PHILIPS

Electronic components and materials

Technical publication

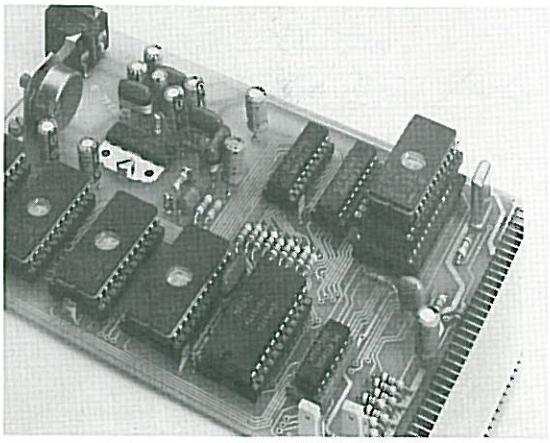
101

MEA8000 voice synthesizer: principles and interfacing

Until now, communication between machine and man has mainly been in sign language from visual displays, audible communication being restricted to cries of alarm emitted by beepers, hooters, bells and buzzers. Voice communication has been impractical because analogue storage of speech required the use of moving parts, e.g. tape drives, which unduly prolonged the retrieval time, and digital storage of speech required handling an enormous number of bits. The development of speech synthesis techniques (Fig.1) has dramatically reduced the bit rate and the memory required for digital speech synthesis so that it is now economically feasible to open a wideband voice channel between machine and man. Such a channel is provided by our totally-digital integrated voice synthesizer MEA8000.

FEATURES OF THE MEA8000

- interfaces easily with most 8-bit microprocessors and microcomputers
- 4 kHz speech bandwidth
- can generate melodies
- bit rate from 500 to 4000 bit/s thanks to a variable speech frame duration
- synthesis occupies a very small percentage of the control processor's time
- 8th-order digital filter with three programmable formant frequencies, one fixed formant frequency and four programmable formant bandwidths
- operates with standard PROMS
- requires minimal external audio filtering
- timing: crystal-controlled internal oscillator or external TTL clock
- dynamic NMOS technology
- 30 mA current consumption (typ.) from one +5 V supply
- 24-pin plastic DIL package.



Part of a demonstration board incorporating the MEA8000 voice synthesizer and an integrated audio output stage TDA1011. The MEA8000 uses the technique of vocal tract modelling to generate quality speech from digital code. Formant synthesis, a variation of linear predictive coding (LPC), is used to control the electronic model representing the human speech production mechanism. Formant synthesis has all the advantages of LPC plus a lower bit rate than straight LPC for the same speech quality.

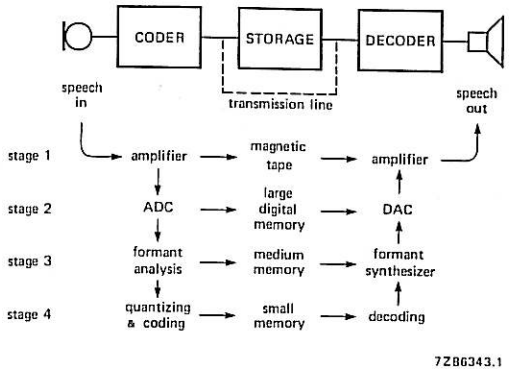


Fig.1 The development of speech storage methods. In stages 1 and 2 (waveform coding), the actual speech waveform is stored, the digital approach (stage 2) requiring a very large memory. In stages 3 and 4, a much smaller memory is required, because redundant speech information is eliminated and only the essential characteristics of the speech sounds are stored. This voice 'score' is then used to control a voice generating instrument (speech synthesizer). In stage 4, the method used in the MEA8000, formant coding allows further reduction of the required bit rate.

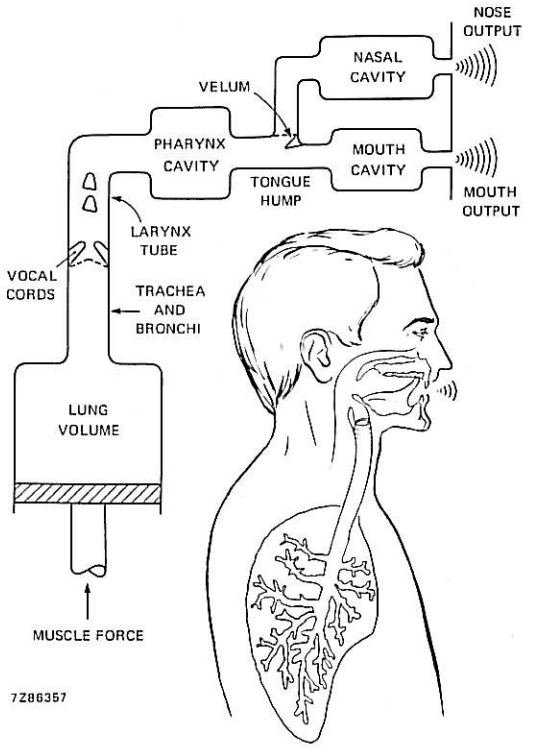


Fig.2 The human speech production mechanism.

PRINCIPLES OF FORMANT SPEECH SYNTHESIS

Figure 2 shows the human speech mechanism. To produce speech, the lungs build up air pressure like a pump. This increasing pressure causes the initially-closed vocal cords to open. As a result, pressure drops, the vocal cords close and pressure builds up again. This mechanism excites the vocal tract with a periodic train of sawtooth pressure pulses. Sounds generated in this way, e.g. the vowels A and E, are called voiced sounds. Voiced sounds contain a lot of harmonics which fall off at about 12 dB/octave. The frequency of this periodic signal is commonly referred to as pitch.

The vocal tract can also be excited with the vocal cords always slightly open, so that air passes continuously through them, causing turbulence in the vocal tract. Sounds generated in this way, for instance the sibilants, are called unvoiced sounds.

All speech is derived from either a periodic or a noise source, i.e. a voiced or an unvoiced source. During speech, the source and its amplitude are always varying, sometimes quite rapidly.

Situated above the vocal cords are the pharyngeal, oral and nasal cavities which shape the spectrum of the generated sounds. The nasal cavity is accessed via the velum. Like all other synthesizers, the MEA8000 does not simulate the velum and the nasal cavity, so their functions are not separately represented. Consequently, for speech synthesis, the vocal tract can be analysed as if it were a tube of constant diameter, Fig.3.

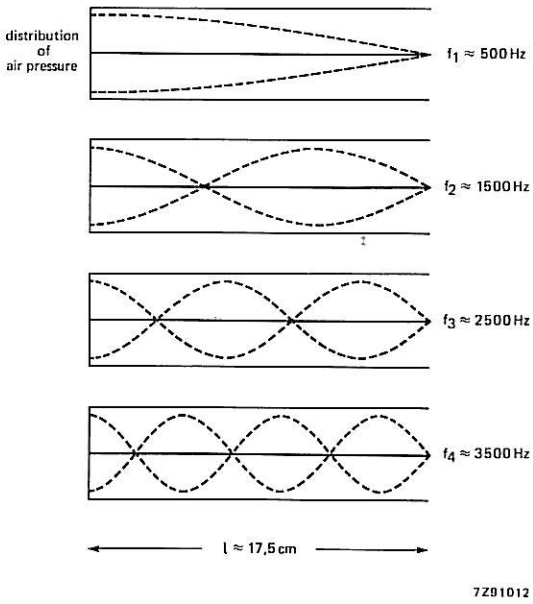


Fig.3 Resonance of a cylindrical tube.

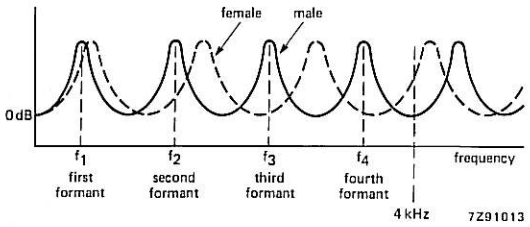


Fig.4 Frequency response of a cylindrical tube.

This tube is almost closed by the vocal cords at one end and is open at the other where mouth radiation takes place. The frequency response of the tube is characterized by a number of equally-spaced resonances at frequencies given by:

$$f(n) = \frac{340\,(2n-1)}{41} \qquad \text{for } n = 1, 2, 3, 4 \ldots$$

where 1 is the tube length in metres.

These resonant frequencies are called the formants of the vocal tract. Within a 4 kHz bandwidth, as used in the MEA8000, four formants are present for a male voice ($1 \approx 0{,}175$) and three for a female voice due to a woman's shorter vocal tract, see Figs.3 and 4.

During speech, the shape of the vocal tract is constantly changing. When an E is pronounced, for example, the pharyngeal cavity is large while the oral cavity is small. This increases the frequency of the second formant. When an A is pronounced, the situation is reversed, reducing the separation between the first and second formant, see Fig.5.

Each formant is further characterized by its bandwidth. The first two or three formants are the most important for intelligible speech; the MEA8000 generates and shapes four. The first three have adjustable frequency and adjustable bandwidth; the fourth has a fixed frequency of 3500 Hz and adjustable bandwidth.

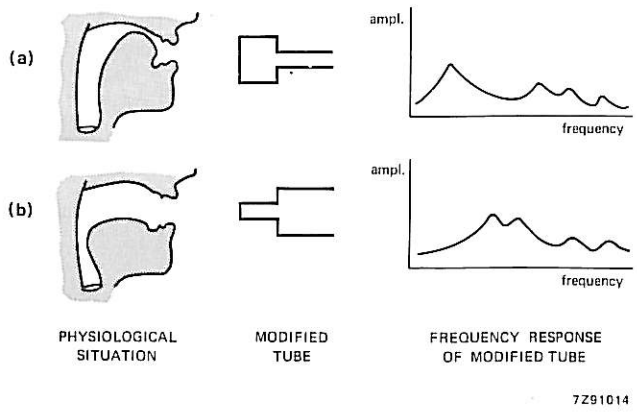


Fig.5 (a) Pronouncing E; (b) Pronouncing A.

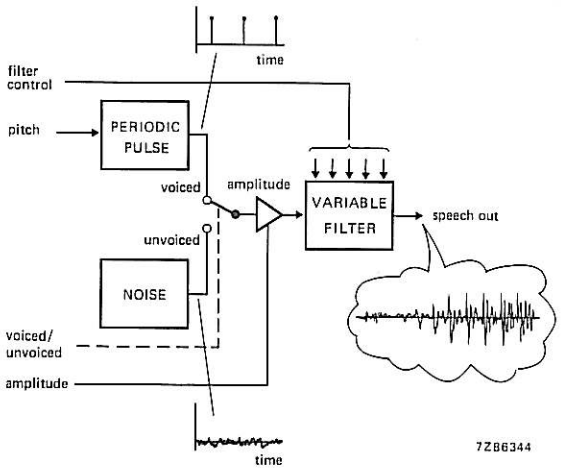


Fig.6 Simple electronic model of the human speech production mechanism.

Figure 6 shows a simplified electronic model of the human speech production mechanism (i.e. a formant synthesizer). A combination of a periodic signal, representing the pitch of the original speech, and an aperiodic signal, representing the unvoiced sound in the speech, is fed to a variable filter comprising four resonators, via an amplifier which controls the amplitude of the synthesized sound. The resonators model the sound in accordance with the formants in the original speech. Each resonator is controlled by two parameters, one for the resonant frequency and one for the bandwidth. The information required to control the synthesizer is:

– pitch – amplitude – voiced/unvoiced source selector	excitation source (vocal cords)
– filter settings	spectrum shaping (vocal tract)

A good replica of the original speech is obtained by periodic updating of this control information.

In the MEA8000, each formant is simulated by a second-order digital filter, comprising three multipliers, an adder and two delays (Fig.7). The resonant frequency and bandwidth are set by assigning different values to the multipliers A, B and C where:

$A = 1 - B - C$	for unity gain
$B = 2\sqrt{-C}\cos 2\pi f_o/f_s$	sets the formant frequency (f_o is the resonant frequency, f_s is the sampling frequency)
$C = -\exp(-2\pi b/f_s)$	sets the 3 dB bandwith b.

The filter of Fig.7 can be modified slightly to give explicit expressions for bandwidth and resonant frequency, see Fig.8. To simulate four formants, four such filters are cascaded, see Fig.9.

Figure 10 shows the complete formant synthesizer. As mentioned earlier, the jaws, tongue and lips are constantly moving during speech. In addition, the pitch and amplitude are constantly changing and the type of source can change too. This is why all input parameters controlling the synthesizer need regular updating. In the MEA8000, the standard updating period is 8 ms.

Speech parameters are determined using the LPC encoding algorithm on sampled speech data. This involves sampling the speech waveform and passing the samples through a digital analysing filter to obtain an 'inverse' voice spectrum. From this spectrum, it is possible to extract the centre frequencies and bandwidths of the four formants that represent the vocal tract resonances up to 4 kHz. From the formants, the coefficients of the digital filter in the MEA8000 which control the electronic vocal tract can be set to give faithful reproduction of the original recorded speech. The advantage of formant encoding over LPC encoding is a lower bit rate for equal speech quality.

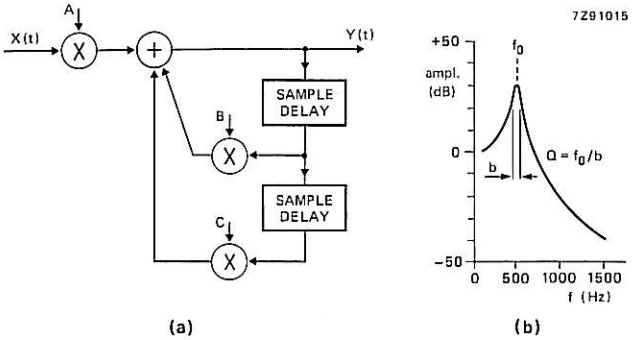


Fig.7 (a) Second-order digital filter and (b) its frequency response for: a formant centre frequency f_0 = 500 Hz (A = 0,15); a 3 dB bandwidth b = 100 Hz (B = 1,77); and a sampling frequency f_s = 8 kHz (C = −0,92).

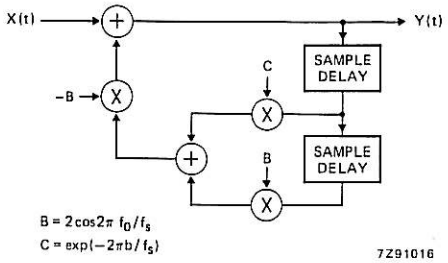


Fig.8 Modified digital filter.

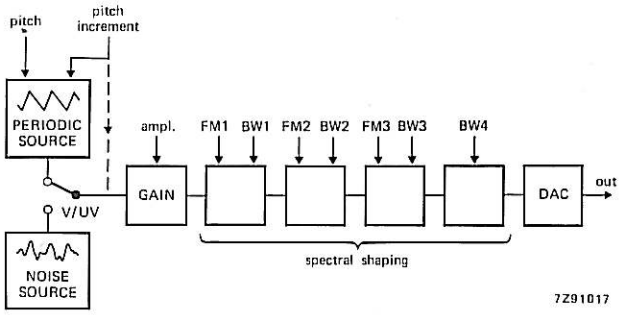


Fig.10 Formant synthesizer.

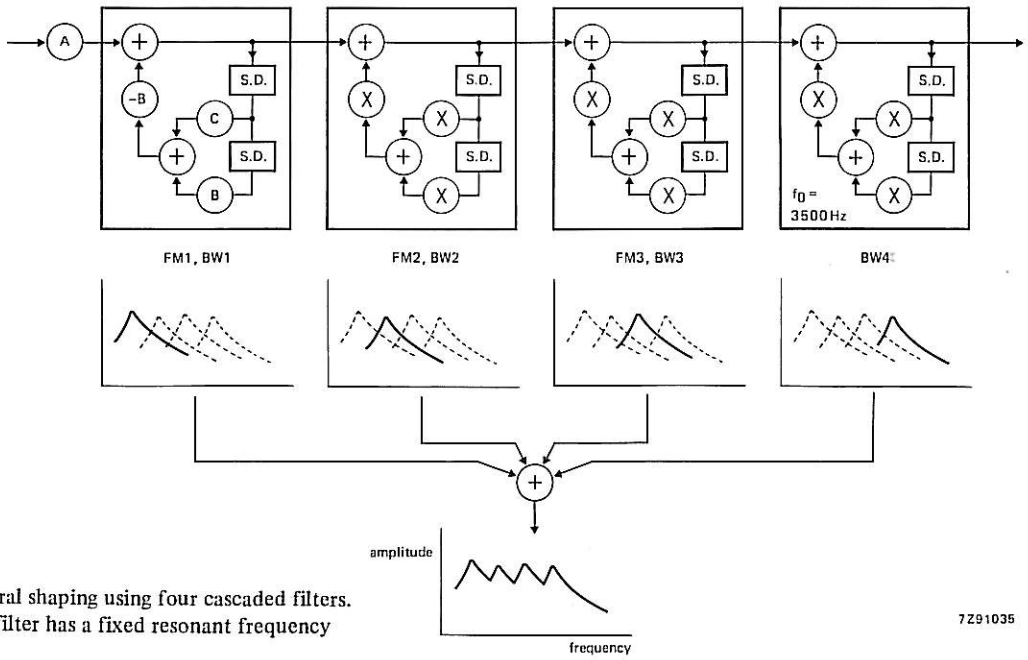


Fig.9 Spectral shaping using four cascaded filters. The fourth filter has a fixed resonant frequency of 3500 Hz.

MEA8000 BLOCK DIAGRAM

Figure 11 is a block diagram of the MEA8000. Speech codes from external ROM are sent to the synthesizer and converted to the parameters controlling the formant synthesizer. The output of the formant synthesizer is converted to an analogue signal, ready for amplification and filtering.

Formant synthesizer

The periodic and noise sources and the four formant resonators actually consist of a 16-bit multiplication and addition unit which calculates the voice samples at a rate of 8 kHz. The synthesizer is controlled by eleven parameters representing pitch, pitch increment (rate of pitch change) for voiced operation or noise selection for unvoiced operation, amplitude, four filter centre frequencies and four filter bandwidths.

A second-order digital filter is used to simulate each formant resonator. To simplify the ROM containing the filter coefficients, we have used three multipliers in the digital formant filters (see Fig.8) so that each bandwidth and each centre frequency are determined by only one filter coefficient.

Output circuit

The 16-bit samples from the formant filter bank are truncated to 11-bits before entering the interpolation and digital-to-analogue circuitry. This circuitry combines pulse-width and current modulation techniques to perform the dual functions of 8-bit DAC and linear interpolator, the latter generating seven additional samples between each 8 kHz sample from the formant filter bank. The sample rate of the DAC is therefore 64 kHz which is far above the audible frequency range, allowing the use of a simple external audio filter.

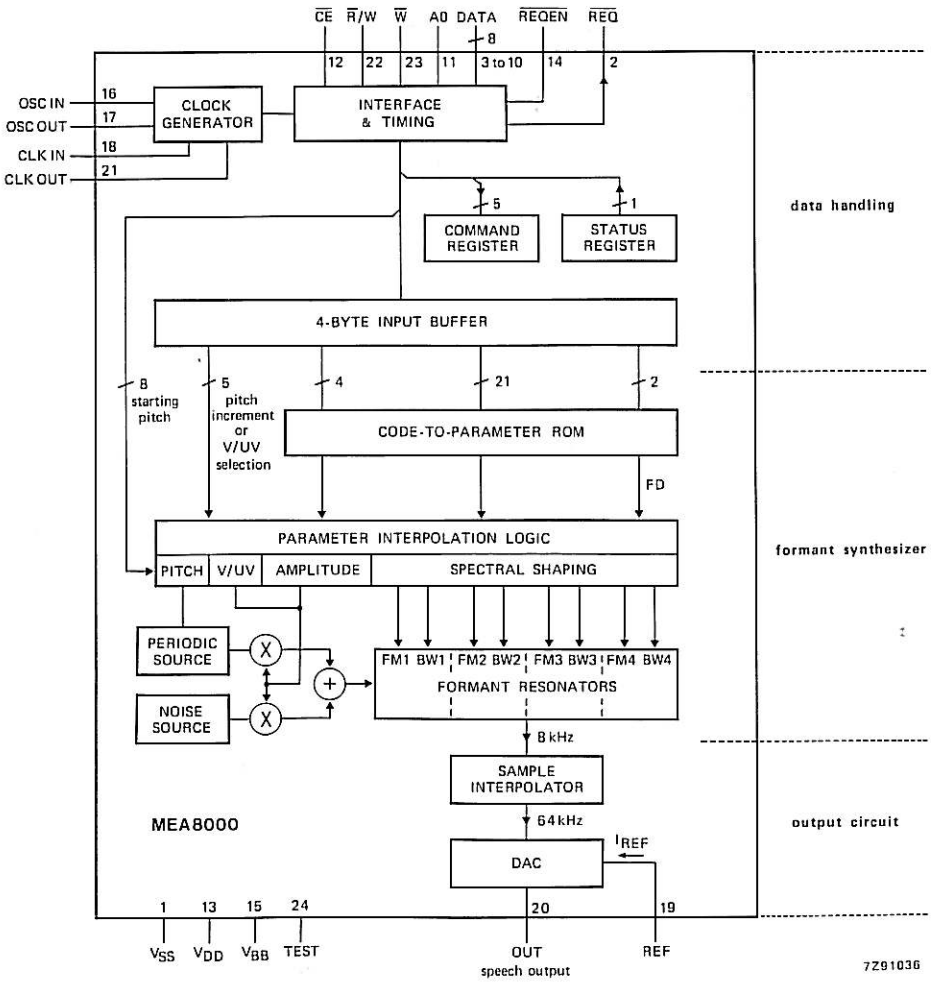


Fig.11 Block diagram of the MEA8000.

Input data handling, speech code format and parameter quantization

Since the human vocal tract is a mechanical system, its characteristics change quite slowly during the formation of voice sounds. It has been found that the speech synthesizer control parameters can be adequately represented if they are updated once every few tens of milliseconds with linear interpolation during the intervals to ensure a smooth changeover from one set of parameters to the next. In the MEA8000, the updating period (called a speech frame) can be set to 8, 16, 32 or 64 ms.

The duration of a speech frame must be long enough for it to contain sufficient speech samples to allow the speech parameters to be calculated, yet short enough to isolate changes of the parameters. Using a long speech frame when the utterance is either not changing, or changing linearly means that intelligible speech can be created using average bit rates of about 1000 bit/s.

During voice output, the speech codes from a microcomputer or external ROM are transmitted on an 8-bit data bus to the DATA port of the MEA8000 in blocks of four bytes, each block characterizing a speech frame, see Fig.12 and Table 1. Byte four contains a 5-bit pitch increment code which can be positive or negative. However, when the synthesizer starts to talk, a preliminary byte containing the full starting pitch code must be transmitted. This byte goes directly to the pitch generating circuitry via the input interface. This method of encoding pitch contributes to a lower bit rate.

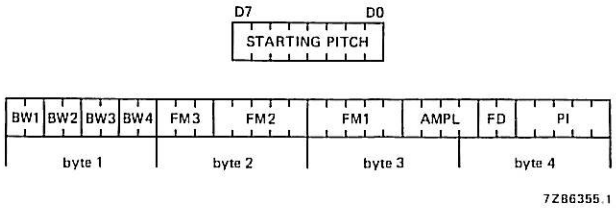


Fig.12 Format of a speech frame.

After the starting pitch code, the codes of each speech frame are shifted into a four-byte input buffer (when A0 = 0), before being translated into control parameters by the code-to-parameter ROM. The parameter interpolation logic calculates the difference between consecutive parameters and interpolates linearly between them to smooth the parameter transients. The interpolation interval is decoded using the two frame duration (FD) bits in each speech frame.

TABLE 1 Speech frame bit allocation.

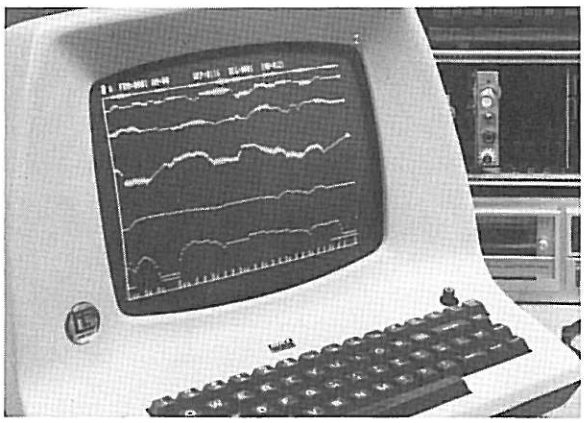
code	bits	parameter
PI	5	pitch increment or noise selection
FD	2	speech frame duration
AMPL	4	amplitude
FM1	5	frequency of 1st formant
FM2	5	frequency of 2nd formant
FM3	3	frequency of 3rd formant
BW1	2	bandwidth of 1st formant
BW2	2	bandwidth of 2nd formant
BW3	2	bandwidth of 3rd formant
BW4	2	bandwidth of 4th formant

FM4, the frequency of the fourth formant, is fixed.

MEA8000 EDITING SYSTEM

During speech encoding, digitized speech samples from a recorded voice are analysed by a computer to produce speech codes which, after manual editing, are stored in PROM. These codes are applied to the synthesizer which translates them into the pitch, amplitude, voiced/unvoiced source and filter control information required to reproduce the original speech.

In order to obtain the lowest possible bit rate and the highest possible speech quality, any voice synthesizer needs an editing system. Present editing systems for both waveform analysis and LPC synthesizers have the severe disadvantage that the speech to be edited is usually displayed on a screen in the form of complex tables of coded parameters. It is a specialized task to edit these parameters, because the editor must know what the codes mean before he can start. In our editing system, the synthesized speech is displayed on a screen as a waveform instead of as codes. Speech editing is so simplified by this method that the entire speech editing process be learnt in a day. At present, speech encoding and editing for the MEA8000 is a service that we provide. A stand-alone speech editing system will be included in a popular type of personal computer available in autumn 1983.



The pitch increment code goes directly from the buffer to the parameter interpolation logic. One of the pitch increment codes, 10000_2 (16_{10}), does not change the pitch, and is used to select the voiced or unvoiced source.

The outputs of the interpolation logic control the aforementioned synthesizer functions of pitch, amplitude, voiced/unvoiced source selection and filter formants.

The speech code parameters are shown in Table 2. Fig.13 shows the beginning of a stream of speech codes for the utterance 's'. The starting pitch is 98 Hz (represented by H31) and 104 ms of speech are represented by these codes, since the FD bits indicate speech frames of 32, 64 and 8 ms. The average bit rate of this utterance would therefore be about 1000 bit/s. Note, a common speech frame of 8 ms is used initially, this being changed if necessary during editing.

TABLE 2 Speech code parameters when using a clock frequency of 3,84 MHz.

decimal code	hex. code	FD (ms)	pitch (Hz)	PI (Hz/8 ms)	amplitude	FM1 (Hz)	FM2 (Hz)	FM3 (Hz)	BW (Hz)
0	00	8	0	0	0	150	440	1179	726
1	01	16	2	1	0,008	162	466	1337	309
2	02	32	4	2	0,011	174	494	1528	125
3	03	64	6	3	0,016	188	523	1761	50
4	04		8	4	0,022	202	554	2047	
5	05		10	5	0,031	217	587	2400	
6	06		12	6	0,044	233	622	2842	
7	07		14	7	0,062	250	659	3400	
8	08		16	8	0,088	267	698		
9	09		18	9	0,125	286	740		
10	0A		20	10	0,177	305	784		
11	0B		22	11	0,250	325	830		
12	0C		24	12	0,354	346	880		
13	0D		26	13	0,500	368	932		
14	0E		28	14	0,707	391	988		
15	0F		30	15	1,00	415	1047		
16	10		32	noise		440	1110		
17	11		34	−15		466	1179		
18	12		36	−14		494	1254		
19	13		38	−13		523	1337		
20	14		40	−12		554	1428		
21	15		42	−11		587	1528		
22	16		44	−10		622	1639		
23	17		46	−9		659	1761		
24	18		48	−8		698	1897		
25	19		50	−7		740	2047		
26	1A		52	−6		784	2214		
27	1B		54	−5		830	2400		
28	1C		56	−4		880	2609		
29	1D		58	−3		932	2842		
30	1E		60	−2		988	3105		
31	1F		62	−1		1047	3400		
⋮	⋮		⋮						
49	31		98						
⋮	⋮		⋮						
255	FF		510						

The frequency of FM4 is fixed at 3500 Hz. The BW (bandwidth) column applies to all four filters. For exact values of pitch and pitch increment, multiply the values given above by 1,024.

starting pitch	byte 1	byte 2	byte 3	byte 4	speech frame
31	05	D2	FE	50	1
	0A	D7	FE	70	2
	1A	D8	F5	90	3

(a)

byte 1				byte 2		byte 3		byte 4			speech frame
BW1	BW2	BW3	BW4	FM3	FM2	FM1		AMPL.	FD	PI	
00	00	01	01	110	10010	11111	110	0	10	10000	1
00	00	10	10	110	10111	11111	110	0	11	10000	2
00	01	10	10	110	11000	11110	101	1	00	10000	3

(b)

BW1	BW2	BW3	BW4	FM3	FM2	FM1	AMPL.	FD	PI	speech frame
726	726	309	309	2842	1254	1047	0,354	32	noise	1
726	726	125	125	2842	1761	1047	0,354	64	noise	2
726	309	125	125	2842	1897	988	0,250	8	noise	3

(c)

Fig.13 (a) Beginning of a stream of speech codes in hexadecimal notation; (b) binary representation of the codes shown in (a); (c) parameter values of the speech codes in (a) and (b).

CONTROL

Control inputs

The inputs $\overline{CE}$, $\overline{W}$, $\overline{R}/W$ and A0, together with the $\overline{REQ}$ output pin which signals a request for speech codes, are used to control the transmission of codes to the synthesizer and to set the synthesizer's operating mode.

The functions of the control inputs are:

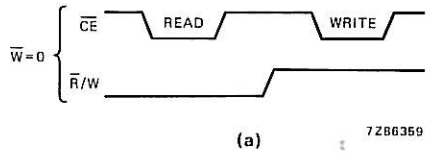
$\overline{CE}$ enables the circuit

$\overline{W}$ controls the writing of data

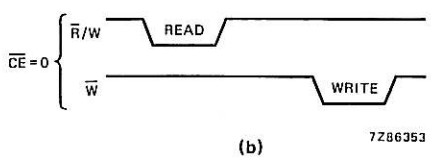
$\overline{R}/W$ controls the reading/writing of data

A0 when A0 = 0, the input buffer is addressed,
when A0 = 1, the command register is addressed.

Table 3 is the control input truth table. The control inputs can be used in many combinations to allow simple interfacing of the MEA8000 to a variety of host processors. Figure 14 shows two ways of interfacing the MEA8000 to most popular microcomputers. Read and write timing are shown in Figs.15 and 16 and Table 4.



(a) 7Z86359



(b) 7Z86353

Fig.14 (a) Chip enable ($\overline{CE}$) used as a read or write strobe; (b) separate read and write strobes.

TABLE 3 Control input truth table.

$\overline{CE}$	$\overline{W}$	R/W	A0	operation
0	0	1	0	write data
0	0	1	1	write command
0	X	0	X	read status
0	1	1	X	three-state data bus
1	X	X	X	

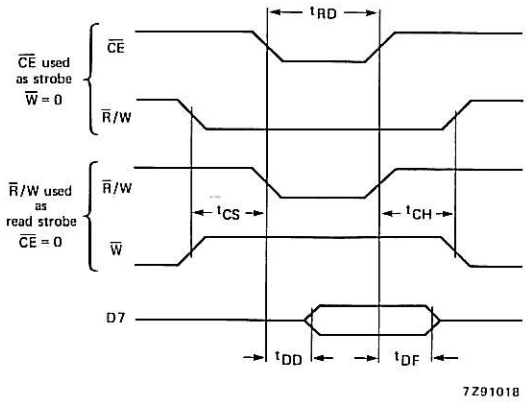


Fig.15 Read timing.

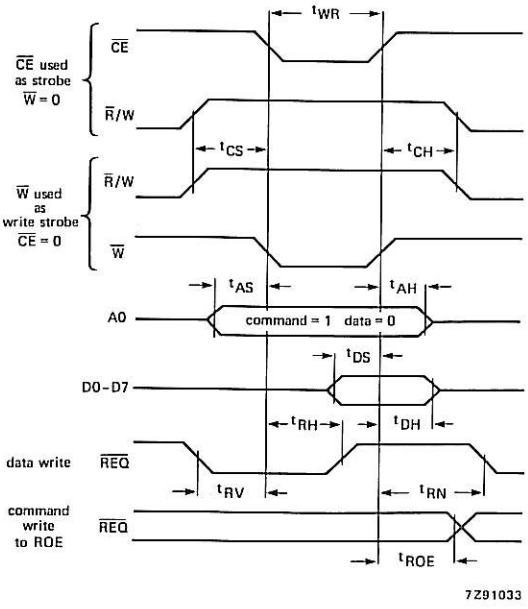


Fig.16 Write timing.

TABLE 4 Timing characteristics[1] (Figs.15 and 16).

		min.	*max.*	
write enable	t_{WR}	200	–	ns
address set-up	t_{AS}	30	–	ns
address hold	t_{AH}	30	–	ns
data set-up for write	t_{DS}	150	–	ns
data hold for write	t_{DH}	30	–	ns
request hold[2])	t_{RH}	–	350	ns
request next[3]) (3,84 MHz clock)	t_{RN}	–	3	µs
read enable	t_{RD}	200	–	ns
data delay for read[4])	t_{DD}	–	150	ns
data floating for read[4])	t_{DF}	–	150	ns
request valid before write	t_{RV}	0	–	ns
request output enable response	t_{ROE}	–	350	ns
control set-up	t_{CS}	–	20	ns
control hold	t_{CH}	–	20	ns

[1]) Timing reference level is 1,5 V.

[2]) $\overline{REQ}$ is an open drain output, requiring an external pull-up. The time stated is that to reach 2,0 V via a 3,3 kΩ and 50 pF pull-up connected to 5 V.

[3]) Between two data write operations of a speech frame.

[4]) Levels greater than 2,0 V for a 1, or less than 0,8 V for a 0 are reached with a load of one TTL input and 50 pF.

Command register and status register

The contents of the command register (command word) determines whether the synthesizer is silent or active (operating mode) and the procedure that will be followed when the supply of speech codes is intermittent. It also determines how the contents of the status register, i.e. the data request bit (REQ) used to request more data, will be broadcast. Table 5 gives the bit allocation of the command register. It is written into via data lines D4 to D0 when $\overline{CE} = \overline{W} = 0$ and A0 = R/W = 1.

The request bit REQ

REQ is stored in the status register and signals a request for the next byte of speech code, or the starting pitch byte if a STOP command has just been received. The request for data can be broadcast in two ways:

– on the $\overline{REQ}$ pin,

– on data port D7.

The $\overline{REQ}$ pin can be enabled in hardware or software. The hardware method is to connect $\overline{REQEN}$ (pin 14) to ground. The software method is to set ROE in the command register to a 1 (and to hold $\overline{REQEN}$ high). The $\overline{REQ}$ pin can be connected to the control processor's interrupt input or polled.

The status bit REQ can also be broadcast to the control processor via bidirectional port D7, D7 being read when $\overline{CE} = R/\overline{W} = 0$ (see Table 3).

TABLE 5 Command word bit allocation and truth table.

D4 STOP	D3 CONT enable	D2 CONT	D1 ROE enable	D0 ROE
0 = no action	0	0 = no action	0	0 = no action
1 = STOP	0	1 = no action	0	1 = no action
	1	0 = SLOW STOP procedure	1	0 = disable $\overline{\text{REQ}}$
	1	1 = CONTINUOUS procedure	1	1 = enable $\overline{\text{REQ}}$

D7, D6 and D5 are not used;
ROE = request output enable.

After power-on reset, the command register bits CONT and ROE will both be zero since power-on corresponds to the command word XXX11010. Therefore, the synthesizer will be in the SLOW STOP procedure.

OPERATING MODES AND PROCEDURE DURING INTERMITTENT CODE SUPPLY

The operating modes (Fig.17) are:

– the SILENT mode

– the ACTIVE mode.

The SILENT mode, characterized by a silent output and the status REQ bit high, is entered after a power-on reset or reception of a STOP command, or at the end of a SLOW STOP procedure. The STOP command mutes the synthesizer immediately. The active mode is re-entered after a starting pitch byte has been received.

In the ACTIVE mode, speech codes are synthesized. During the synthesis of one speech frame, all four bytes of the next frame must be received. If this is not the case, one of two procedures is followed depending on the CONT bit of the command word:

– the CONTINUOUS procedure (CONT = 1)

– the SLOW STOP procedure (CONT = 0).

The CONTINUOUS procedure causes the synthesizer to repeat the last speech frame indefinitely until all the codes of the next frame have been received or until a STOP command is received. When generating melodies, this procedure can be used to advantage.

The SLOW STOP procedure causes the synthesizer to enter the SILENT mode by repeating the last frame once, decreasing amplitude to zero and then going silent.

Figure 18 shows the audio output in the case of intermittent code supply for the CONTINUOUS and SLOW STOP procedure.

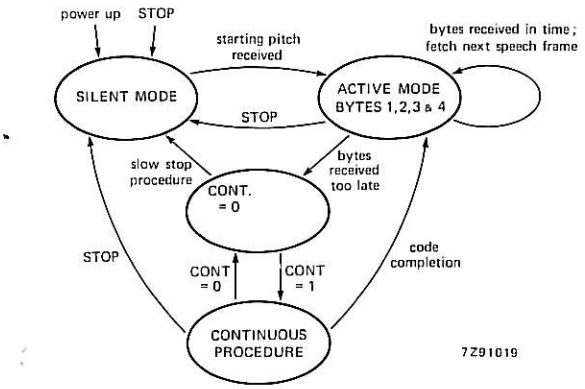


Fig.17 Operating modes and procedures.

Figure 19 shows the speech codes for the utterance 's' again, plus the audio output during the utterance. The timing for the first frame is shown in Fig.20. Subsequent frames have identical timing, with the exception of the starting pitch. The utterance starts from the SILENT mode, in this case after a STOP command has been written to the synthesizer, so $\overline{\text{REQ}}$ is low. After the starting pitch byte has been written, $\overline{\text{REQ}}$ remains high for up to 8 ms, then goes low indicating that the first byte of speech codes may be sent. To write data to the input buffer, $\overline{\text{CE}}$ is first brought low, data being written on the rising edge of $\overline{\text{CE}}$. After a byte has been received, $\overline{\text{REQ}}$ remains high for about 2 μs between the first and second, second and third, and third and fourth bytes. After the fourth byte has been written, $\overline{\text{REQ}}$ remains high for 8 to 64 ms during which time the synthesizer sets the speech parameters of the first frame according to the codes 05, D2, FE and 50.

Preparing the first frame is different to preparing those that follow, because, except for the amplitude which starts from zero and reaches its correct value at the end of the frame, the value of each parameter is set before the first frame is spoken. For subsequent frames, the values of all parameters are only reached at the end of the frame owing to the internal linear parameter interpolation.

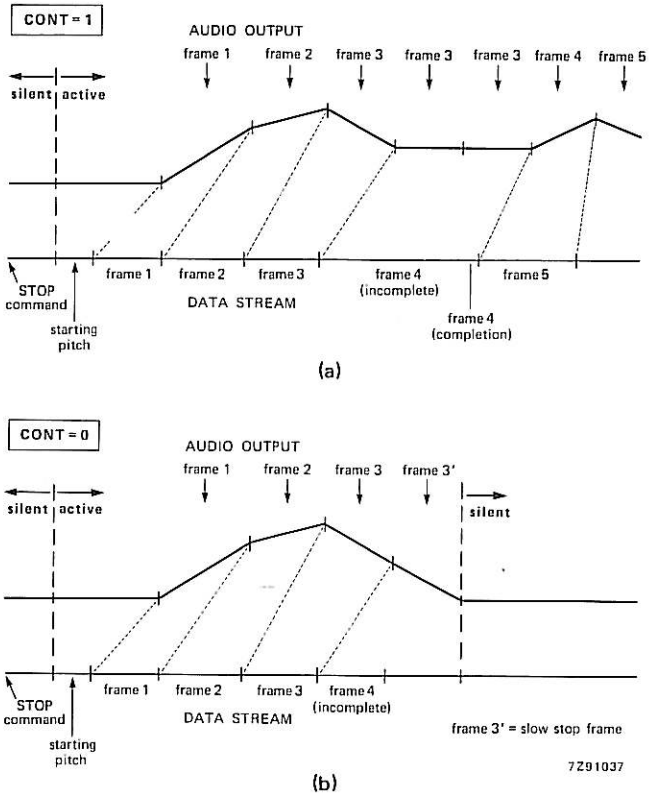


Fig.18 Audio output in the case of intermittent code supply
(a) for the CONTINUOUS procedure
(b) for the SLOW STOP procedure.

At t_2 (Fig.19), the synthesizer starts speaking, $\overline{REQ}$ going low at the same time, requesting the next speech codes. The duration of the first speech frame is 32 ms (the FD bits being 10_2) during which the four bytes of the second frame must be received.

At t_3, pronounciation of the first frame ends, the synthesizer starts pronouncing the second and, by means of $\overline{REQ}$, requests the codes of the third. In this example, 64 ms are available to receive the third frame, since that is the time taken to pronounce the second.

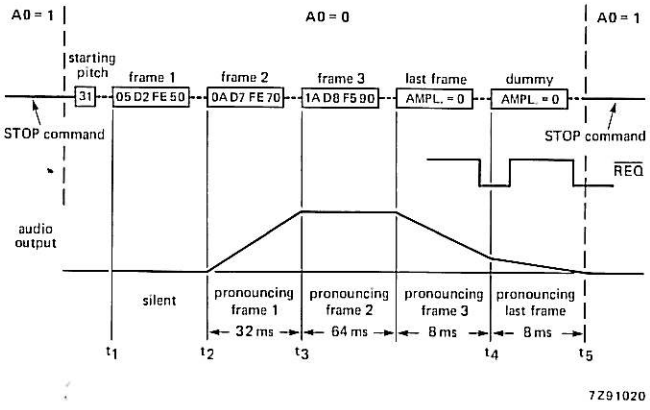


Fig.19 A four frame utterance in which a dummy frame is used to ensure that pronounciation of the last frame is not curtailed by the STOP command.

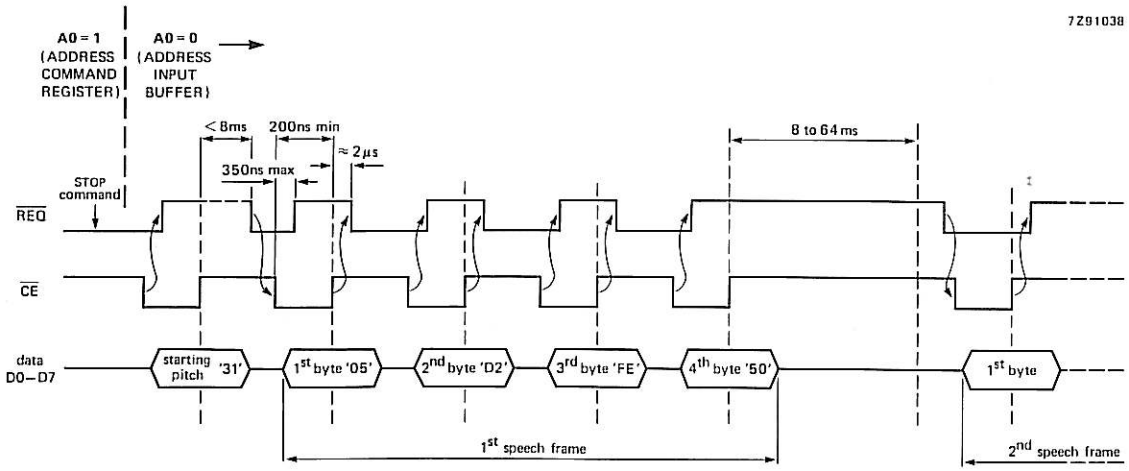


Fig.20 Timing of the first frame of Fig.19 in detail. Data is written on the rising edge of $\overline{CE}$. Polling of $\overline{REQ}$ is unnecessary since it returns to the high state within 3 µs of receiving each byte. The trailing edge of $\overline{REQ}$ is good timing reference, occurring at multiples of 8 ms.

At t_4, the synthesizer starts pronouncing the last frame. Since the utterance has to be terminated by a STOP command so that the new starting pitch of the next utterance can be received, the microprocessor has to determine when STOP may be written without shortening the last frame. A practical way of doing this is to provide a dummy frame when $\overline{REQ}$ goes low at t_4, the amplitude bits of the dummy frame being set to zero. After the dummy frame has been received, $\overline{REQ}$ goes low again at t_5 marking the end of speech output and STOP may be written.

INTERPOLATION, D/A CONVERSION AND SPECTRAL RESPONSE

The 8 kHz samples from the digital filters are fed to a linear interpolator which increases the effective sampling rate to 64 kHz, simplifying analogue post filtering.

The 8-bit digital samples are converted into a series of synthesized speech waveform increments by the D/A converter shown in Fig.21. The converter consists of two parallel-connected open-drain current sinks of amplitude I and 16I and an external capacitor. Current I equals the d.c. current injected into the REF pin of the MEA8000. When all eight bits of the digital sample are zero, both current sinks are off and capacitor C charges through resistor R. For other digital samples, one or both sinks are activated and three capacitor discharge currents can be defined (I, 16I and 17I). To ensure that the charge on the capacitor is well-defined before each digital sample is converted, both sinks are turned off for the last two clocks of each 20-clock sampling period and on the first three, leaving 15 clocks per sampling period to define each analogue increment.

The current sinks are switched on for the number of clocks indicated by the two nibbles of each 8-bit sample. The least significant nibble indicates the time for which the capacitor discharges into the I current sink, and the most significant nibble that for which it discharges into the 16I sink, see Fig.22. Each nibble can determine discharge durations from 0 clocks (both sinks off for whole sampling period) to 15 clocks. The sixteen possible discharge durations of the 16I sink combined with the sixteen of the I sink allow 256 average voltage levels (16 x 16) to be defined for each increment of the speech waveform.

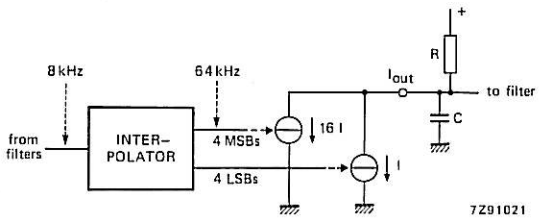


Fig.21 Output circuit.

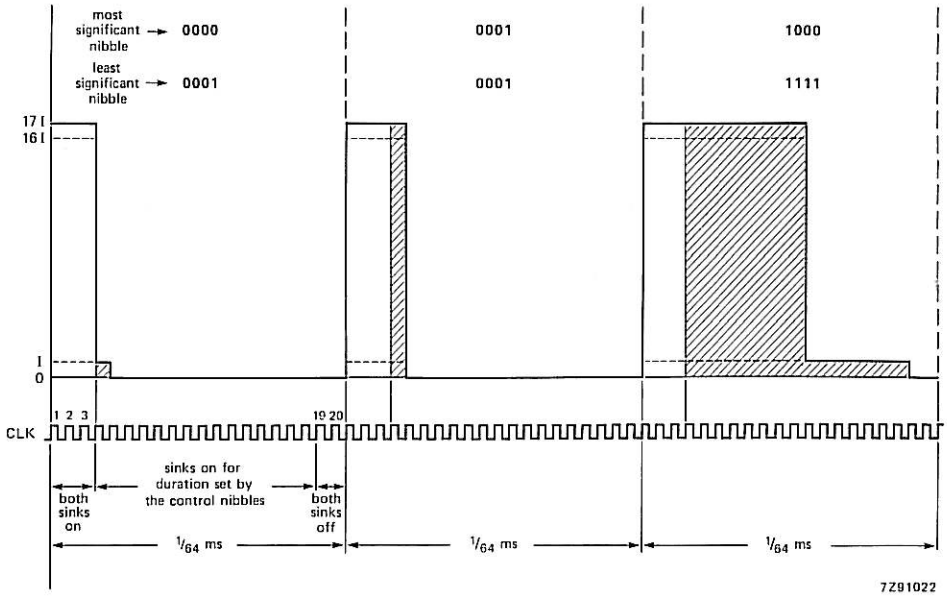


Fig.22 Output current pulses from the MEA8000 for three samples. Two 4-bit codes are used to produce a possible 256 average charge levels of which three are shown here shaded.

Note, to compensate for the fall-off in the frequency response due to the linear interpolation, an analogue post filter for the MEA8000 should have an $(x/\sin x)^2$ correction, Fig.23. The interpolation can be regarded as a convolution of the 8 kHz samples with a triangular non-causal impulse response, see Fig.24.

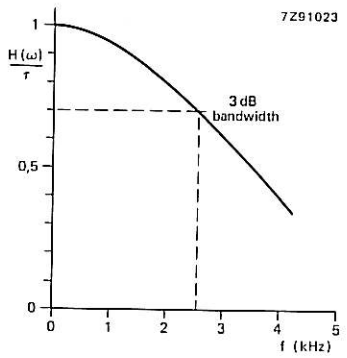


Fig.23 $H(\omega)/\tau$. Frequency response of the MEA8000. $1/\tau$ is the sampling frequency.

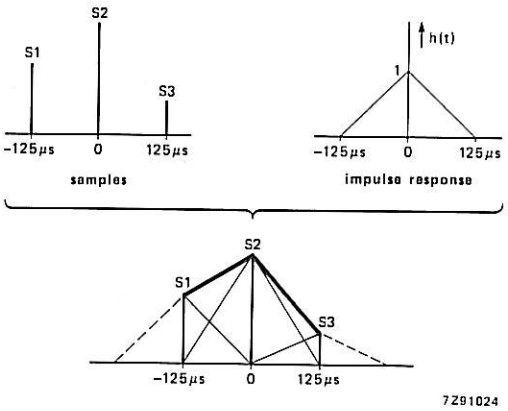


Fig.24 Convolution of 8 kHz samples with a triangular non-causal impulse response.

AUDIO OUTPUT STAGE

Figure 25 shows an integrated 6 W audio output stage for the MEA8000 and filter which performs $(x/\sin x)^2$ correction as well as bandlimiting. The best voice quality is obtained with an audio filter having the transfer characteristic shown in Fig.26 and Table 6. The characteristic of the actual filter is a good approximation of the optimum.

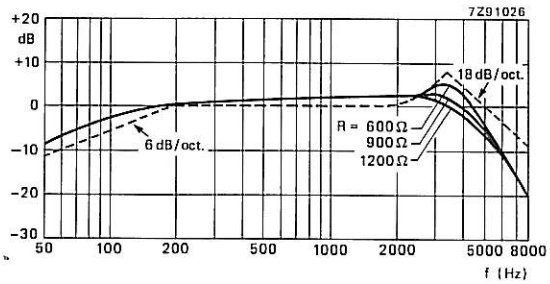


Fig.26 Output filter transfer characteristic.
--- for optimum voice quality;
—— characteristic of the filter shown in Fig.25.

The first section of the filter integrates the pulses from the output of the MEA8000 and cuts off at about 3400 Hz. The second section is an LCR circuit with resonant frequency 3400 Hz which compensates for the $(x/\sin x)^2$ distortion. If less high frequency compensation is required, the Q of the resonant circuit ($\omega L/R$) can be decreased by connecting a resistor in series with the inductor. The 600 Ω resistance R is the winding resistance of the 100 mH inductor. The third section is a first order low-pass section which removes any d.c. components.

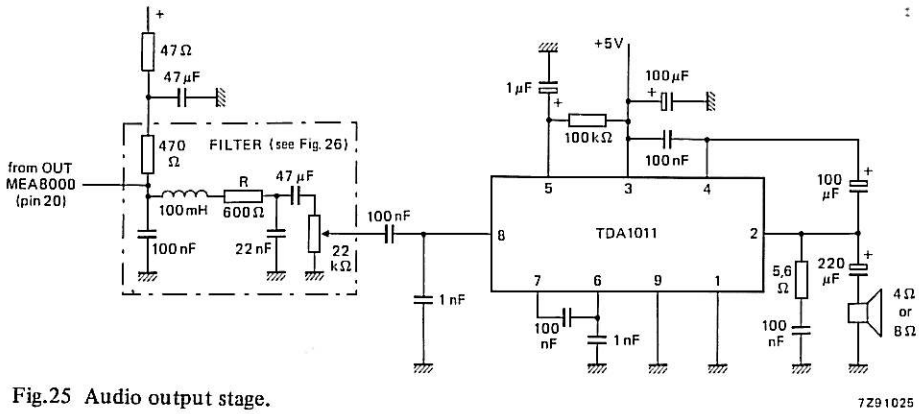


Fig.25 Audio output stage.

TABLE 6 $(x/\sin x)^2$ correction factors for the audio output filter.

freq. (Hz)	$\left(\frac{\sin x}{x}\right)^2$	$20 \log \left(\frac{x}{\sin x}\right)^2$ (dB)
0	1	0
400	0,99	0,09
800	0,97	0,26
1200	0,93	0,63
1600	0,88	1,11
2000	0,81	1,83
2400	0,74	2,62
2800	0,66	3,61
3200	0,57	4,88
3600	0,49	6,20
4000	0,41	7,74

Table 7 shows a header and a starting pitch byte followed by twenty-one four-byte speech frames for synthesizing the word 'stop'.

Usually, more than one speech file will be stored in a ROM. An index is made by listening the 2-byte starting addresses of each file at the beginning of the ROM. The end of the index is indicated by the bytes FF FF. Figure 27 shows examples of ROM mapping.

Speech output routine

This routine controls the transmission of speech codes to the synthesizer. Figure 28 shows the flow chart.

Each utterance is terminated with the STOP command.

SOFTWARE

ROM mapping

The external ROM that stores the speech codes of an utterance or a word (called a speech file) also stores the starting pitch byte and the file header. The header comprises three bytes, two that indicate the number of bytes in the file and one that allows additional data to be encoded for each file.

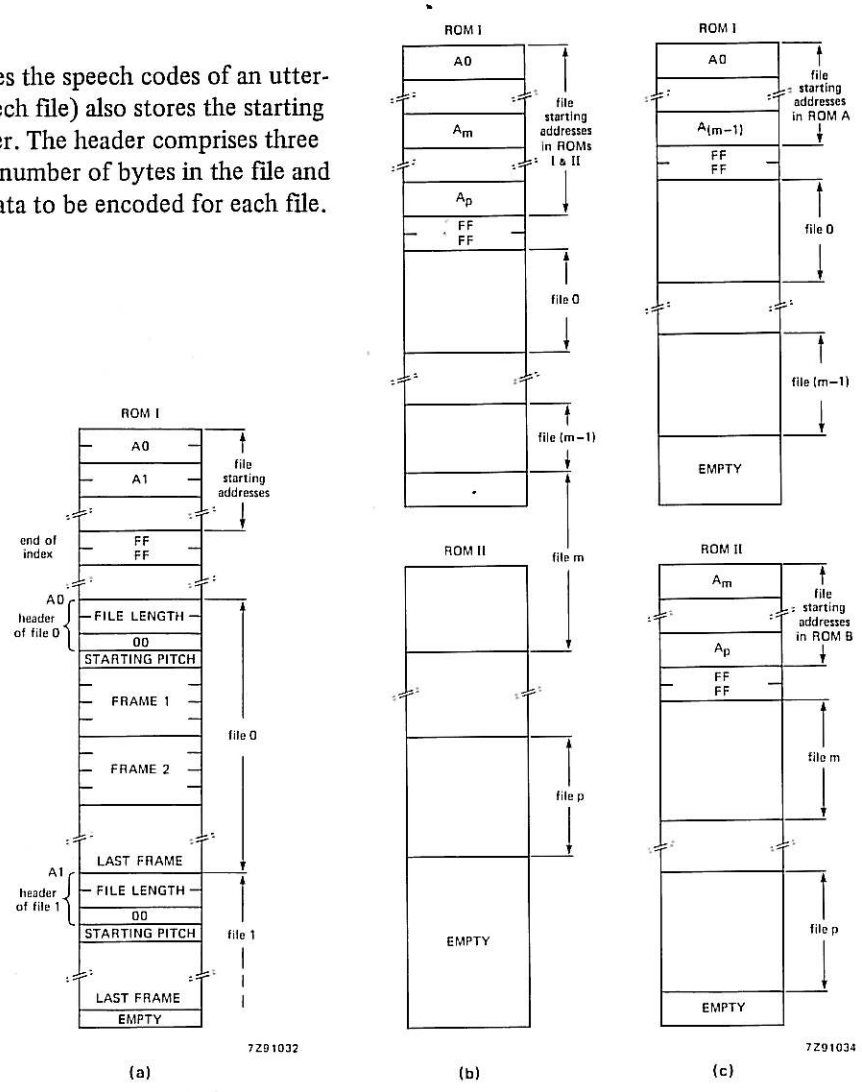


Fig.27 Examples of ROM memory mapping; (a) vocabulary in one ROM, (b) and (c) vocabulary in two ROMs.

TABLE 7 Hexadecimal speech codes for the word 'stop'.

												(a)			(b)	
												00	58	00	31	
05	D2	FE	50	0A	D7	FE	70	1A	D8	F5	90	1A	D8	F0	10	Speech codes
44	D9	F8	70	48	DA	FF	B0	08	DB	FF	90	46	DB	FF	02	
79	D1	67	0F	AA	CF	9F	7E	BB	AD	96	C1	96	CC	85	BE	
46	8F	B4	99	55	D7	24	12	45	B6	13	9E	05	B5	A3	1F	
05	B5	A0	00	2A	B3	B0	70	59	B3	E5	B0	2A	B2	DD	B0	
19	B2	ED	90													

(a) header containing byte count of word code file for microcomputer.
(b) starting pitch.

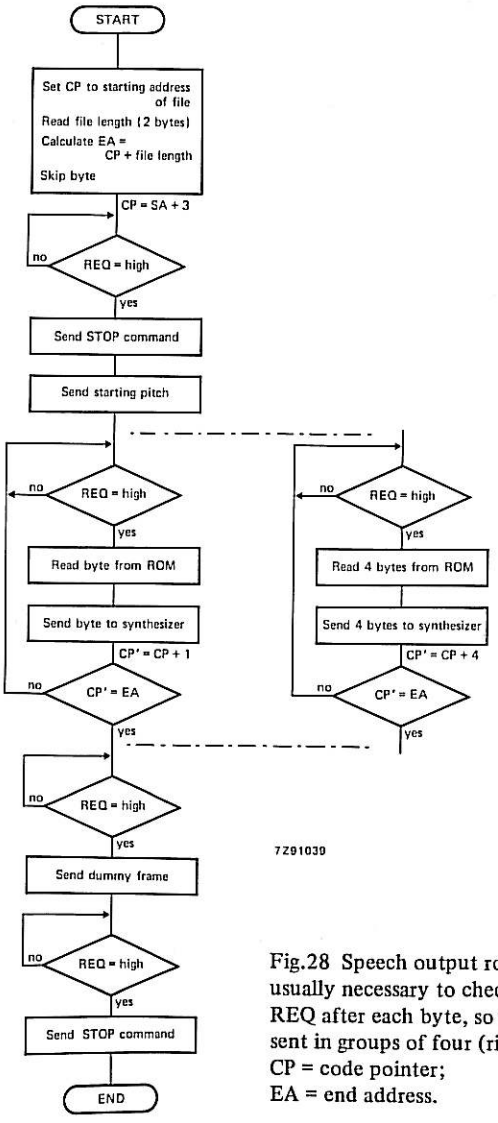


Fig.28 Speech output routine. It is not usually necessary to check the status of REQ after each byte, so bytes can be sent in groups of four (right).
CP = code pointer;
EA = end address.

A STOP command should be given when the status bit REQ is high. It can also be useful to send a STOP command at the beginning of an utterance, e.g. if a new frame is to override the present one.

INTERFACING

System timing is set using clock pulses from an internal oscillator controlled by a crystal connected between OSC-IN and OSC-OUT. The synthesizer can also be driven by an external clock via CLK IN. Pulses at one third of the clock frequency are available at CLK OUT. Figure 29 shows examples of oscillator/clock configurations.

Figure 30 shows how the MEA8000 can be interfaced to different control devices.

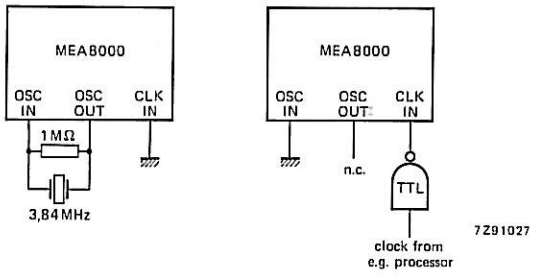
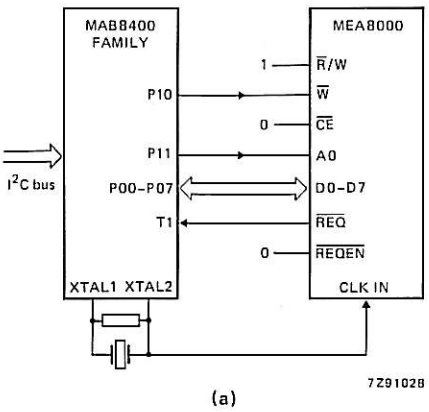


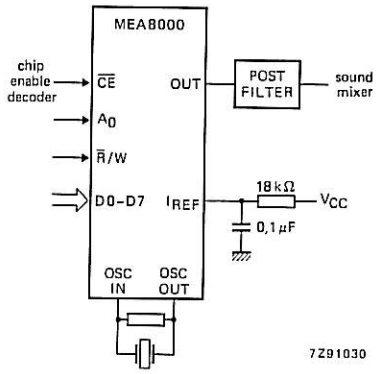
Fig.29 Oscillator/clock configurations.

REFERENCE

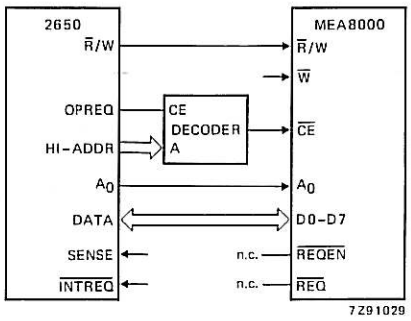
1. 'Higher quality synthesised speech through fast and easy editing', Electronic Components and Applications, Vol.4 No.4, Aug. 1982, p. 241.



(a)



(c)



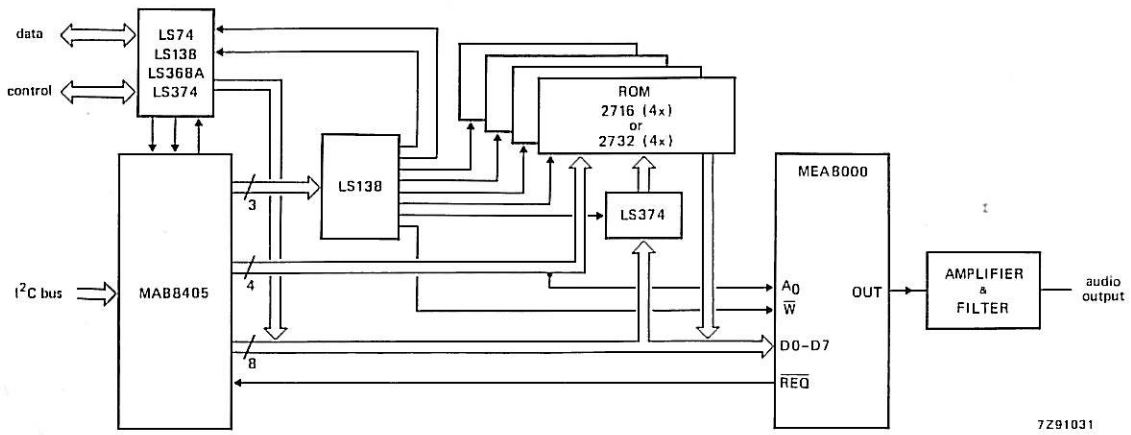
(b)

Fig.30 (a) Small vocabulary system. Status is tested on the T1 input via the $\overline{REQ}$ signal. $\overline{REQ}$ is enabled via $\overline{REQEN}$ = 0. The MAB8400 microcomputer holds both the program and the speech codes.

(b) Small vocabulary system. Status is tested via data port D7. R/W pins are connected so that the REQ signal can be read on D7. The program and the speech codes are in external ROM.

(c) Adding a voice output to an existing video game. In addition to the hardware shown, the game cartridge ROM needs to be increased by about 125 bytes for the speech output routine. The speech codes can also be put in the same ROM. When used with an interrupt routine, only 1% of the processor's time is used.

(d) General system for applications in card systems, small-volume large-vocabularly systems. Serial or parallel input data. Status is read via $\overline{REQ}$.



(d)

© 1983 Philips Export B.V.

This information is furnished for guidance, and with no guarantee as to its accuracy or completeness; its pulication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

Printed in The Netherlands Date of release: 13 May 1983 X1, X2, X3, V5, V7 I.K. 9398 041 80011