.865234
207	445	.869141
208	447	.873047
209	449	.876953
210	451	.880859
211	453	.884766
212	455	888672
213	457	.892578
214	459	896484
215	461	.900391
216	463	.904297
217	465	.908203
218	467	.912109
219	469	.916016
220	471	.919922
221	473	.923828
222	475	.927734
223	477	.931641
224	479	.935547
225	481	.939453
226	482	.941406
227	483	.943359
228	464	.945313
229	485	.947266
230	486	.949219
231	487	.951172
232	488	.953125
233	489	.955078
234	490	.957031
235	491	.958984
236	492	960938
237	493	.962891
238	494	.964844

$\frac{B}{F}$	$\frac{B_t \text{ X } 512}{F_t \text{ X } 512}$	$\frac{B_t}{F_t}$
239	495	.966797
240	496	.968750
241	497	.970703
242	498	.972656
243	499	.974609
244	500	.976563
245	501	.978516
246	502	.980469
247	503	.982422
248	504	.984375
249	505	.986328
250	506	.988281
251	507	.990234
252	508	.992188
253	509	.994141
254	510	.996094
255	511	.998047

DEVICE ARCHITECTURE

VTM Source Timing

The VTM operates under control of the 17 parameter registers listed in Table 1. The duration and pitch of the sounds produced by the SP device are controlled by the R and P registers respectively. P specifies the number of sample periods in one pitch period.

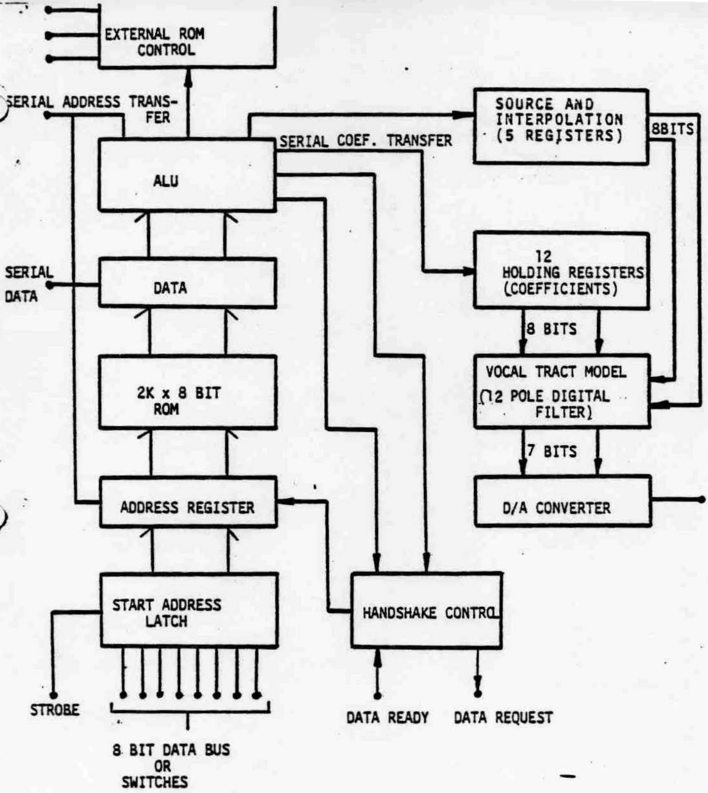
Expressed in terms of the pitch (Fo) and the sampling frequency (FS)

$$P = \frac{FS}{Fo} \qquad 1 \leq P \leq 255$$

The pitch source injects unit impulses, spaced P sample periods apart. The R register (repeat count) is the number of pitch cycles executed before a register update occurs.

When P is assigned a zero value, the pitch source is replaced by a zero mean, pseudo noise source. This mode of operation is referred to as unvoiced mode. In this mode the system requests a register update after 64 times R samples. The amplitude of the source is controlled by the A register. It is coded as 5 bits of mantissa and 3 bits of exponent (i.e. binary shift).

At the completion of the pitch period, the value stored in the AI register is added to the A register contents, and similarly the PI register is added to the P register. This allows the pitch and amplitude of the frame of speech to be smoothly interpolated between updates.



SP 0256 BLOCK DIAGRAM

Figure 2

TABLE 1 - VTM PARAMETER REGISTERS

R	6 bits	Repeat Count (Number of Pitch Periods)
A	8 bits	Source Amplitude
P	8 bits	Pitch Period (Number of Sampling Periods)
B1	8 bits	B Coefficient of Filter Stage 1
F1	8 bits	F Coefficient of Filter Stage 1
B2	8 bits	B Coefficient of Filter Stage 2
F2	8 bits	F Coefficient of Filter Stage 2
B3	8 bits	B Coefficient of Filter Stage 3
F3	8 bits	F Coefficient of Filter Stage 3
B4	8 bits	B Coefficient of Filter Stage 4
F4	8 bits	F Coefficient of Filter Stage 4
B5	8 bits	B Coefficient of Filter Stage 5
F5	8 bits	F Coefficient of Filter Stage 5
B6	8 bits	B Coefficient of Filter Stage 6
F6	8 bits	F Coefficient of Filter Stage 6
AI	8 bits	Amplitude Interpolation Delta
PI	8 bits	Pitch Period Interpolation Delta

SP Controller

The SP Controller is a sequential processor which fetches instructions and data from ROM, and is capable of altering the contents of the 17 parameter registers controlling the SP VTM. The controller has 16 executable instructions, and supports 1 level of subroutine nesting. The instruction set is designed specifically to allow selective updates of the parameter registers to be performed. In addition,the JMP and JSR instructions allow chaining of segments, and sharing of code sequences to eliminate redundancy.

The SP instruction set consists of two groups of instructions, register modification instructions, and branch control instructions.

PROGRAMMABILITY

Register Modification Instructions

The purpose of the SP register modification instructions is to update the VTM parameters.

The R and P registers determine how many sample periods of a particular sound are output by the VTM before control of the parameter registers is returned to the controller. The controller waits until the completion of the last of R pitch periods (or 64 X R samples in unvoiced mode) before executing the next register modification instruction.

Each of the 12 register modification instructions, with the exception of RCU, consists of a 4 bit op code followed by 4 bits of data which are loaded into the lower 4 bits of register R. RCU is a 1 byte instruction which loads the upper 2 bits of the Repeat register (i.e. Register R), the Precision Select Bit (P), and the Filter Order Select Bit (M).

RCU passes control to the next instruction following execution. RCU will not cause an immediate transfer of control to the VTM.

The Precision Select Bit determines which of the two bit precision maps stored in the SP0256 is to be used in updating the parameter registers. When the P Bit is set high, a greater number of bits are used to represent the parameters which renders the highest quality speech. Setting the P Bit low selects a bit precision map which calls for fewer bits to specify the parameters, offering a bit savings at a slight speech quality loss, which is often imperceptible. The Filter Order Select Bit (M Bit) selects between a 10 pole VTM configuration, and a 12 pole configuration. In the 10 pole mode, the F6 and B6 registers are set to zero and not loaded by any of the register modification instructions. When synthesizing speech sections that can be modeled with only 10 poles without a loss in quality, this configuration may be used to reduce bit rate. The 12 pole configuration is selected by setting the M Bit high.

The register modification instructions fall into two groups, absolute load instructions, and Delta coded load instructions.

The absolute load instructions replace the current values of the parameter registers to be updated with new data pulled from the Speech ROM. The number of bits used to represent each parameter and their placement within the 8 bit register word is determined by the Bit Precision Mapping ROM inside the SP0256 Controller section.

The Delta coded load instructions operate similarly to the absolute load instructions with the exception that the variable length data word which is pulled from the ROM for each parameter register to be updated, is added to the previous value of the parameter instead of replacing it. A different bit precision map is employed for Delta loads. In all cases, the Delta load instructions require less bits than their absolute equivalents.

All parameter modification data is expressed in twos complement form.

REGISTER MODIFICATION INSTRUCTIONS

MNEMONIC	INSTRUCTION CODE $N_3N_2N_1N_0$=Lower 4 Bits of R	REGISTERS MODIFIED
RCU	0 0 0 1 M P N_5 N_4	R, (N_5,N_4 are upper 2 bits of R)
PRL	0 0 1 0 $R_3R_2R_1R_0$	R,A,P,B4,F4,B5,F5, B6,F6 ZERO → B1,F1,B2,F2, B3,F3,AI,PI
FRL	0 0 1 1 $R_3R_2R_1R_0$	R,A,P,B1,F1,B2,F2, B3,F3,B4,F4,B5,F5, B6,F6 ZERO → AI,PI
FRL(API)	0 1 0 0 $R_3R_2R_1R_0$	R,A,P,B1,F1,B2,F2, B3,F3,B4,F4,B5,F5 B6,F6,AI,PI
FFL	0 1 0 1 $R_3R_2R_1R_0$	R,A,F1,F2,F3 ZERO → AI,PI
FFU	0 1 1 0 $R_3R_2R_1R_0$	R,A,F4,F5,F6 ZERO → AI,PI
PCT	0 1 1 1 $R_3R_2R_1R_0$	R,A,P, ZERO → AI,PI
FPL (API)	1 0 0 0 $R_3R_2R_1R_o$	R,A,P,B1,F1,B2,F2, B3,F3,B4,F4,B5,F5 B6,F6,AI,PI (All parameters with the exception of R are loaded with full 8 bit precision for this instruction only)
DFRL	1 0 0 1 $R_3R_2R_1R_0$	FRL instruction with parameters except R, Delta coded

REGISTER MODIFICATION INSTRUCTIONS

PNEMONIC	INSTRUCTION CODE $N_3N_2N_1N_0$ = Lower 4 Bits of R	REGISTERS MODIFIED
FFLP	1 0 1 0 R_3 R_2 R_1 R_0	R,A,P,F1,F2,F3 Zero ⟶ AI,PI FFL with Pitch
DPRL	1 0 1 1 R_3 R_2 R_1 R_0	Delta coded PRL Instruction
FFL(API)	1 1 0 0 R_3 R_2 R_1 R_0	R,A,F1,F2,F3,AI,PI
SIL	1 1 1 1 R_3 R_2 R_1 R_0	R Zero ⟶ P,A,AI,PI

BRANCH CONTROL INSTRUCTIONS

MNEMONIC	INSTRUCTION CODE	# OF BYTES	REGISTERS MODIFIED
JSR	1 1 0 1 A_8 A_9 A_{10} A_{11} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7	2	PC → RB, A_0 thru A_{15} → PC, Return Flag Set
RET	0 0 0 0 0 0 0 0	1	If Return Flag Set RB → PC, Return Flag Reset If Return Flag not Set IB → PC 0 → IB IBF Reset
JMP	1 1 1 0 A_8 A_9 A_{10} A_{11} A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7	2	A_0 thru A_{15} → PC
PAG	0 0 0 0 A_{12} A_{13} A_{14} A_{15}	1	On Execution of next JSR or JMP instruction $A_{15}A_{14}A_{13}A_{12}$ → PC (A_{12} thru A_{15} must be greater than zero)

HIGH PRECISION (P = 1) BIT MAP

PARAMETER	BIT PLACEMENT ABSOLUTE	BIT PLACEMENT DELTA
F1	D D D D D D X X	X X D D D D X X
B1	X D D D D D D X	X X X D D D D X
F2	D D D D D D X X	X X D D D D X X
B2	X D D D D D D X	X X X D D D D X
F3	D D D D D D X X	X X D D D D X X
B3	X D D D D D D X	X X X D D D D X
F4	D D D D D D D X	X X D D D D D X
B4	X D D D D D D X	X X X D D D D X
F5	D D D D D D D D	X X X D D D D D
B5	D D D D D D D D	X X X D D D D D
F6	D D D D D D D D	X X X D D D D D
B6	D D D D D D D D	X X X D D D D D
P	D D D D D D D D	X X X D D D D D
A	D D D D D D X X	X X D D D D X X
PI	X X X D D D D D	X X X D D D D D
AI	X X X D D D D D	X X X D D D D D

(D INDICATES PLACEMENT OF DATA BITS IN 8 BIT WORD)

LOW PRECISION (P = 0) BIT MAP

PARAMETER	BIT PLACEMENT ABSOLUTE	BIT PLACEMENT DELTA
F1	D D D D D X X X	X X D D D X X X
B1	X D D D X X X X	X D D D X X X X
F2	D D D D D X X X	X X D D D X X X
B2	X D D D X X X X	X D D D X X X X
F3	D D D D D X X X	X X D D D X X X
B3	X D D D X X X X	X D D D X X X X
F4	D D D D D D X X	X X D D D D X X
B4	X D D D D X X X	X X D D D X X X
F5	D D D D D D X X	X X D D D D X X
B5	D D D D D D D X	X X X D D D D X
F6	D D D D D D D D	X X X D D D D D
B6	D D D D D D D D	X X X D D D D D
P	D D D D D D D D	X X X D D D D D
A	D D D D D D X X	X X D D D D X X
PI	X X X D D D D D	X X X D D D D D
AI	X X X D D D D D	X X X D D D D D

(D INDICATES PLACEMENT OF DATA BITS IN 8 BIT WORD)

Instruction Chaining

In the case where more than one register modification instruction is required to update the parameter registers, yet it is not desired to perform a full update (FRL), 2 or more instructions may be chained together. This feature is also useful for initializing the registers before a JMP, JSR, or return instruction, without causing the VTM to start a sound sequence.

If an instruction is to be chained with one or more other instructions, the lower 4 bits of its' instruction byte are set to zero. The last instruction in a series of chained instruction bytes, is the only one with a non-zero lower 4 bits. The lower 4 bits of the last instruction byte in a chained series is loaded into the lower 4 bits of the R register. When chaining instructions, the data bytes appear after the last instruction. The order of the data bytes is given by Table 3.

For example, if it was desired to chain an FFU with an FFL instruction, the following registers would be updated:

FFU Modifies F4, F5, F6, R, A

FFL Modifies F1, F2, F3, R, A

Referring to Table 3, it can be seen that the proper sequence for the above data is:

A, F1, F2, F3, F4, F5, F6

If we assume that low precision mode has previously been selected, the proper bit string to execute the entire sequence would be:

TABLE 3

DATA BYTE LOAD SEQUENCE FOR CHAINED INSTRUCTIONS

Loaded First	1	A
	2	P
	3	B1
	4	F1
	5	B2
	6	F2
	7	B3
	8	F3
	9	B4
	10	F4
	11	B5
	12	F5
	13	B6
	14	F6
	15	AI
Loaded Last	16	PI

FFU	0	1	1	0	0	0	0	0
FFL	0	1	0	1	R_3	R_2	R_1	R_0
A	D	D	D	D	D	D		
F1	D	D	D	D	D			
F2	D	D	D	D	D			
F3	D	D	D	D	D			
F4	D	D	D	D	D	D		
F5	D	D	D	D	D	D		
F6	D	D	D	D	D	D	D	D

chain

load Repeat Register

See Loc [illegible] bit maps.

(D REPRESENTS DATA BITS CORRESPONDING TO PARAMETER)

Where R_3 R_2 R_1 R_0 represent the lower 4 bits of the repeat register (R). Since the instructions are both absolute load instructions the data will be read from the ROM using the bit placement absolute map. As can be seen from the above, this chained sequence requires 57 bits of ROM storage. The first bit of the next instruction would be the ROM bit following the above bit stream. Note that the beginning and end of an instruction sequence may occur anywhere within a byte boundary.

HOW TO PACK ROM DATA

ASSUME THE DATA SEQUENCE ON PAGE 25. ASSUME A JUMP INSTRUCTION FOLLOWS. IF UNPACKED THE DATA WOULD BE AS FOLLOWS:

FFU	0	1	1	0	0	0	0	0
FFL	0	1	0	1	R_3	R_2	R_1	R_0
A	A_5	A_4	A_3	A_2	A_1	A_0		
F1	F_{14}	F_{13}	F_{12}	F_{11}	F_{10}			
F2	F_{24}	F_{23}	F_{22}	F_{21}	F_{20}			
F3	F_{34}	F_{33}	F_{32}	F_{31}	F_{30}			
F4	F_{45}	F_{44}	F_{43}	F_{42}	F_{41}	F_{40}		
F5	F_{55}	F_{54}	F_{53}	F_{52}	F_{51}	F_{50}		
F6	F_{67}	F_{66}	F_{65}	F_{64}	F_{63}	F_{62}	F_{61}	F_{60}
JMP	1	1	1	0	A_8	A_9	A_{10}	A_{11}
	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7

WHEN PACKED, THE DATA WOULD BE AS FOLLOWS:

0	1	1	0	0	0	0	0
0	1	0	1	R_3	R_2	R_1	R_0
F_{11}	F_{10}	A_5	A_4	A_3	A_2	A_1	A_0
F_{24}	F_{23}	F_{22}	F_{21}	F_{20}	F_{14}	F_{13}	F_{12}
F_{42}	F_{41}	F_{40}	F_{34}	F_{33}	F_{32}	F_{31}	F_{30}
F_{54}	F_{53}	F_{52}	F_{51}	F_{50}	F_{45}	F_{44}	F_{43}
F_{66}	F_{65}	F_{64}	F_{63}	F_{62}	F_{61}	F_{60}	F_{55}
1	1	0	A_8	A_9	A_{10}	A_{11}	F_{67}
A_1	A_2	A_3	A_4	A_5	A_6	A_7	1
							A_0

→

DATA IS SHIFTED OUT TO THE RIGHT. JUMP INSTRUCTIONS COME OUT MSB FIRST. ALL DATA AND OTHER INSTRUCTIONS COME OUT LSB FIRST.

Branch Control Instructions

The SP Branch Control instructions differ from the register modification instructions in that they do not modify any of the VTM parameter registers. The sole purpose of these instructions is to determine the location in the ROM from which the next instruction will be fetched. The JSR (Jump to Subroutine) instruction stores the present address (ie. the contents of the PC register) in the return buffer (RB) register. The PC register is loaded with the 12 bit address specified by the last 4 bit of the instruction byte and the following data byte. In addition, an internal return flag is set to indicate that the RB register has been loaded. The controller then fetches and executes the instruction located at PC + 1 in the ROM. Only one level of subroutine is allowed.

The JMP (Jump) instruction loads the PC register with the 12 bit address specified by the lower 4 bits of the instruction byte, and the following data byte. Neither the return flag nor the return buffer are modified. Upon completion of execution, the next instruction is fetched from location PC + 1 in the ROM.

RET (Return From Subroutine) is an instruction whose function depends on the state of the return flag. When the return flag is set (indicating that a subroutine is being executed), execution of an RET instruction will cause the contents of the RB register to be moved into the PC. The return flag is reset, and the controller fetches the instruction located at PC + 1, and continues execution from that location. Only one level of subroutine is allowed. When an RET instruction is encountered and the return flag is not set, the status of the input buffer flag (IBF) is checked. If IBF is set, (indicating that the starting address of the next sound sequence has been loaded into the SP device) the contents of the IB register (8 bits) is loaded into the PC.

If the IBF flag is not set, the controller will disable any further output from the VTM, and wait for the IBF flag to become set. The SBY (standby) pin will go high and remain high until the IBF flag is set. The SBY pin is discussed further in the section on standby operation. When the IBF flag is set, execution continues as described above.

PAG is a one byte instruction used to perform Jumps to addresses in memory outside of the present page (4K byte block). The lower 4 bits of the instruction byte specifies the upper 4 bits of the address in the next JMP or JSR instruction encountered. The argument of the PAG instruction must be greater than zero since the base page in the system is Page 1. If a JMP or JSR instruction is executed without a PAG instruction preceeding it, the upper 4 bits of the JMP address are set to the upper 4 bits of the PC. The fact that the PAG instruction is optional allows a 2 byte short form address to be used for short jumps, and a three byte extended Jump consisting of a PAG instruction and a JMP or JSR, to be used for Jumps outside the present page.

Although instruction sequences may begin and end within byte boundaries, JMP, JSR, and RET instructions always cause a branch to a byte boundary. For example, if a JSR instruction had its last bit stored in the 4th bit position of byte number 125_8 in ROM, and the instruction called for a Jump to location 250, the next instruction to be executed would be read starting with the first bit of location

250. If a return instruction was executed, the next instruction would be read starting with the first bit of location 126_8, not the fifth bit of location 125_8.

DEVICE INTERFACE

Program Entry Control

In the SP System, the individual sound sequences stored in ROM, are accessed by use of the 8 bit input port. The significance of this input byte is a function of the state of the SE (Strobe Enable) input pin. When the SE line is high, and the input port has been loaded from the external system the contents of the 8 bit input port is loaded into the 8 bit IB register. This allows any one of 256 entry points to be specified with a one byte input. The entry points are spaced at 2 byte increments (ie. 0, 2, 4, 6....) throughout the internal ROM. Two byte inputs may be used to expand the number of entry points to 3825. This addressing scheme is most useful in applications where the SP is interfaced to the data bus of an external microprocessor, or in other environments where the use of a strobe line may be desirable.

In applications where an appropriate strobe is not available (such as a stand-alone environment with switches selecting the desired sequence) the SP 0256 can latch its data without an external strobe. When the SE input is tied low, the SP 0256 will latch in the data on the input bus approximately 1usec.after the detection of a high level on any one of the 8 address input lines.

Input Handshake Control

Input to the SP device is accomplished using 8 input data pins, 2 handshake lines, and an input mode select pin. As mentioned in the previous section, the handshake pins are not necessary for some applications. When the SE (Strobe Enable) pin is kept high the handshake lines (LRQ and DLD) are used to coordinate the data input. LRQ (Load Request) is an output pin which is low whenever the IBF (Input Buffer Flag) is set. When LRQ is high, the input port is loaded by placing the 8 bits of data on the input lines, and pulsing the DLD (Data Load) input. The rising edge of DLD will cause LRQ to go low, where it will remain until the internal IBF Flag is reset by an RET instruction.

DAC Output

The output of the SP VTM section drives an internal 7 bit pulse width modulation (PWM) digital to analog converter. The design of the PWM DAC is such that all noise components are at or above 10KHz. The output is low pass filtered to 5KHz, and amplified externally.

Standby Operation

The SP 0256 has two power supply pins and a common ground. The VD1 supply pin powers the interface logic and provides standby current to the controller and parameter registers. The VDD pin powers the VTM, the controller and the internal ROM. When the SBY pin is high (indicating that the SP is inactive) the VDD pin can be powered down externally to conserve power. This will provide a standby current which is a fraction of the normal operating current.

When the SP is loaded with an entry byte, the SBY pin is brought low signaling the external circuitry to power-up the VDD pin. The SP will delay execution of the selected sequence to allow the power supply to settle. If it is not desired to implement the standby mode of operation, the VD1 and VDD pins should be tied to a common supply.

Clocks

The SP 0256 requires one 3.12MHz clock, which is generated by an onboard oscillator with external crystal control. The crystal is connected between the XTAL in, and XTAL out pins.

Vocabulary Expansion

The SP 0256 is capable of directly addressing 61,440 bytes (8 bits wide) of ROM. The devide has 2K bytes of ROM on board. If additional vocabulary space is required, external ROM may be added to the system.

The external ROM interfaces to the SP device through the use of the serial in, and serial out lines. The serial out pin provides the 16 bit address to the external ROMS. The addressed ROM sends serial data back to the SP 0256 through the serial in pin. To accommodate this serial communication, special purpose ROMS are used. These expansion ROMS are controlled using the C_1, C_2, and C_3 outputs of the SP 0256. The architecture of the SP is such that a 16 bit address need be output only upon execution of a JMP or JSR instruction. External ROM interface timing is shown in Figure 3. Table 4 lists the control functions of the vocabulary expansion ROM, and the action taken by the ROM. FIgure 4 shows the interface between the SP 0256 and two SPR-16 ROMS.

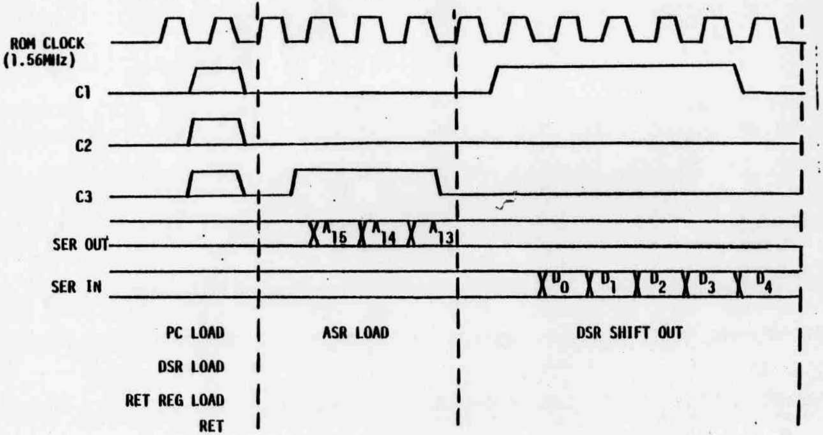
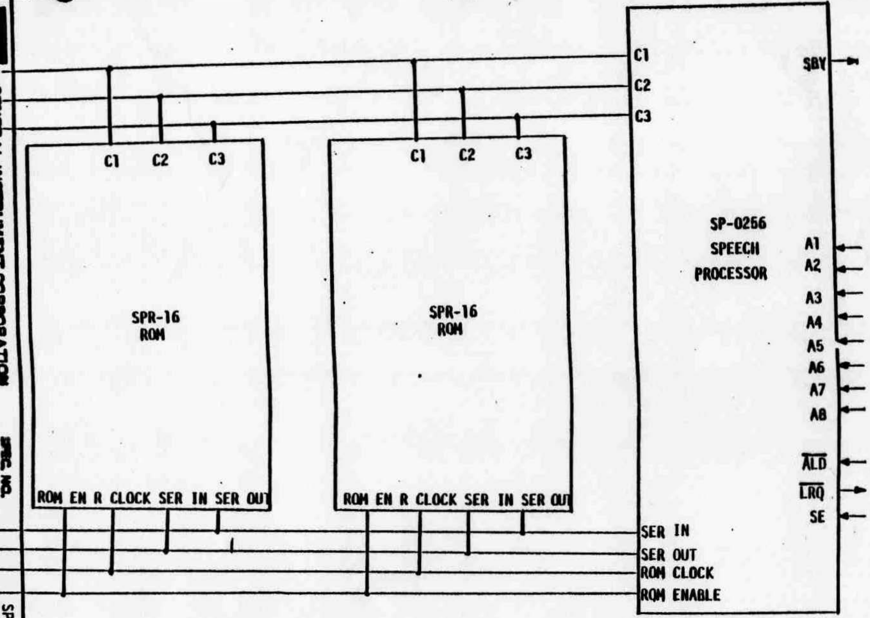


FIGURE 3 EXTERNAL ROM INTERFACE TIMING

TABLE 4 EXTERNAL SPEECH ROM CONTROL STATE

C1	C2	C3	FUNCTION
0	0	0	NOP - No Action Taken
0	0	1	ASR Load - Accepts data from the serial output synchronous to the externally supplied ROM clock. This data is shifted into the ASR holding register in preparation for loading into the PC. Although ASR is 16 bits long, it is not necessary to load all 16 bits of address sequentially in one ASR load.
0	1	0	PC Load - Loads the contents of the ASR register into the PC.
0	1	1	DSR Load - Loads the 8 bits of data pointed to by the present value of the least significant 11 bits of the PC into the data output shift register (DSR). At the completion of the DSR load the PC is incremented.
1	0	0	DSR Shift Out - Shifts out the contents of DSR to the serial in pin, synchronous to the ROM clock.
1	0	1	RET. Register Load - Loads the return register (RB) with the current value of the P.C.
1	1	0	Return - Loads the PC with the contents of the RB register.
1	1	1	NOP - No action taken in Speech ROMS. Does not exist in SP-0256.



INTERFACE OF SPR-16 ROM TO SP-0256 SPEECH PROCESSOR

Document Control # 0380-20

5. Electrical Characteristics

Maximum Ratings

All pins with respect to VSS ---------------------------- 0.3 to 8.0V
Storage Temperature ------------------------------------ -25°C to 125°C

Standard Conditions

Clock - Crystal Frequency ------------------------------ 3.120MHz
Operating Temperature ---------------------------------- 0°C to 70°C

DC CHARACTERISTICS

Name	Sym	Min	Typ	Max	Units	Conditions
Primary Supply Voltage	VDD	4.6	-	7.0	V	
Standby Supply Voltage	VDI	4.6	-	7.0	V	
Primary Supply Current	IDD	-	-	90	ma	25°C No loads. Reset & STBY. Reset high. All other inputs floating 105 maMAX @ 0°C 75ma MAX @ 70°C
Standby Supply Current	IDI	-	-	21	ma	25°C same as above. 24.0 ma MAX @ 0°C 17.5ma MAX@70°C
INPUTS						
A1-A8, ALD, SERIN, TEST, SE.						
LOGIC 0	VIL	0.0	-	0.6	V	
LOGIC 1	VIH	2.4	-	VDI	V	
CAPACITANCE	CIN	-	-	10	pf	
LEAKAGE	ILC	-	-	±10	ua	Vpin = 7.0V Other Pins = 0.0V
RESET, SBY RESET						
LOGIC 0	VRSIL	0.0	-	0.6	V	
LOGIC 1	VRSIH	3.5	-	VDI	V	
OUTPUTS						
SBY, Digital Out, C1, C2, C3, LRQ, ROM DIS, ROM CLK, SEROUT						
LOGIC 0	VOL	0.0	-	0.5	V	0.72ma (2LS TTL Loads)
LOGIC 1	VOH	2.5	-	VDI	V	-50ua (2LS TTL Loads)
OSCILLATOR						
OSC 2 Output						When driven from external source.
LOGIC 0	VOL	0.0	-	0.6	V	Pin 27 = INPUT = 3.90V MIN
LOGIC 1	VOH	2.5	-	VDI	V	Pin 27 = INPUT = 0.60V MAX