MICROPROCESSOR CONTROLLED PROGRAMMABLE AUDIO SYSTEM

BY

ABDULLAH FARUQUE

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ENGINEERING (ELECTRICAL & ELECTRONIC)

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA, BANGLADESH.

CERTIFICATE

This is to certify that this work has been done by me and it has not been submitted elsewhere for the award of any degree or dimploma or for publication.

Countersigned by the Supervisor	Signature of the Candidate
....	....
(DR. SYED MAHBUBUR RAHMAN)	(ABDULLAH FARUQUE)

Accepted as satisfactory for partial fulfilment of the requirements for the degree of M. Sc. Engineering in Electrical and Electronic Engineering.

EXAMINERS

1. ....
(Dr. Syed Mahbubur Rahman)
Supervisor

2. ...... 11/8/84
(Dr. Shamsuddin Ahmed)
Member

3. ....
(Dr. A.K.M. Mahfuzur Rahman Khan)
Member

4. ....
(Dr. A. R. Khan)
External Examiner

DHAKA
AUGUST, 1984

ACKNOWLEDGEMENT

The author expresses his indebtedness and deep sense of gratitude to Dr. Syed Mahbubur Rahman, Assistant Professor of Electrical and Electronic Engineering Department, Bangladesh University of Engineering and Technology, Dhaka, for his continuous guidance, valuable suggestions, constant encouragement and help all along the course of this work.

The author wishes to express his sincere gratitude and great indebtedness to Prof. A. M. Patwari, Vice-Chancellor, Bangladesh University of Engineering and Technology, for his constant advice and encouragement during this work.

The author expresses his greatfulness and thanks to Prof. Shamsuddin Ahmed, Head of the Department of Electrical and Electronic Engineering, who was kind enough to provide all the facilities of the Department for this work.

The author wishes to give whole hearted thanks to Mr. Mukhtar Ahmed, Assistant architect, Engineering and Planning Consultants Limited, Dhaka, Bangladesh for his helpful assistance in printing this thesis.

The author wishes to thank the staff of Electronics Laboratory of the Department of Electrical and Electronic Engineering, for their co-operation and help during this work.

TKS.

ABSTRACT

Recently microprocessors are used in a wide range of applications, such as Process Control Systems, Communication Systems, Digital Instruments and Consumer Products.This work deals with the application of microprocessors in audio system. A microprocessors controlled programmable audio system has been develöpd, which is able to generate and compose the tones of different musical instruments. This thesis describes the complete hard-ware and soft-ware design of a microprocessor controlled musical instrument. A conventional musical key-board has been interfaced with a microprocessor for real time operations. A soft-ware program has been developed to store or record a song in the memory (RAM) of the system for auto-play operation. This work is intended for musicians and composers of contemporary music who are interested in the application of microprocessor technology to the arts.

CONTENTS

CHAPTER - I

INDRODUCTION

Can a microprocessor be considered as a musical instrumant ? A microprocessor can be programmed to create musical tones of unlimited variety and versatility. It is possible for a microprocessor to produce different musical sounds, if it is properly programmed and interfaced with a key-board,loudspeaker and other audio electronic devices.This work describes a microprocessor controlled system design for generating tones of different musical instruments. As this work has emphasized, a microprocessor may be employed as a tremendously versatile and flexible musical instrument. Micro-processor controlled musical instruments have been compared and contrasted with conventional instruments, thereby revealing some of their special aspects and distinctions. The introduction of microprocessor music has opened avenues of creative possibilities that music composers have never before seen. A microprocess controlled musical instrument can compose or synthesize any possible combination of musical tones, their pitches may be set to any arbitrary value with tremendous precision.

This work describes the complete software and hardware design of a practical microprocessor controlled musical instrument. We have also developed such a system in our laboratory using Intel 8085 microprocessor. Our system has been interfaced with a conventional musical key-board for real time operation like an Organ. Our system is able to generate tones of Organ, violin, flute, fantasy and horn. It is also able to compose any combination of these five musical instruments. The keyboard can be selected to any desired octav of the above mentioned five instruments. Our musical instrument is able to store a song played on any above mentioned five instruments in its RAM. The stored song can also be replayed on any of the five instruments. This recording and play back system is a unique feature of our microprocessor controlled musical instrument compared to a conventional musical instrument, which needs a cassette recorded to store a song. By changing software one can easily add or change any desired feature in our developed system which is another distinct advantage over any conventional musical instruments.

A block diagram of our microprocessor controlled musical system is shown in figure 1.1. Here a conventional musical key board has been interfaced for real time operation i.e. play on key-board and listen to speaker without any time delay. The key-board is the input to the microprocessor. ROM has been used to store the software programme developed for our system. RAM has been used to record a song. A song is recorded in the RAM in coded form. The key-board, software programme inside the ROM and the programmable 14-Bit binary counter (8155) work togather to generate a desired tone, which is the processed by an analog processing circuit to give the actual wave shape of a particular musical instrument. The audio amplifier has been used to amplify the output of the analog processing circuit to drive the speaker. The detailed descriptions of the hardware and software of the above mentioned musical system are given in the following chapters.

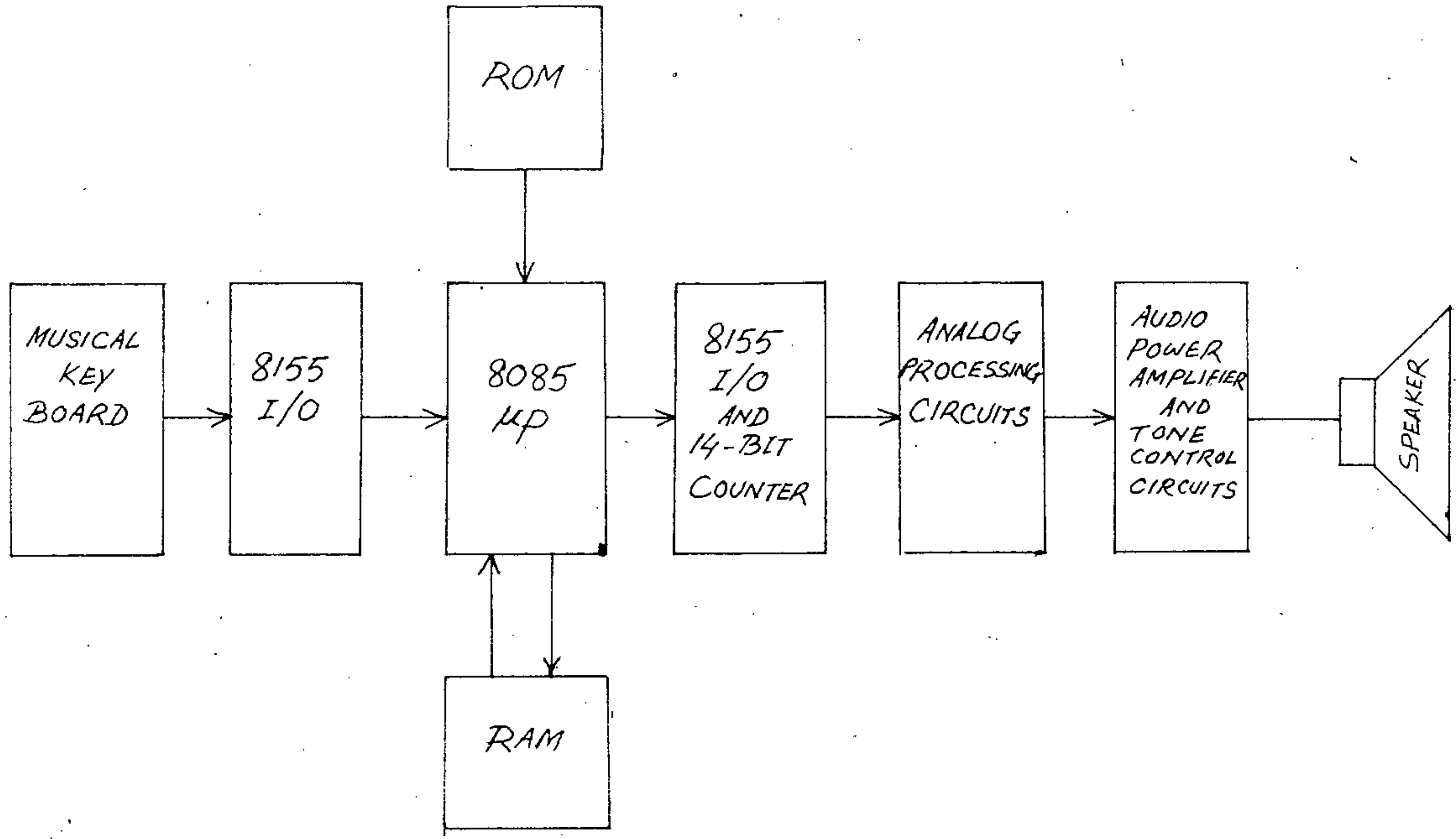


FIG. 1.1 BLOCK DIAGRAM OF THE MICROPROCESSOR CONTROLLED MUSICAL SYSTEM.

CHAPTER - II

CHARACTERISTICS OF MUSICAL TONES

2.1 PARAMETERS OF MUSICAL TONES

The basic parameters of musical tones are frequency (pitch), intensity (loudness), overtones or harmonics content, vibrato and tremolo.

Frequency and pitch are distinguished in that the former is a physical concept, whereas the later is a subjective experience. The unit of frequency is hertz, whereas the unit of pitch is mel. A 1 KHZ sin wave tone, 40 dB above a listener's threshold of hearing, produces a pitch of 1000 mels. The pitch of any sound that is judged by the listeners to be n times that of a 1 mel tone is denoted as n mels.

Inensity and loudness are distinguished in that the former is a physical concept, whereas the later is a subjective experience intensity denotes the strength or amplitude of a sound wave, loudness is a measure of the sensitivity of human hearing to the strength of a sound wave. The strength of a sound wave is measured in microbar units. Loudness is measured in sone or phon units.

Vibrato denotes the frequency modulation of a tone at a rate of approximately 7 Hz. Tremolo denotes the amplitude modulation of a tone at a rate of approximately 7 Hz.

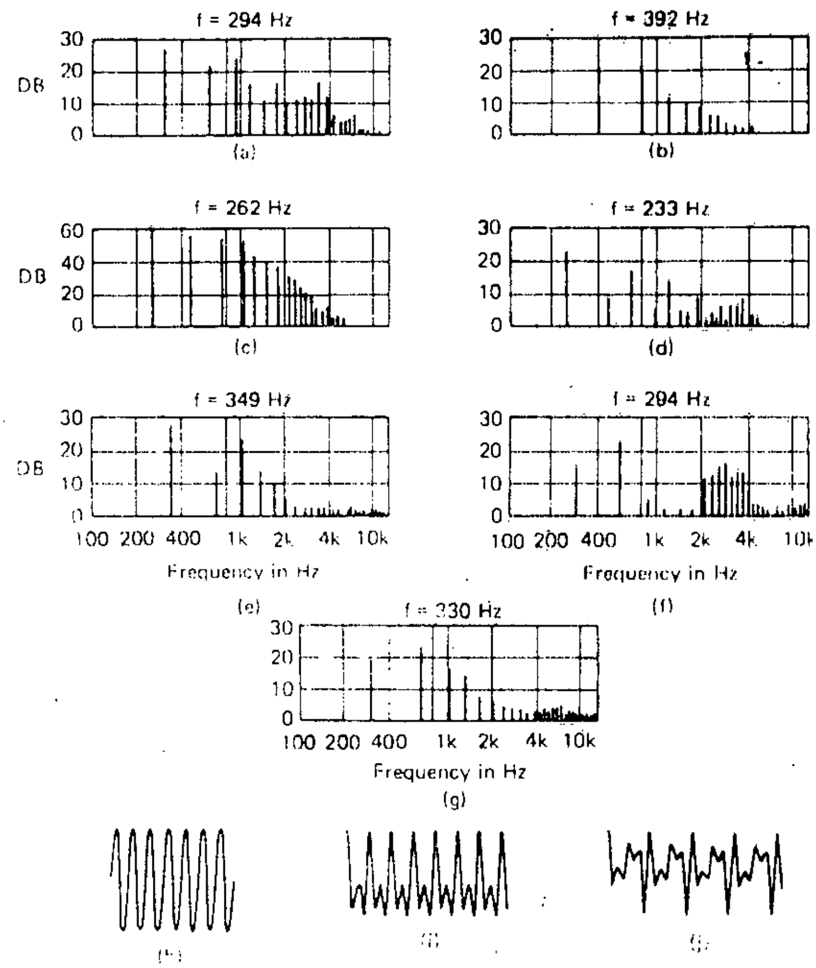
A musical tone has overtones or harmonics. Overtone (harmonics) spectra for various instrumental tones and Organ voices are shown in figure 2.1.

2.2 SCALES OF MUSICAL INSTRUMENTS

Key board music is generally restricted to the 12-note tempered scale. The tempered scale divides the octave into 12 intervals whose pitch ratios are exactly equal. Since the octave interval ratio is 2:1, the pitch interval ratio is twelfth root of 2. Figure 2.2 shows the pitch ratios of the equally tempered scale. Table 2.3 lists the frequencies of the tempered scale, which is used in the great majority of organs, Different musical instruments have different range of frequencies in audio frequency spectrum. Figure 2.4 shows the range of frequencies for various musical instruments.

Interval	Pitch ratio
Prime	1
Semitone	$2^{1/12} = 1.059463$
Whole step	$2^{2/12} = 1.122462$
Minor third	$2^{3/12} = 1.189207$
Major third	$2^{4/12} = 1.259921$
Perfect fourth	$2^{5/12} = 1.334840$
Tritone	$2^{6/12} = 1.44214$
Perfect fifth	$2^{7/12} = 1.498307$
Minor sixth	$2^{8/12} = 1.597401$
Major sixth	$2^{9/12} = 1.681793$
Minor seventh	$1^{10/12} = 1.781797$
Major seventh	$2^{11/12} = 1.887749$
Octave	2

Fig.2.2 Pitch ratio in tempered scale.



(All Harmonics are Octavely Related with Respect to the Fundamental)

Figure 2-1 Overtone (harmonic) spectra for various instrumental tones and organ voices, and waveforms for typical tones. (a) Open violin string; (b) flute; (c) reed organ pipe; (d) clarinet; (e) trumpet; (f) soprano voice; (g) guitar string; (h) pure pitch waveform; (i) "oo" as in "loose;" (j) "o" as in "low.".

C	C#	D	D#
16.35			
32.70	17.32	18.35	19.44
65.41	34.65	36.71	38.89
130.81	69.30	73.42	77.78
261.62	138.59	146.83	155.56
523.25	277.18	293.66	311.13
1046.50	554.36	587.33	622.25
2093.00	1108.73	1174.66	1244.51
4186.01	2217.46	2349.32	2489.01
8372.02	4434.92	4698.64	4978.03
6744.03	8869.84	9397.27	9956.06

E	F	F#	G
20.60	21.83	23.12	24.50
41.20	43.65	46.25	49.00
82.41	87.31	92.50	98.00
164.81	174.61	185.00	196.00
329.63	349.23	369.99	392.00
659.26	698.46	739.99	783.99
1318.51	1396.91	1479.98	1567.98
2637.02	2793.82	2959.95	3135.96
5274.04	5587.65	5919.90	6270.93
10548.08	11157.30	11839.81	12541.86

G#	A	A#	B
25.96	27.50	29.14	30.87
51.91	55.00	58.27	61.74
103.83	110.00	116.54	123.47
207.65	220.00	233.08	246.94
415.30	440.00	466.16	493.88
830.61	880.00	932.33	987.77
1661.22	1760.00	1864.65	1975.53
3322.44	3520.00	3729.31	3951.06
6644.87	7040.00	7458.02	7902.13
13289.74	14080.00	14917.23	15804.26

CCCC = 16.35 Hz is the lowest note of 32 ft pitch
CCC = 32.70 Hz is the lowest note of 16 ft pitch
CC = 65.41 Hz is the lowest note of 8 ft pitch
C = 261.62 Hz is the popularly termed middle C of the keyboard

TABLE : 2-3 FREQUENCIES OF THE TEMPERED SCALE

FREQUENCY RANGES OF MUSICAL INSTRUMENTS AND THE HUMAN VOICE

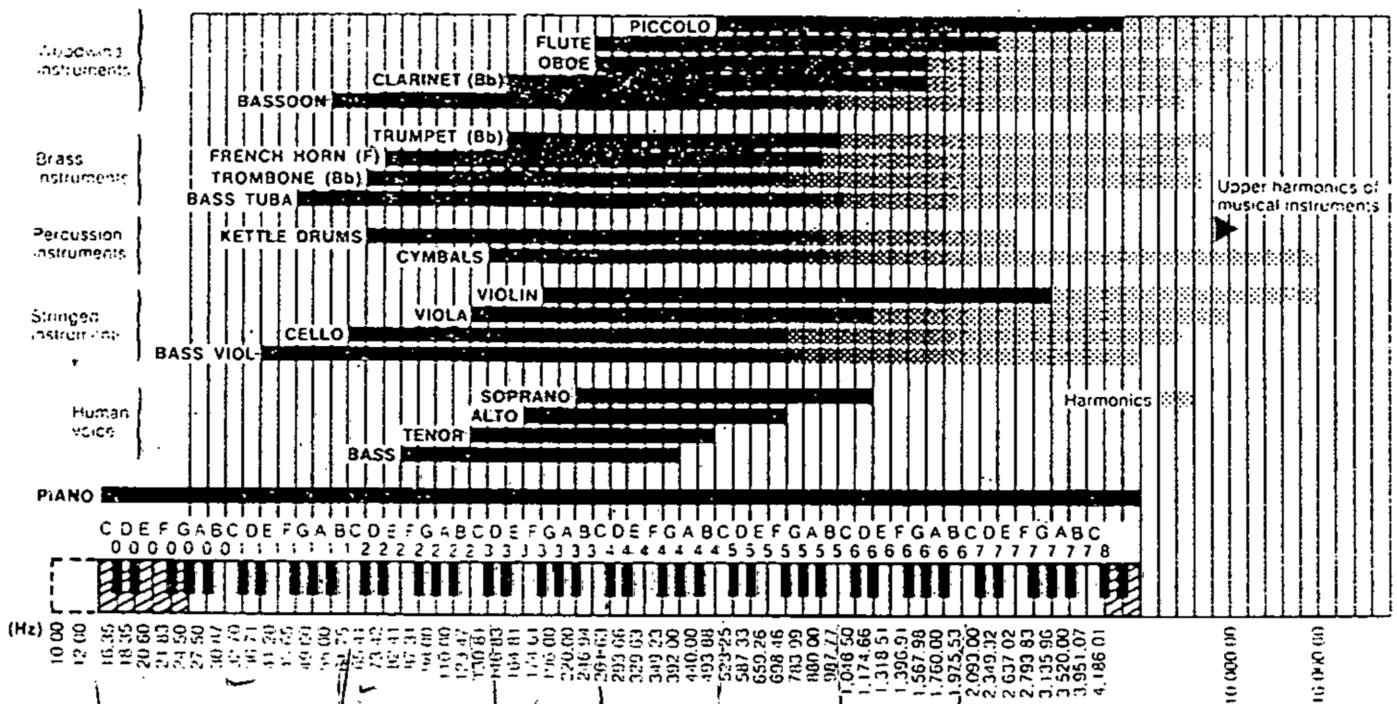


Figure 2-4 Range of frequencies for various sources of sound.

CHAPTER - III

INTRODUCTION TO MICROPROCESSOR

3.1. DESCRIPTION OF 8085 MICROPROCESSOR

We have developed the microprocessor controlled musical instrument by using Intel 8085 microprocessor and peripherals. A brief description of 8085 microprocessor is given below.

The 8085 is an 8-bit general purpose microprocessor, which is capable of accessing up to 64K bytes of memory. The 8085 CPU functional bock diagram is shown in figure 3.1. The 8085 transfers data on an 8-bit, bidirectional 3-state bus (AD_{0-7}) which is time-multiplexed so as to also transmit the eight lower-order address bits. A additional eight lines (A_{8-15}) expand the 8085 system memory addressing capability of 16 bits, thereby allowing

64 K bytes of memory to be accessed directly by the CPU. The 8085 has eight addressable 8-bit registers. Six of them can be used either as 8-bit registers or as 16-bit register pairs. The accumulator (ACC or A register) is an eight-bit register. The programme counter (PC) always points to the memory location of the next instruction to be executed. It contains a 16-bit address. The stack pointer (SP) is a special data pointer that always points to the stack top (next available stack address). It is an indivisible 16-bit register. The flag register contains five one-bit flags, each of which records processor status information and may also control processor operation.

A microprocessor needs to be interfaced to memories for programme and data storage and to input/output devices for the purpose of communicating with environment (which consists of men and machine). Our system has got RAM and ROM arrangements for its memory operation and keyboard, T.V. monitor, cassette recorder and printer interfacing capabilities for its input/output operation.

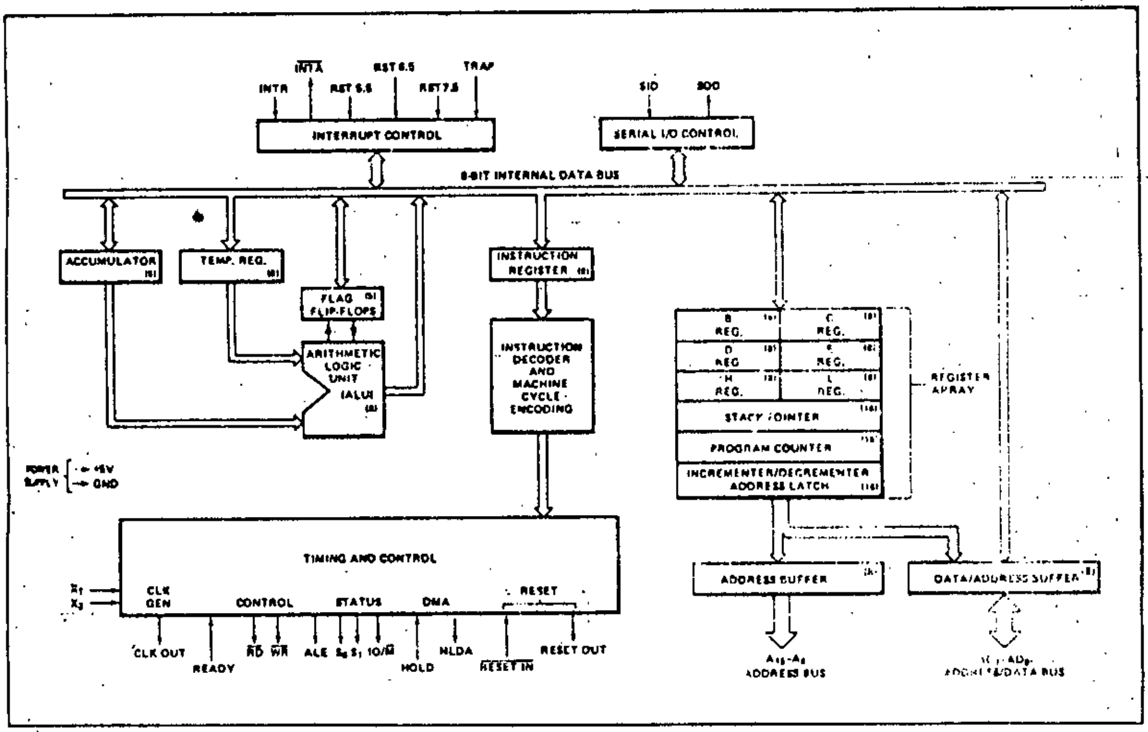


FIG. 3.1 8085 CPU FUNCTIONAL DIAGRAM

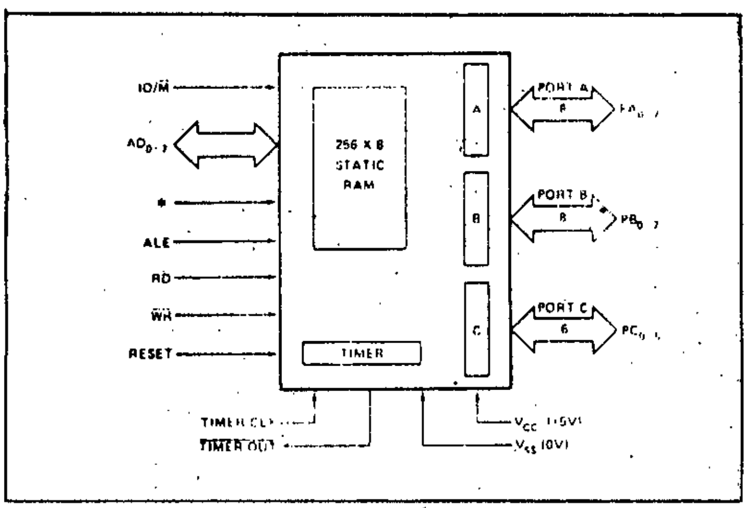


FIG. 3.2 BLOCK DIAGRAM OF 8155

3.2. DESCRIPTION OF 8155 RAM AND I/O PORTS

The 8155 RAM chip has three I/O ports apart from 256 bytes of RAM. Two of these are 8-bit ports and are referred to as PA and PB. The third one, the PC port, is of 6-bits. The ports can be individually addressed, when the chip is selected and IO/$\bar{M}$ line is high, using the least significant 3 bits of the address sent out by the 8085 to the 8155. A block diagram of the 8155 chip is given in figure 3.2. We have used these I/O ports for interfacing our musical key-board with the microprocessor. At the time of system initialization, the three ports may be suitably programmed using the bits of a command status register (CSR). The CSR is an internal 8-bit register in the 8155. The 6 least significant bits of this register can be suitably set or reset to programme the three ports.

The 8155 has a 14-bit programmable counter/timer to provide either a square or terminal count pulse for the CPU system depending on timer mode. The timer is a 14-bit down counter which counts the input pulses. We have used this programmable counter to get the desired frequency corresponding to a particular key of the musical key-board. The input of this counter is the clock output of 8085 microprocessor, the output of the counter is the square wave of desired frequency. The output frequency depends upon the content of the COUNT LENGTH REGISTER. To programme the timer, the COUNT LENGTH REGISTER is loaded first, one byte at a time, by selecting the timer addresses.

CHAPTER - IV

KEYBOARD INTERFACING AND HARDWARE DESIGN

We have intefaced a conventional musical key board to the microprocessor through the input/output ports of 8155 input/output chip. Our key board has 37 individual keys. We have coded these keys by using a 8 by 5 matrix. Figure 4.1 shows the key numbers and corresponding key code. Table 4.2 shows the key connections to form 8 by 5 matrix. The 8 lines (Return lines) have been connected to the input port A of the 8155 chip and the 5 (Scan lines) lines have been connected to the output port B of the 8155 chip. Figure 4.3 shows the connections of 5 lines and 8 lines from the keyboard to the microprocessor output/input ports. The five lines from the keyboard have been connected through open collector inverters to avoid the short circuit if

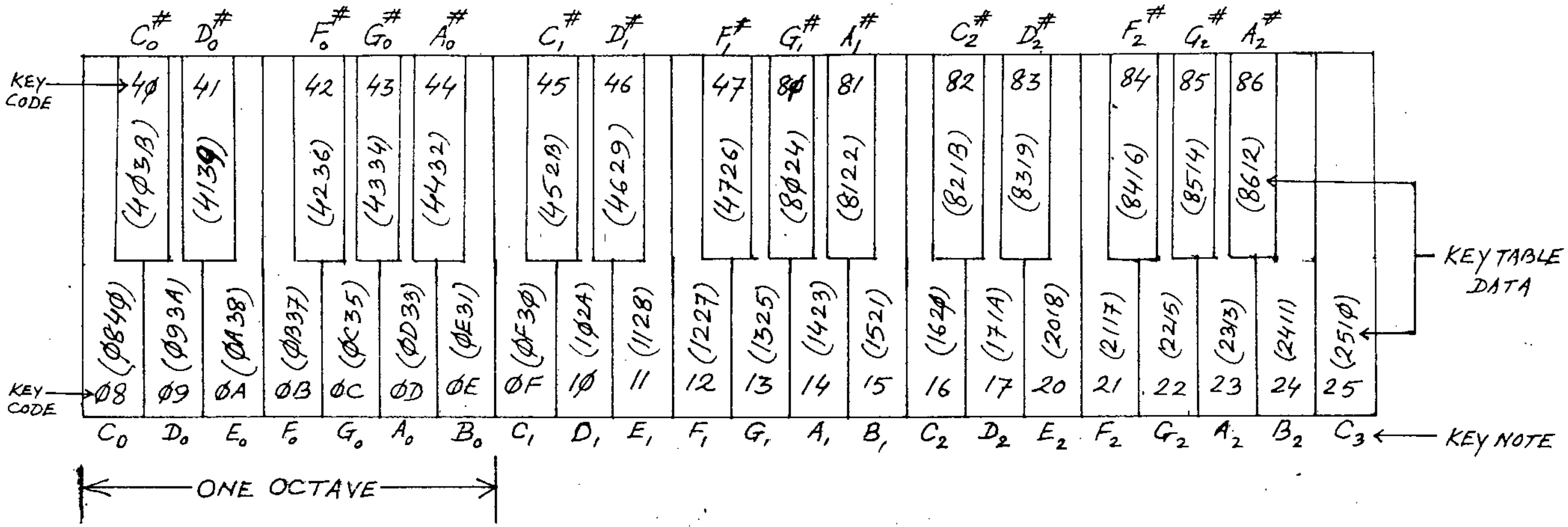
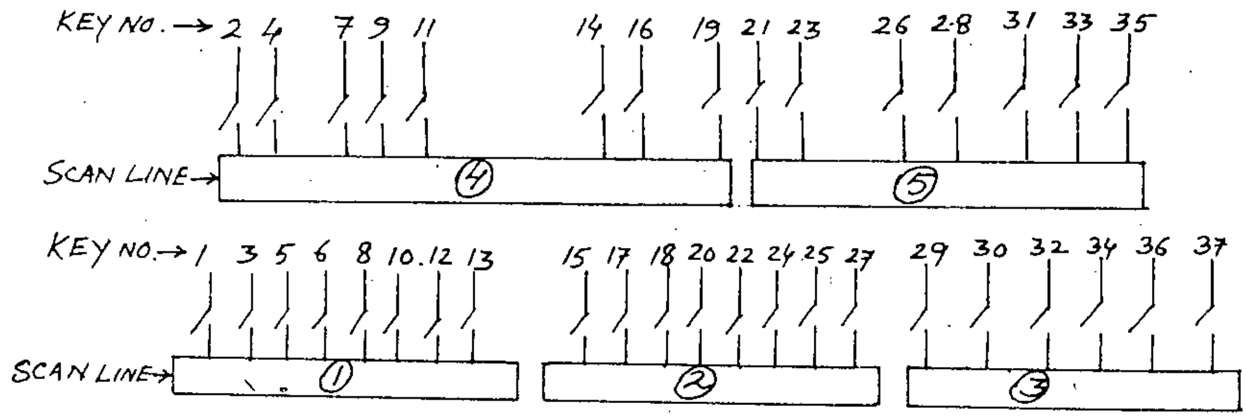


FIG. 4.1. KEY POSITION, KEY CODE AND KEY TABLE DATA



RETURN LINE	KEYS CONNECTED TOGETHER
1 →	2, 21, 1, 15, 29
2 →	4, 23, 3, 17, 30
3 →	7, 26, 5, 18, 32
4 →	9, 28, 6, 20, 34
5 →	11, 31, 8, 22, 36
6 →	14, 33, 10, 24, 37
7 →	16, 35, 12, 25
8 →	19, 13, 27

Scan line?

FIG. 4.2 KEY CONNECTIONS TO FORM 8 BY 5 MATRIX

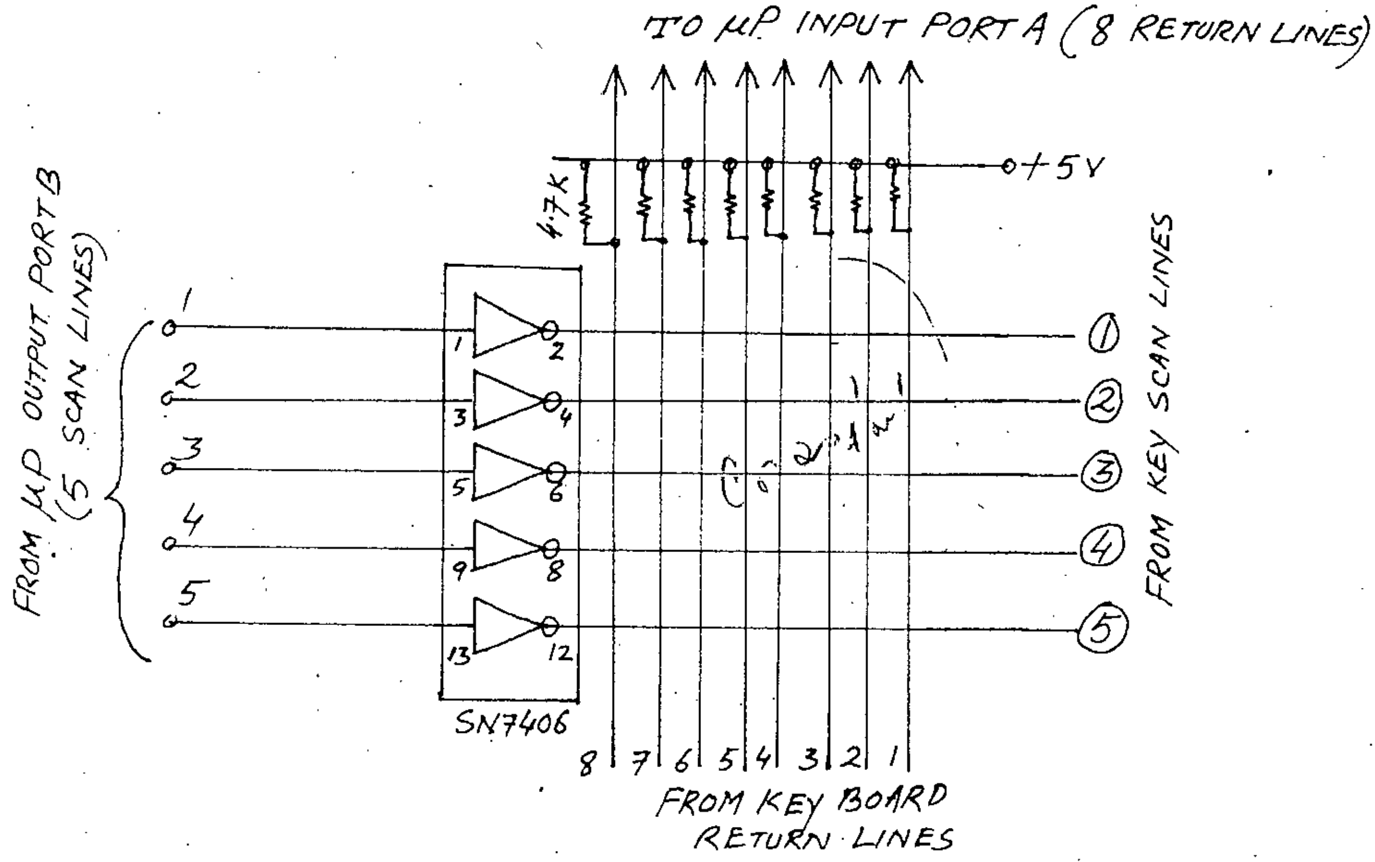


FIG. 4.3 CONNECTIONS OF 5 SCAN LINES AND 8 RETURN LINES TO THE µP

two keys are pressed simultaneously. 8 resistors have been connected between the 8 lines from the keyboard to the +5 volt. So normally the logic level of these light lines are high i.e. logic "1". At the output port B of 8155 a 5 bit data in decoded form is given by the microprocessor in cyclic order at a very high frequency as shown in Table 4.4.

SCAN LINE NO.	PB_4	PB_3	PB_2	PB_1	PB_0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	1	0	0
4	0	1	0	0	0
5	1	0	0	0	0

Table 4.4 5-bit data format at the output port B of 8155.

Simultaneously the microprocessor moniters the 8 input lines (Return lines) at port A of 8155. If a key is pressed and if the corresponding scan lines is high a logic low is obtained at the corresponding input return lines. So by sending a 5 bit data to the scan lines and simultaneously monitering the 8-bit return lines, the microprocessor is able to indentify and code a closed key form the scan lines data and return lines data by using software programme. The key code is a 8-bit binary number. The 3 least significant bits represent the coded form of the 8-bit binary data obtained from the return lines. A software counter has been used to code the 8-bit data obtained from the return lines into a 3-bit binary number. The 5 most significant bits of the key code represent the scan lines data in decoded form. The 8-bit data format of a key code is shown in figure 4.5.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Scan lines data in decoded form

Coded form of return lines data

Figure 4.5 8-bit data format of a key code.

The software programme to identify a pressed key and to generate the key code has been described in chapter 5.

CHAPTER - V

SOFTWARE PROGRAMME DEVELOPMENT

5.1. SOFT-WARE PROGRAMME DEVELOPMENT FOR MUSICAL TONE GENERATION.

62085

We have developed a software programme to generate different musical tones using Intel 8085 microprocessor and a conventional keyboard interfaced with the microprocessor. A flow diagram of this software programme is shown in fig.5.1. To perform the operations as shown in flow diagram of fig.5.1 eight sub-programme are used. These sub-programs are called by the main program (MAINP) in cyclic order. The names of these sub-programs are REFR, INACT, ACTKC, KEY, FRKC, STOP, KB and STIME. The functions of these sub-programs are described below. The different variable names used in these sub-programs are given in Table 5.2 with their functional descriptions. Many memory locations are used for the different variables used in different sub-programs. The memory allocation of different variable

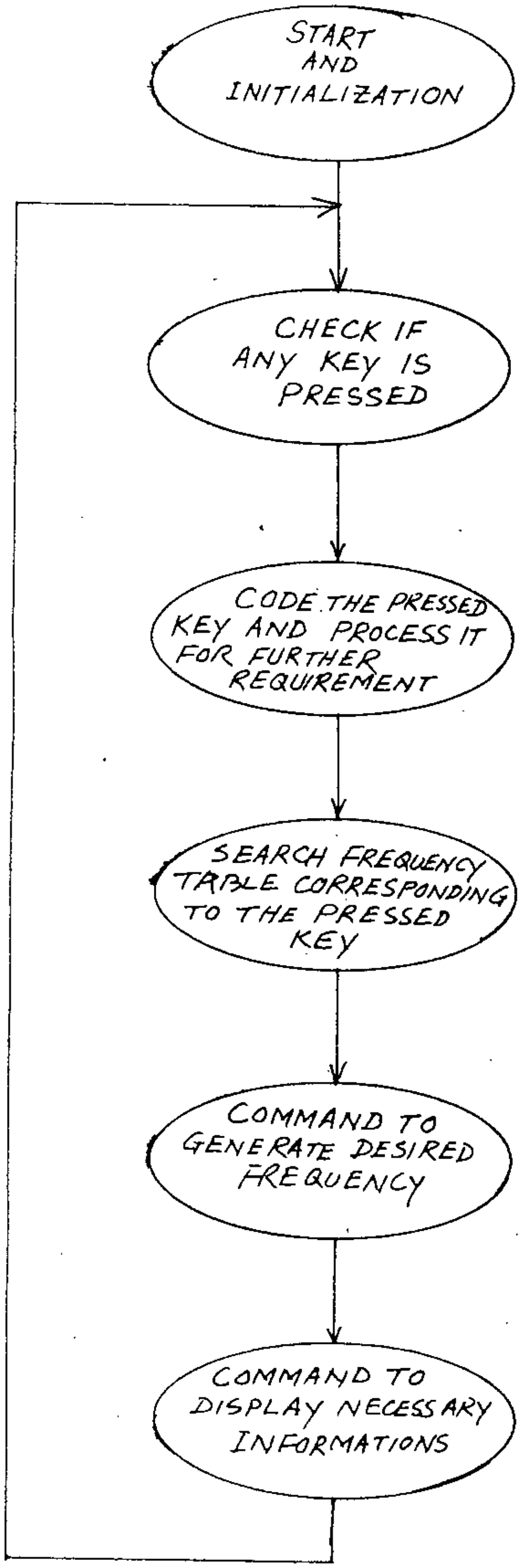


FIG. 5.1
FLOW DIAGRAM OF THE
MAIN SOFT-WARE PROGRAM.

TABLE 5.2

NAMES OF VARIABLES AND THEIR FUNCTIONAL DESCRIPTION:-

NAME	DESCRIPTIONS:
MAINP	Main program which calls different sub-programs.
REFR	Refresh sub-program, which is used to generate the key code.
INACT	Inactive Key check sub-program.
ACTKC	Active key check sub-program.
KEYP	Key on process sub-program.
FRKC	Fresh Key check sub-program.
STOP	Stop sub-program which is used to stop the tone generation when a key is released.
KB	Key-board sub-program which is used to control some operations directly from the microprocessor key-board and also used to display some informations for the users.
STIME	Store time sub-program, which is used to store the time duration of a pressed key in a memory location for recording operation.
NTAB	Name table for new pressed keys. A table which contain a maximum of four key codes.
ACTK	Name of the memory location to store the code of the active key.
COM	Name of the 8-bit memory location, which represents the status indicator for different operations.
INATB	Inactive table, which contains the key code of inactive keys.
INTBE	Memory location, which represents the end of the inactive key table.
INSTN	Memory location to store the actave number.
PRCNT	Memory location to store the number corresponding to prolonged time for sustained operation.
MTAB	Memory location, which contains the first address of memory table of a recorded song.
CNTRL	Low-order counter to store the time of a pressed key for recording operation.

Contd...

NAME	DESCRIPTIONS
CNTRH	: High-order counter to store the time of a pressed key for recording operation.
DBCNT	: Debounce counter to store the number for debouncing operation.
PKEY	: Memory location to store the key code of the key on process.
FRQTB	: Frequency table which contains the number corresponding to each key of the zero octave.
NMTAB	: Name table, which contains the names of a recorded song.
NMATB	: Name address table, which contains the starting address of a name in the name table.
NMTBP	: Name table pointer, which contains the serial number of a name.
PTAB	: Memory location, which contains the starting address of the play table for auto-play operation.

are shown in figure 5.3. A system status indicators byte named COM is used to flag the current state of the different operations. The bit assignment of COM for different operations is shown in figure 5.4.

Our system uses a conventional musical key-board, which consists of 37 individual keys. If a key is pressed the microprocessor generates a tone corresponding to that key. If two or more keys are pressed simultaneously the soft-ware program permits the microprocessor to generate tones corresponding to the last key pressed. We define the state of a key in five possible ways inactive key, active key, key on process, fresh key and the normal open key. A key passes through a series of states till it is made ready to the processed for tone generation. A key state diagram is shown in figure 5.5.

5.1.1. REFR SUB-PROGRAM

This sub-program sends the 5-bit decoded data to the scan lines and monitors the 8-bit data from the return lines to code a pressed key. The code of a pressed key is stored in NTAB in sequence. NTAB refers to the first address of a 5-byte memory location as shown in figure 5.3. The first four bytes are used

Variable	Address
NTAB	F8ØØ – F8Ø5
ACTK	F8Ø6
COM	F8Ø7
INATB	F8Ø8 – F8ØB
INTBE	F8ØC
INSTN	F8ØD
PRCNT	F8ØE – F8ØF
MTAB	F81Ø
CNTRL	F812
CNTRH	F813
CKEN	F814
DBCNT	F815
PKEY	F816
ZERO	F817
FRQTB	F818 - F82F
NMTBP	F83Ø
NMTAB	F831
ØFF	F863
PTAB	F864-F865
NMATB	F866
PNTR	F870-F871
WORKA	F872-F873
INSST	F874

*

Variable	Address
ASPD	F875 - 76
LIST PRGR	F877 - 79
AUTO PRG	F87A - 7C
INSTD PRGR	F87D - 7F
TAPRW PRGR	F883 - 85
PRNT PRGR	F886 - 88

FIG. 5.3. MEMORY ALLOCATIONS FOR THE VARIABLES USED IN THE SYSTEM

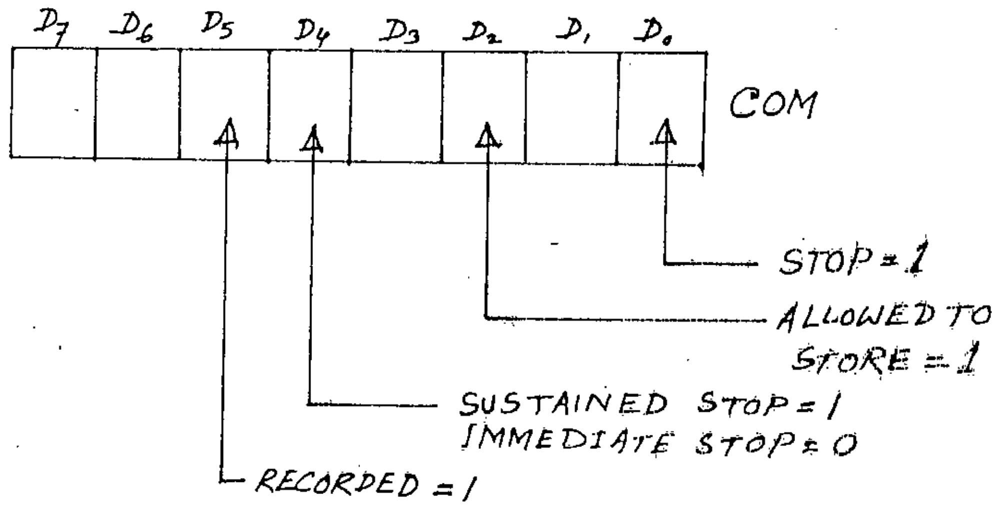


FIG. 5.4. BIT REPRESENTATION OF COM.

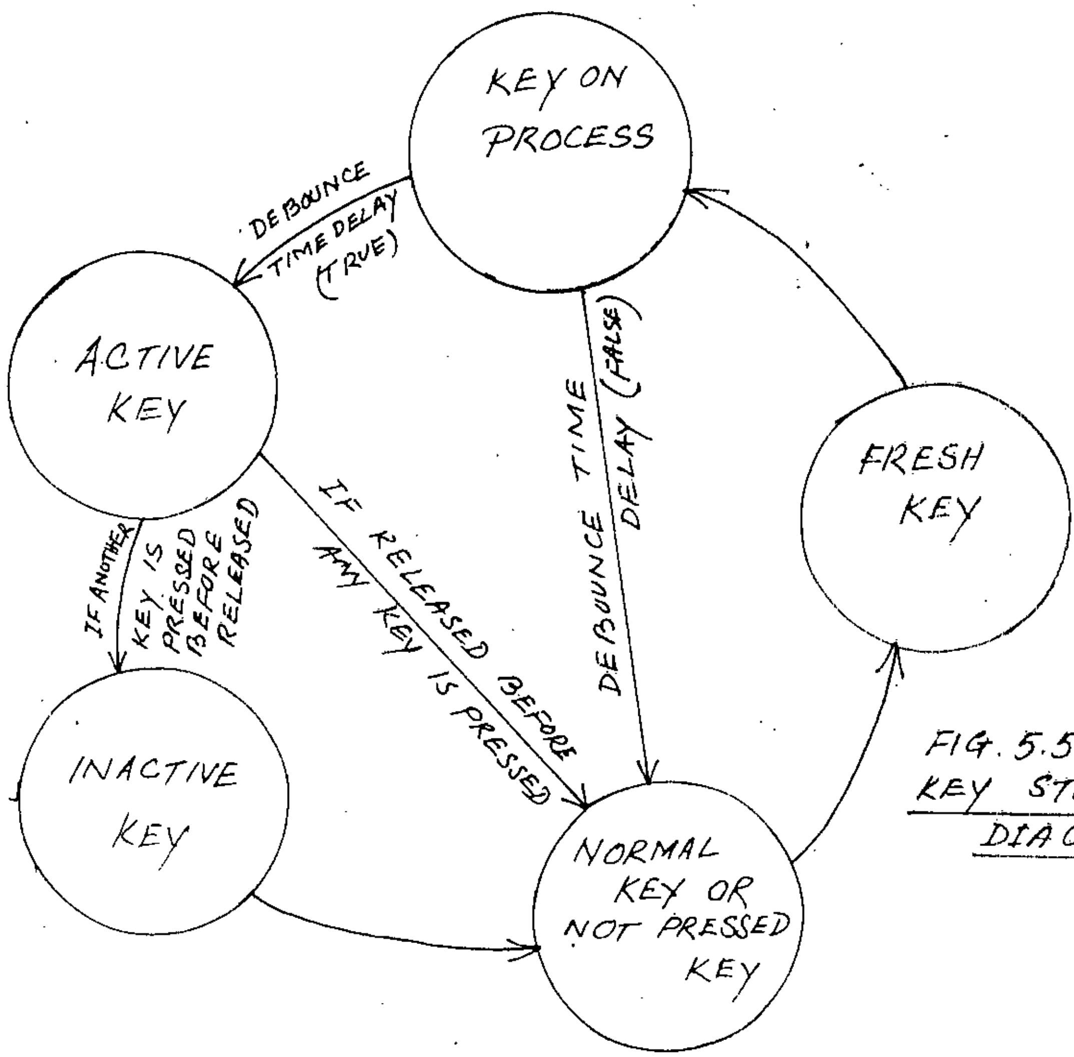


FIG. 5.5 KEY STATE DIAGRAM

to store the key code. The last byte contains FF, which indicates the end of table. The REFR sub-program has been used to store a maximum of 4 key codes in NTAB, when the keys are pressed simultaneously. The flow chart of REFR sub-program is shown in figure 5.6.

5.1.2. ACTKC SUB-PROGRAM

ACTKC is the active key check sub-program, which is used to check if there is any active key. Active key denotes the last key pressed beside the inactive keys. If two or more keys are pressed simultaneously the last key is the active key. The remaining pressed keys are termed as inactive keys. Inactive key codes are stored in a table named as INATB. INATB refers to the first address of a 5-byte memory location as shown in figure 5.3. The first 4 bytes are used to store the key codes of inactive keys, the last byte contains FF, which indicates the end of the table. The key code of an active key is stored in memory location named ACTK. This subprogram also initialises the value of prolong counter (PRCNT). This counter is used to obtain the sustained operation of a tone i.e. a tone is sustained for a while after the key is released. The content of PRCNT determines the time of the sustained operation. The flow chart of ACTKC sub-program is shown in figure 5.7.

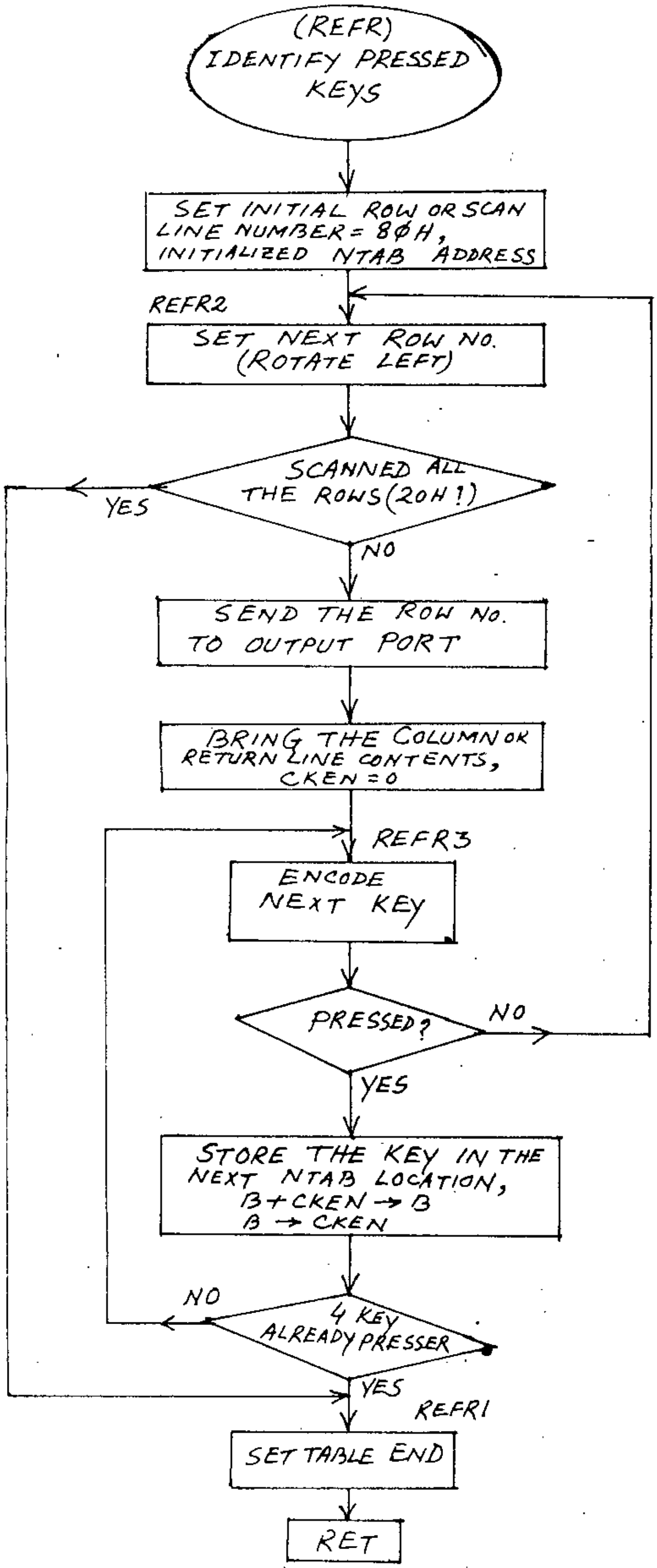


FIG. 5.6.
REFR FLOW CHART

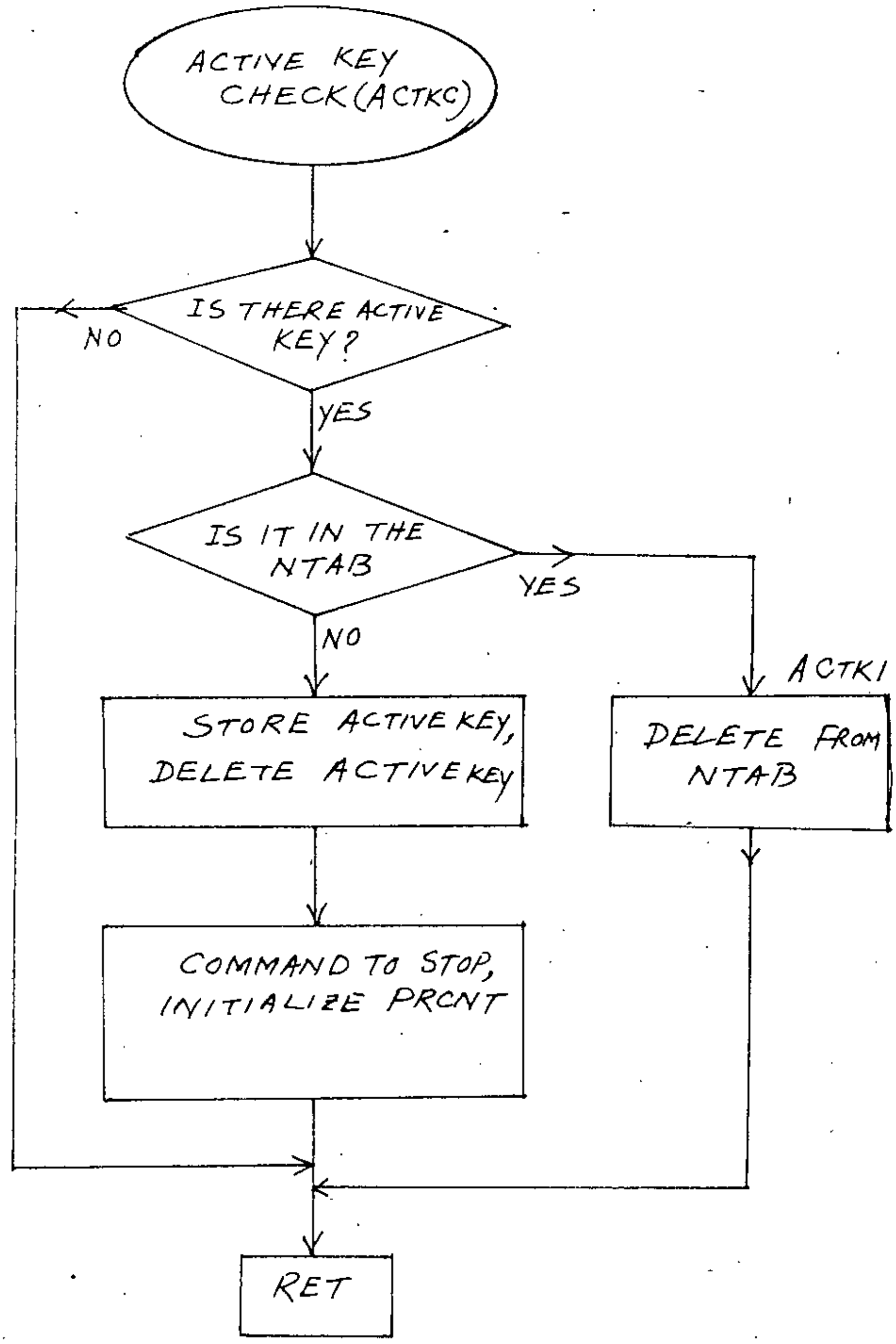


FIG. 5.7 FLOW CHART FOR ACTKC SUB-PROGRAM

5.1.3. INACT SUB-PROGRAM

INACT is the inactive key check sub-program, which is used check if there is any inactive key in inactive key table (INATB). This sub-program also compares the key codes of inactive key table (INATB) and new key table (NTAB). If a same key code is found in the both table, then this sub-program deletes the key code from NTAB. If a key code is found in inactive key table but not found in NTAB, then this sub-program deletes the key code from inactive key table (INATB). Because the key corresponding to this key code has already been released, so we do not require it any more. The flow chart of this sub-program is shown in figure 5.8.

5.1.4. KEYP SUB-PROGRAM

KEYP is the key on process check sub-program, which is used to indentify if there is any key on process. The key code of a key on process is stored in memory location PKEY. This sub-program checks a key on process i.e. a pressed key 20 times to reduce debouncing. A memory location DBCNT (Debounce counter) contains the number 20. This number can be changed if desired. By increasing this number debouncing effect can be reduced more. After observing 20 times this sub-program calls another sub-program named SEND

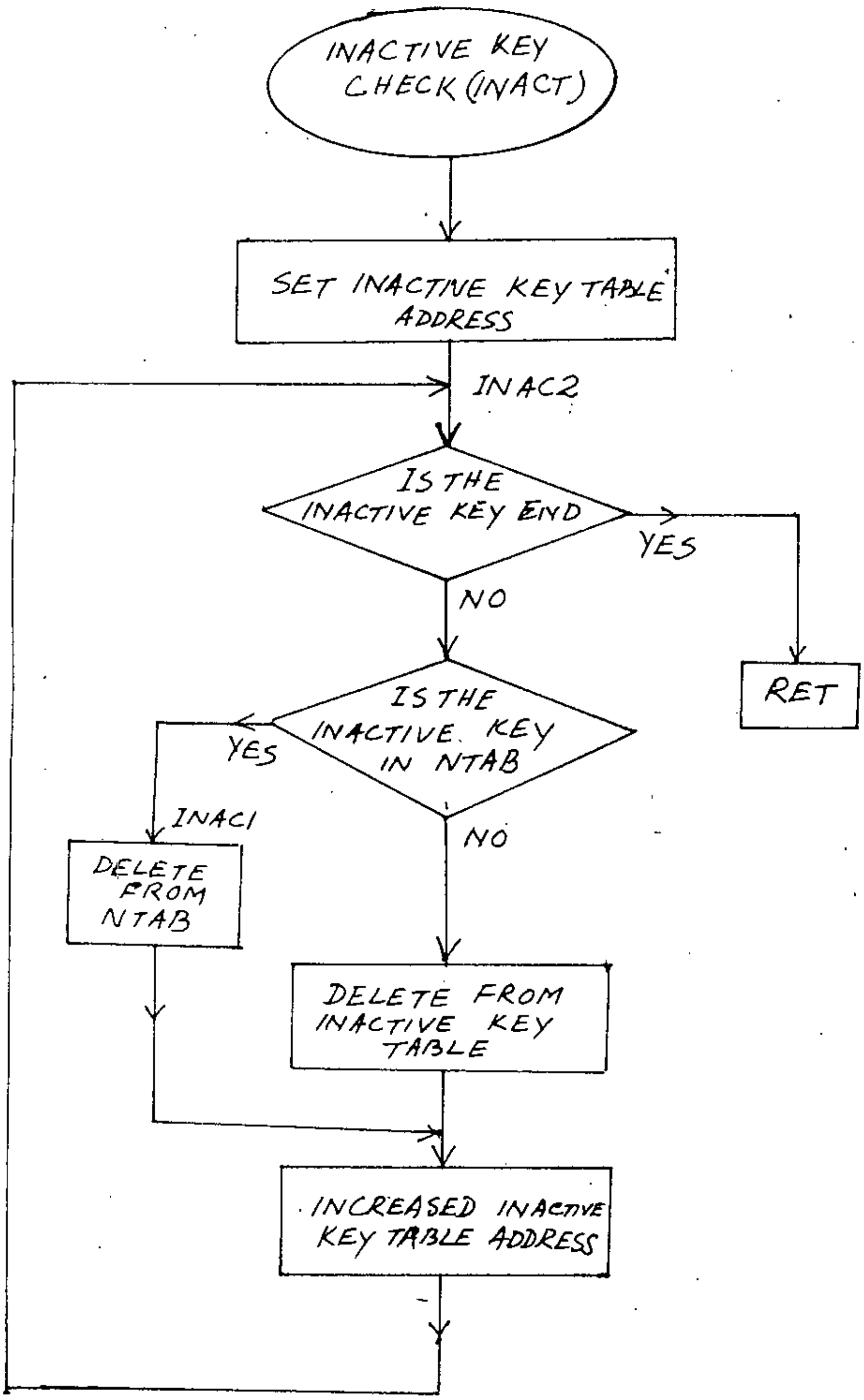


FIG. 58, FLOW CHART FOR INACT SUB-PROGRAM.

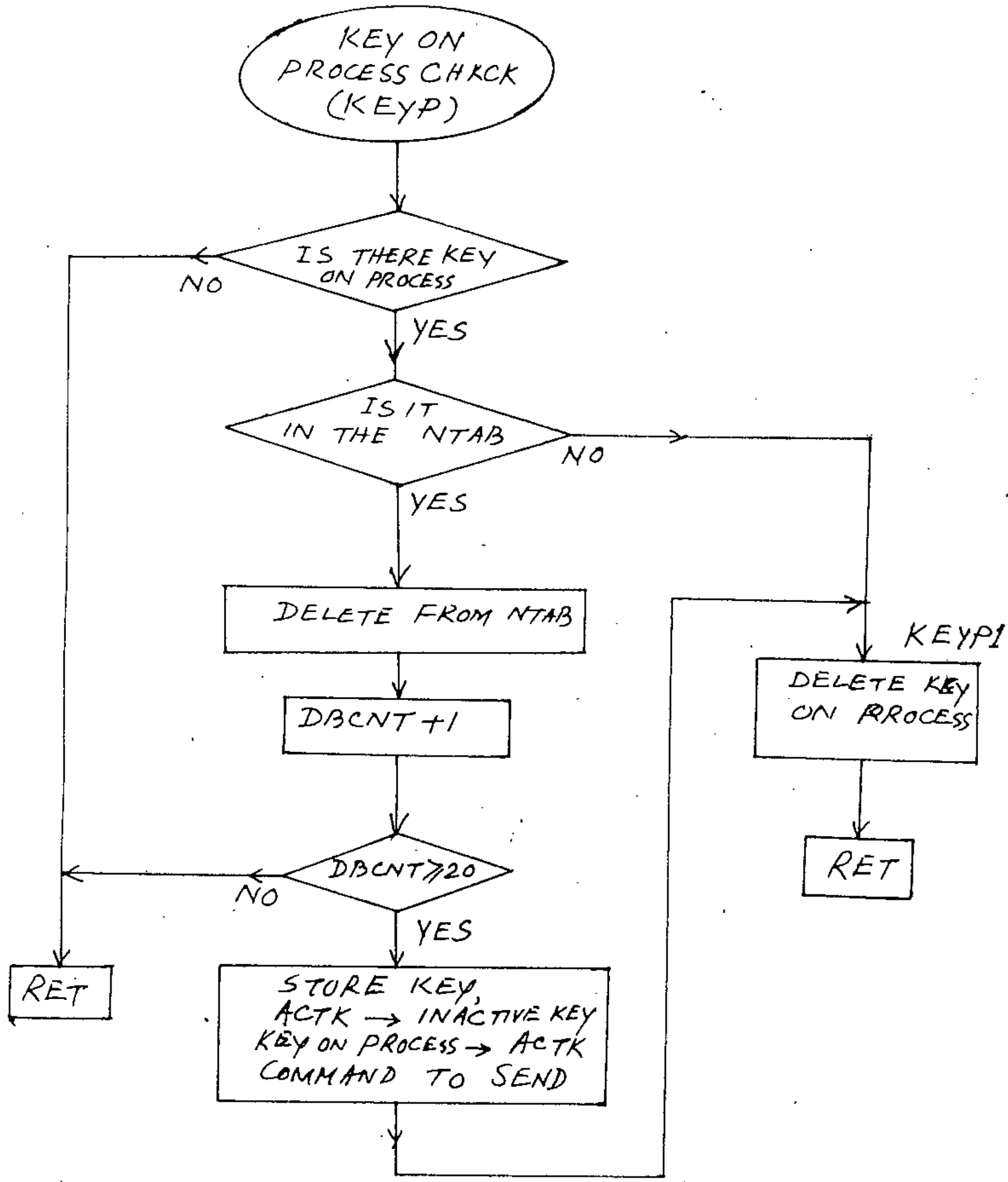


FIG. 5.9. FLOW CHART FOR KEYP SUB-PROGRAM

which allows the 14-bit programmage counter (8155) to give necessary tones corresponding to the pressed key. This sub-program also calls another sub-program named STRK which stores the pressed key codes for recording purpose. The flow chart of this sub-program is shown in figure 5.9.

5.1.5. FRKC SUB-PROGRAM

FRKC is the fresh key check sub-program, which checks if there is any fresh key. A fresh key is the key which is just pressed. This sub-program keeps the code of the key just pressed in the PKEY memory location for further processing. The flow-chart of this sub-program is shown in figure 5-10. This sub-program also initialise the value of DBCNT to zero if a fresh key is detected.

5.1.6. STOP SUB-PROGRAM

This sub-program is used to stop the 14-bit programmable counter (8155) when a key is released by comparing the stop bit of the system status indicator (COM). The COM is a 8-bit memory location for different system status indications. This sub-program also checks the sustained bit of COM to perform sustained stop operation. The flow chart for this sub-program is shown in figure 5-11.

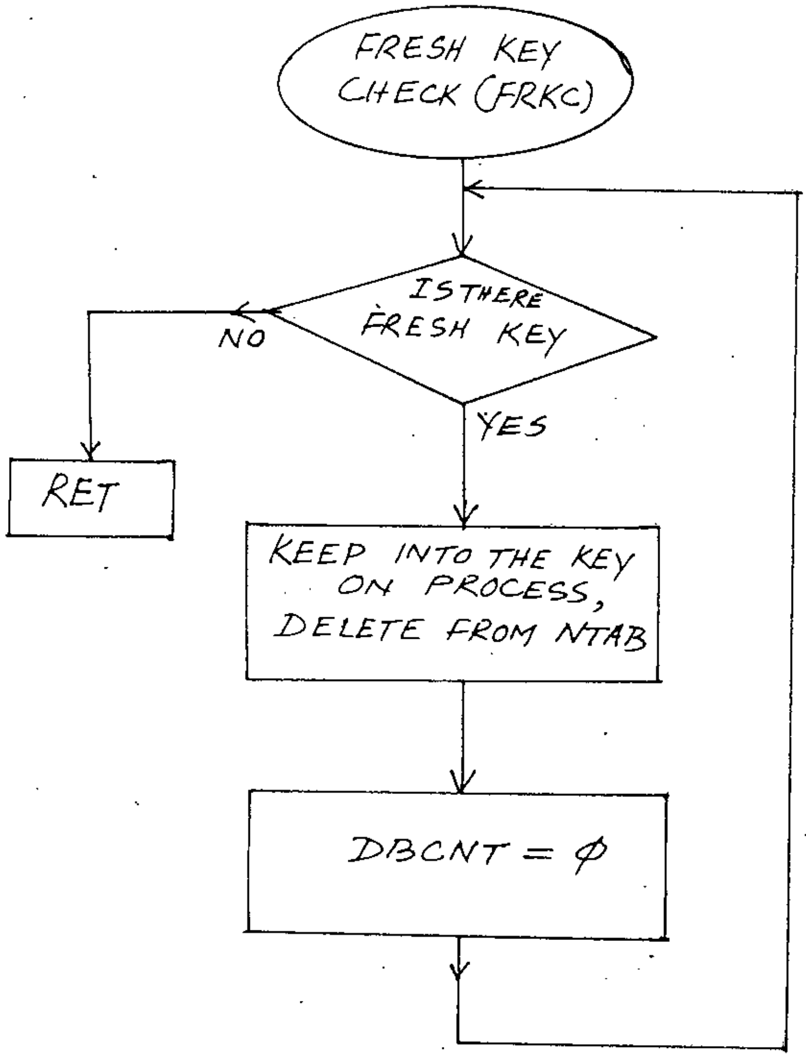


FIG. 5.10. FLOW CHART FOR FRKC SUB-PROGRAM

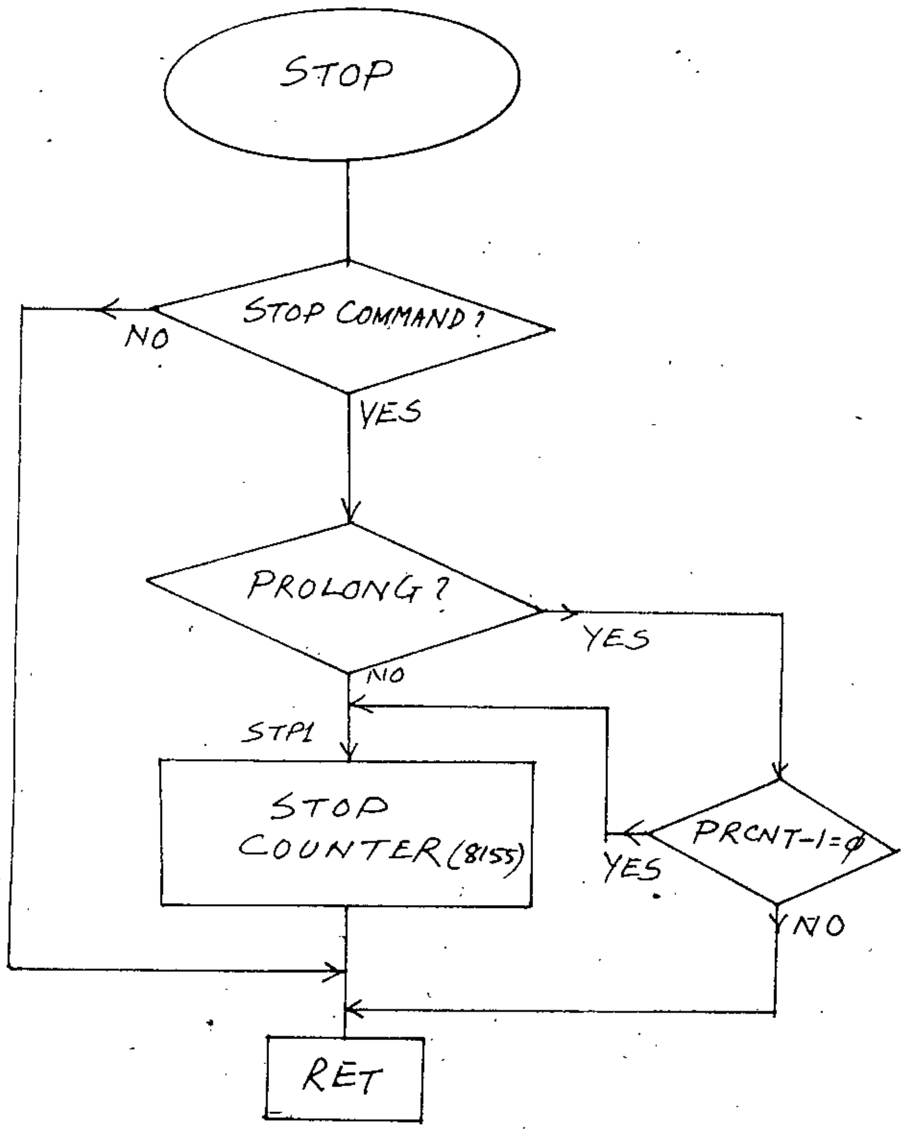


FIG. 5.11. FLOW CHART FOR STOP SUB-PROGRAM

5.1.7. SEND SUB-PROGRAM

This sub-program compares the content of ACTK i.e. the code of a pressed key, to the contents of a key table named KEYTB to get the key position number within an actave and octave number from amongst the 3½ Octave of the keyboard. The KEYTB consists of 37 data word corresponding to the 37 keys of the musical key board. This 37 keys represents the 3½ Octave with 12 keys in each Octave. The key position and Octave arrangements are shown in Fig. 4.1. The high order byte of the data-ward of KEYTB represents the key code, the 4 least significant bits of the low order byte represents the key position number i.e. 1st key, 2nd key etc. within an Octave, the 4 most significant bits of the low order byte is the Octave number in which the key is placed on the keyboard. This program also searches a frequency table named FRQT to get a data used to calculate the required number which is to be loaded to the 14-bit counter (8155) to obtain the tones of desired frequency.

The frequency table (FRQT) contains 12 data -words corresponding to 12 keys in the highest (zero) Octave. The frequency table is shown in figure 5.12. The number in the frequency table represents

KEYTABLE CODE	KEY NOTES	FRQT TABLE DATA	OUTPUT FREQUENCY OF COUNTER (IN KHz)
251Ø	C_3	ØØ87H	16.744
2411	B_3	ØØ8EH	15.804
8612	$A_2^{\#}$	ØØ97H	14.917
2313	A_2	ØØAØH	14.080
8514	$G_2^{\#}$	ØØA9H	13.289
2215	G_2	ØØB3H	12.541
8416	$F_2^{\#}$	ØØBEH	11.839
2117	F_2	ØØCAH	11.157
2018	E_2	ØØD5H	10.548
8319	$D_2^{\#}$	ØØE2H	9.956
171A	D_2	ØØEFH	9.397
821B	$C_2^{\#}$	ØØFDH	8.886

FIG. 5.12 FREQUENCY TABLE

61 10746 / 10465

the data which is to be sent to the timer to get the required tone frequency within the highest (Zero) Octave. The necessary data for the key in the same position in another Octave can be calculated using the formule :-

$$d_n = d_o \cdot 2^n$$

Where, d_o = the data for a key in the zero Octave (from frequency table)

d_n = The data for a key in the corresponding position of the n th Octave.

n = Octave number.

Octave number can be selected from the microprocessor key-board (between 0-5) and is stored in the memory location named INSTN. The content of INSTN represents a number in reference to zero Octave. So if a key is pressed from amongst the key in the 2nd octave of the key-board, then its data can be calculated from the following equation :-

$$d_{1+3} = d_o \, 2^{1+3}$$

Where 1 = selected Octave number

The general form of the formula is given below -

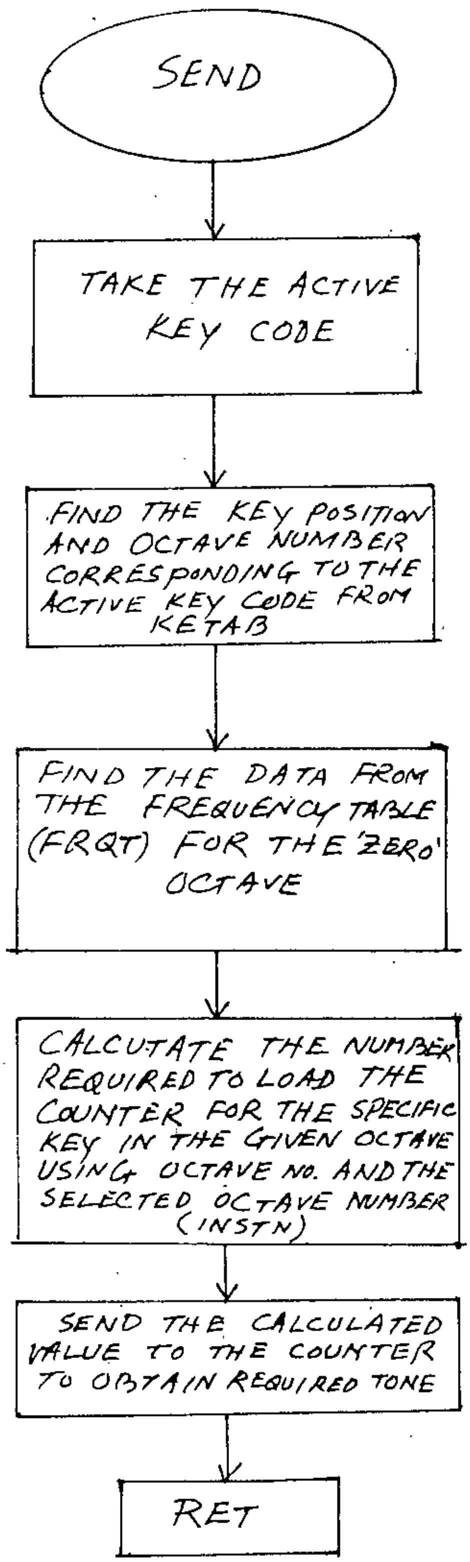


FIG. 5.13. FLOW CHART FOR SEND SUB-PROGRAM

$$d_{1+m} = d_o \cdot 2^{1+m}$$

Where, m = Octave number within the keyboard obtained from the KEYTB (Lies between 0 and 3).

So after getting the key code, the key position within an Octave and the Octave number can be found from the KEYTB and the corresponding data to be loaded into the couhter can be found from FRQT. The musical key board can be tuned to any desired frequency range by changing the 12 data-words of the frequency table (FRQT). The flow chart for the SEND sub-program is shown in figure 5.13.

5.2. SOFTWARE PROGRAM DEVELOPMENT FOR RECORDING

The developed microprocessor controlled musical system is able to record a song in its RAM and is also able to replay the recorded song. A song is recorded with a song name supplied by the users. A maximum of 5 songs can be recorded with their names. This system permits maximum 8 characters for a song name. To perform the record and auto-play operation three sub-programs named STRK, APLAY and STIME are used. The functions of these sub-programs are described below.

5.2.1. STRK SUB-PROGRAM

STRK sub-program is the store key sub-program which stores the key codes of the pressed keys sequentially in a memory location, whose starting address is given by the content of MTAB. The time duration of a pressed key is also stored in the subsequent memory location by using a sub-program named STIME (STORE TIME). One byte of memory is used to store a key code and the next twobytes are used to store the counter value(CNTRL, CNTRH) which represents the time duration of a pressed key. CNTRL is the low-order counter and CNTRH is the high order counter. The flow chart of the STRK sub-program is shown in figure 5-14. The last byte of a recorded song is 'FF' indicating the end of the song.

5.22. APLAY SUB-PROGRAM

APLAY sub-program is the autoplay sub-program which is used to play a recorded song. At first this program takes a key code from the memory location of a recorded song, and then calls the SEND sub-program to get the tones corresponding to the key code. The time interval between two pressed keys is compensated by delay loop. The flow chart of APLAY sub-program is shown in figure 5.15.

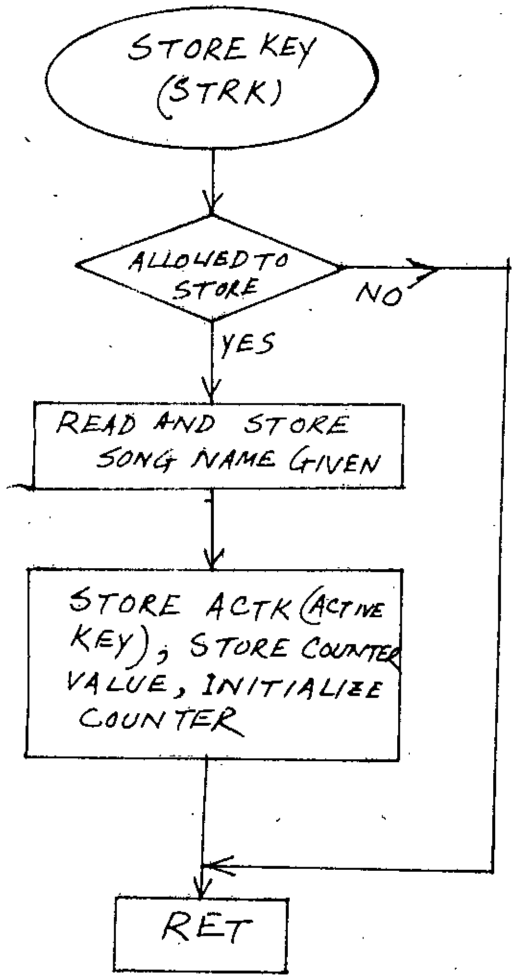


FIG. 5.14. STRK FLOW-CHART DIAGRAM

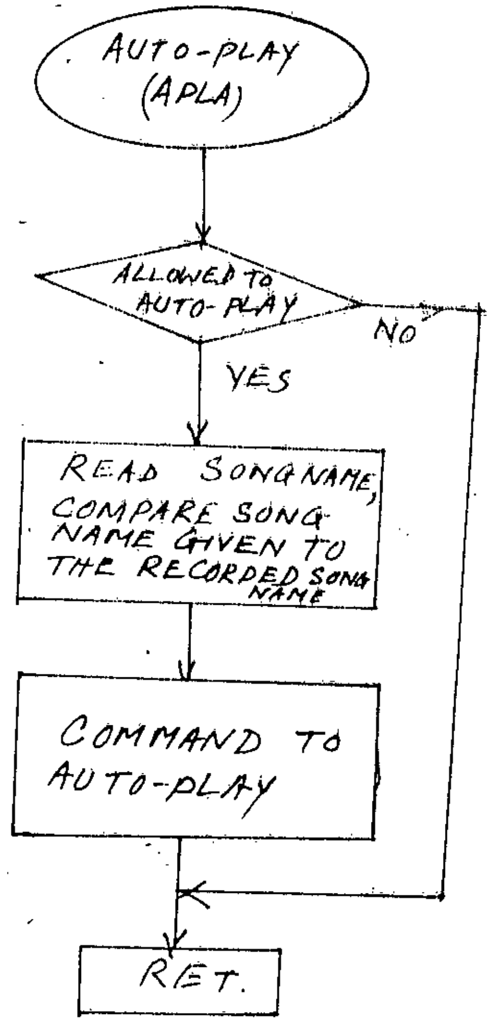


FIG. 5.15 APLA FLOW CHART.

5.3. SOFT-WARE PROGRAM DEVELOPMENT FOR DISPLAY, MONITORING AND CONTROLLING.

A soft-ware program has been developed to display necessary informations for the user, to monitor different operation modes and to control different operations from the microprocessor key-board. To perform these operation two sub-programs named KB and GRAPH are used. The descriptions of these sub-programs are given below.

5.3.1. KB SUB-PROGRAM

This sub-program is used to perform some operations directly from the microprocessor keyboard and to display some necessary informations for the users. Octave number can be selected by pressing the desit 0-5 of the key-board, this is done by putting the pressed number to the INSTN memory location. The selected Octave number is also displayed. Immediate stop operation is performed by pressing the character 'I' of the keyboard. When 'I' is pressed this sub-program sets the corresponding bit of COM to zero for immediate stop. The word IMMEDIATE is also displayed when 'I' is pressed. The sustained stop operation is done by pressing the character 'S' of the keyboard. When 'S' is pressed the corresponding bit of COM is set to '1' for sustained stop operation and also the word SUSTAINED is displayed on the screen. There are two modes of operations for the musical keyboard, one in Monophonic mode and the another is X-phonic mode. Monophonic mode represents the generation of musical tone corresponding to a single pressed key. X-phonic mode represents the operation of musical tones corresponding to multiple pressed keys simultaneously. Monophonic mode can be selected by pressing the character 'M' of keyboard. When 'M' is pressed the word M-PHONIC is display on the screen by this sub-program. X-phonic mode can be selected by pressing the character 'X' of the keyboard. When X is pressed this sub-program puts ØFF at the first memory location of INATB and also displays the word X-PHONIC on the screen. In X-phonic mode there ia no inactive key, all of the keys pressed simultaneously are treated is active keys. The record mode can be selected by pressing character 'R' is pressed the word RECORD-MODE and SONG-NAME are displayed on the screen. A song name of maximum 8-character should be given by the users,which is stored and displayed by the name read (NREARD) and name store (NSTOR) sub-programs. Auto-play operation is performed by pressing the character 'A' of the key-board. When 'A' is pressed and main program calls the APLAY sub-program to perform auto play operation. The song name which is to be played should be given by the user. This sub-program compares the supplied name to the previously recorded song name by using a match sub-program (MATCH) and if the same song name is found from the previously recorded song names, this sub-program allows auto play operation. If no name is found from the previously recorded song name to match with the name supplied for auto play operation, then a message NO NAME is display by this subprogram. The flow chart of KB sub-program is shown in figure 5.16.

5.32 GRAPH SUB-PROGRAM

This sub program is used to generate a graphic representation of the developed microprocessor controlled musical instrument. A picture of the graphic display is shown in fig. 5.17. This sub-program is also used to display necessary informations for the users on the screen.

The complete software program for the developed microprocessor controlled musical system is given in the Appendix A.

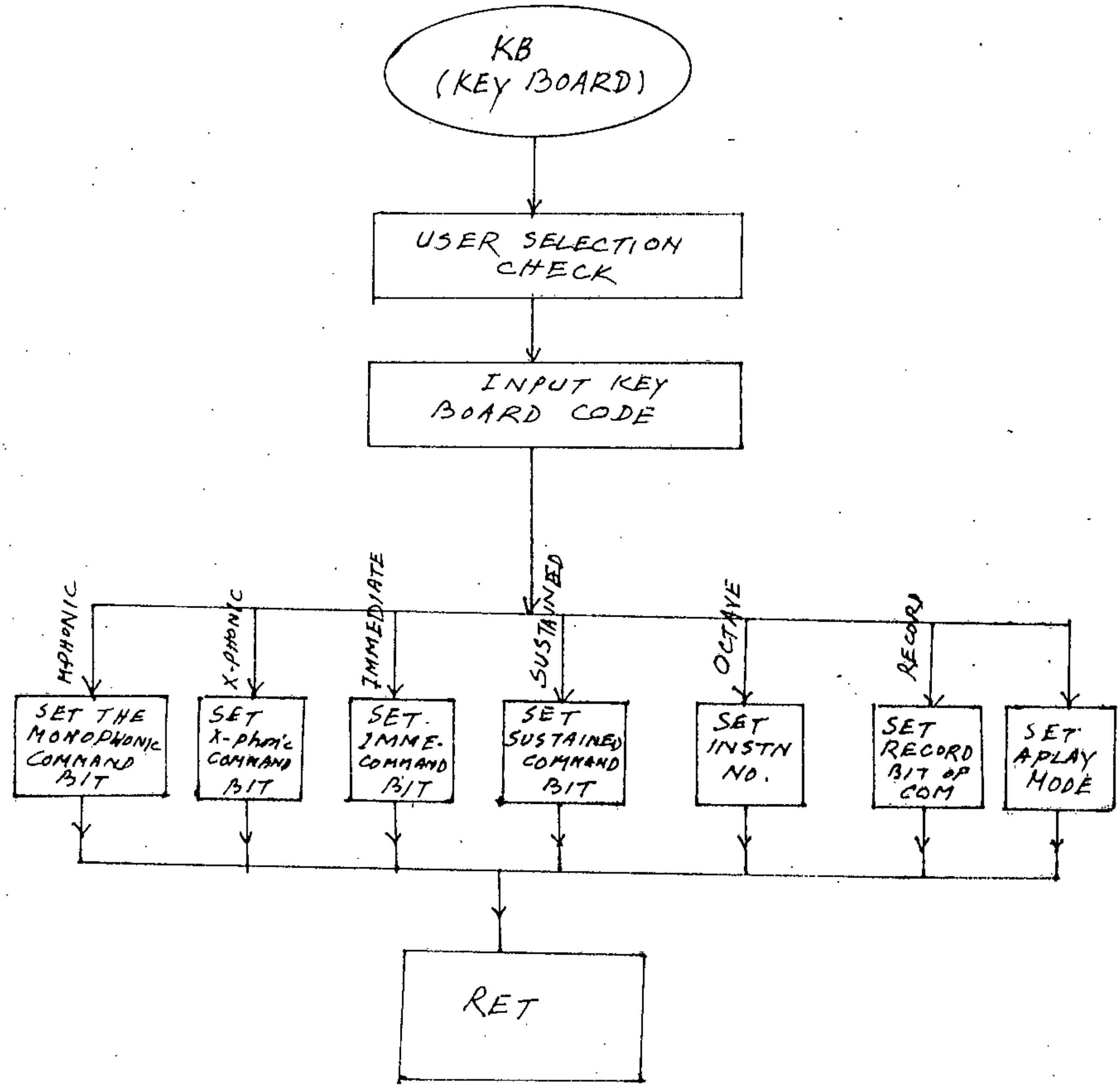
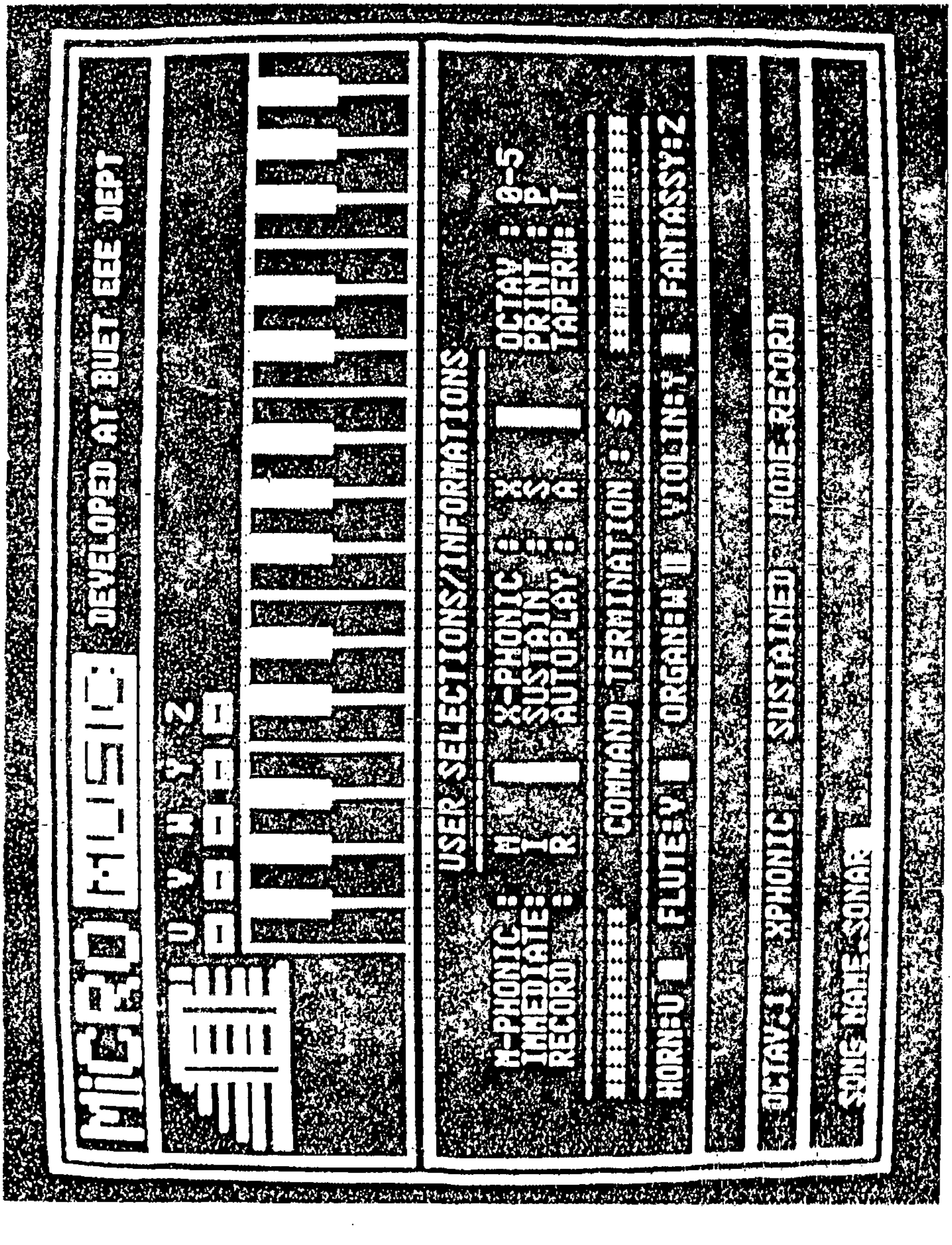


FIG. 5.16 KB FLOW CHART

5-28


MICRO MUSIC
DEVELOPED AT BUET EEE DEPT
U V W Y Z
USER SELECTIONS/INFORMATIONS
M-PHONIC: M
IMMEDIATE: I
RECORD : R
X-PHONIC : X
SUSTAIN : S
AUTOPLAY : A
OCTAV : 0-5
PRINT : P
TAPERW: T
COMMAND TERMINATION : S
HORN:U
FLUTE:V
ORGAN:W
VIOLIN:Y
FANTASSY:Z
OCTAV:3
XPHONIC
SUSTAINED
MODE:RECORD
SONG NAME:SONAR

CHAPTER - VI

ANALOG PROCESSING CIRCUIT

6.1. GENERATION OF ACTUAL WAVE SHAPES OF DIFFERENT INSTRUMENTS

We have observed the actual frequencies and wave shapes of different conventional musical instruments on CRO screen in the laboratory. In our system the actual wave shapes for the five instruments have been generated experimentally by adding different harmonics. For this purpose we have designed an analog processing circuit. The input to this circuit is the square wave output of the 14-bit programmable counter (8155) and the output is the desired wave shape. The output of the 14-bit programmable counter(8155) has been treated as the fundamental frequency and this fundamental frequency has been divided by a binary counter (SN74191) to get necessary harmonics. Operational amplifiers have been used to add the harmonics. Finally R-C filters have been employed for necessary wave shaping. The complete circuit diagram of the analog processing circuit is given in figure 6.1. Here an open collector hex-inverter has been used to select any one of the five instruments or any combination among them. A logic "1" to the input of the hex-inverter disables an instrument, whereas a zero to the input of the inverter enables an instrument. The wave shapes of different organs obtained from the CRO screen are shown in figure 6.2. The organ tone is obtained by adding the A and D output of the counter. The flute tone is obtained from C output of the counter. The fantasy tone is obtained from the A output of the counter. The violin tone is obtained by adding the clock input and D output of the counter. The horn tone is obtained by adding B, C and D output of the counter. We have used Op-amplifier, to add the harmonics, for each instrument. Finally the output of the five op-amplifiers have been mixed by a mixer amplifier. The output of the mixer amplifier has been connected to the input of a audio power amplifier.

6.2. AUDIO AMPLIFIER AND TONE CONTROL CIRCUIT

It is necessary to amplify the output signal of the analog processing circuit to drive the speaker. This has been done by using STK435 I.C. power amplifier. STK 435 is a stereo power

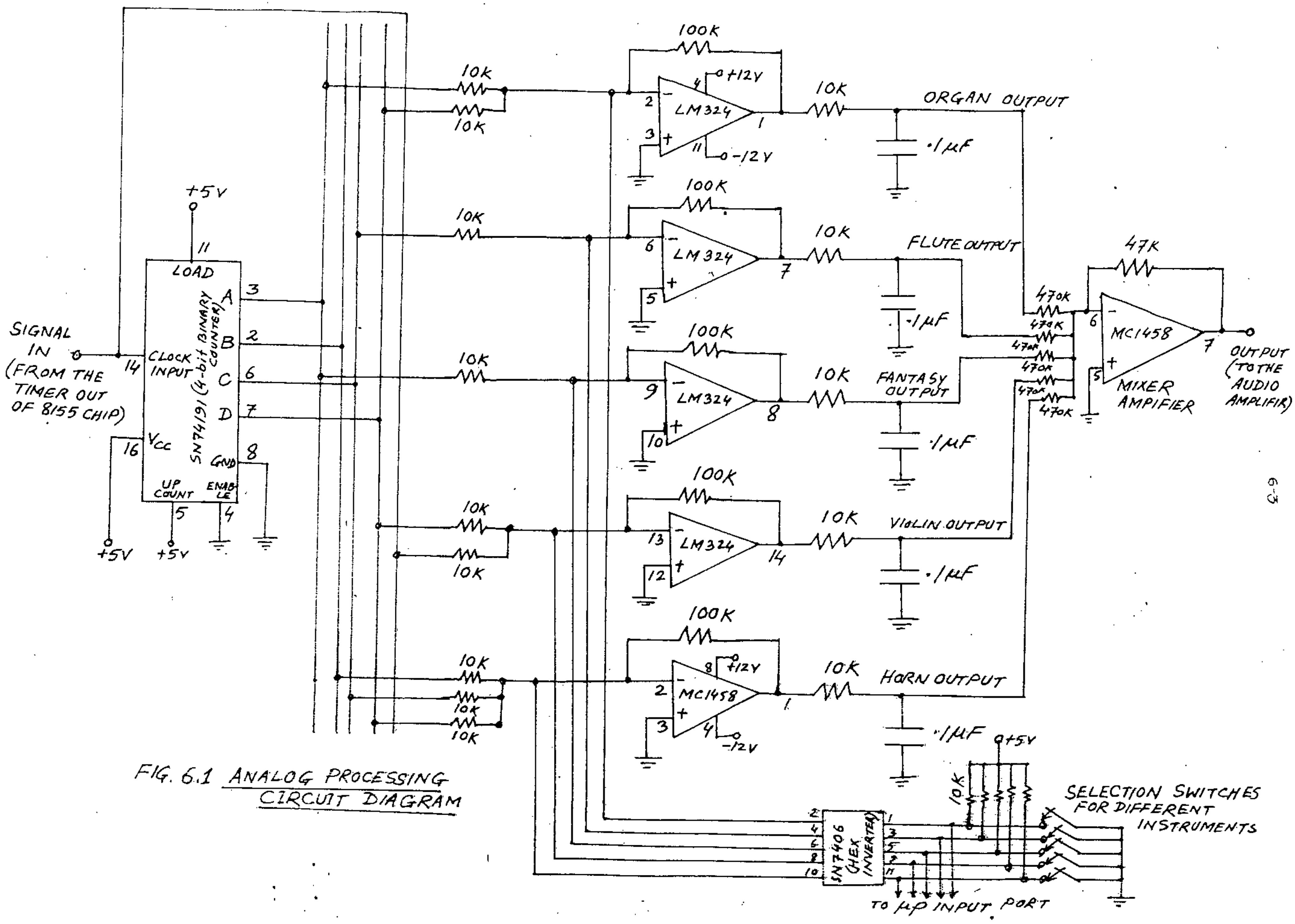


FIG. 6.1 ANALOG PROCESSING CIRCUIT DIAGRAM

amplifier, which gives an output power of 7 watts/channel into 8 ohms load. We have used a tone control circuit by using op-amplifier, which provides bass and treble controls. The tone control stage normally has a unity gain when the control potentiometers are centered. However, this gain is adjustable, with respect to frequency, if the control potentiometers are not centered. The output of the tone control stage directly drives the main power amplifier. The complete circuit diagram of the tone control stage and audio power amplifier is shown in figure 6.3.

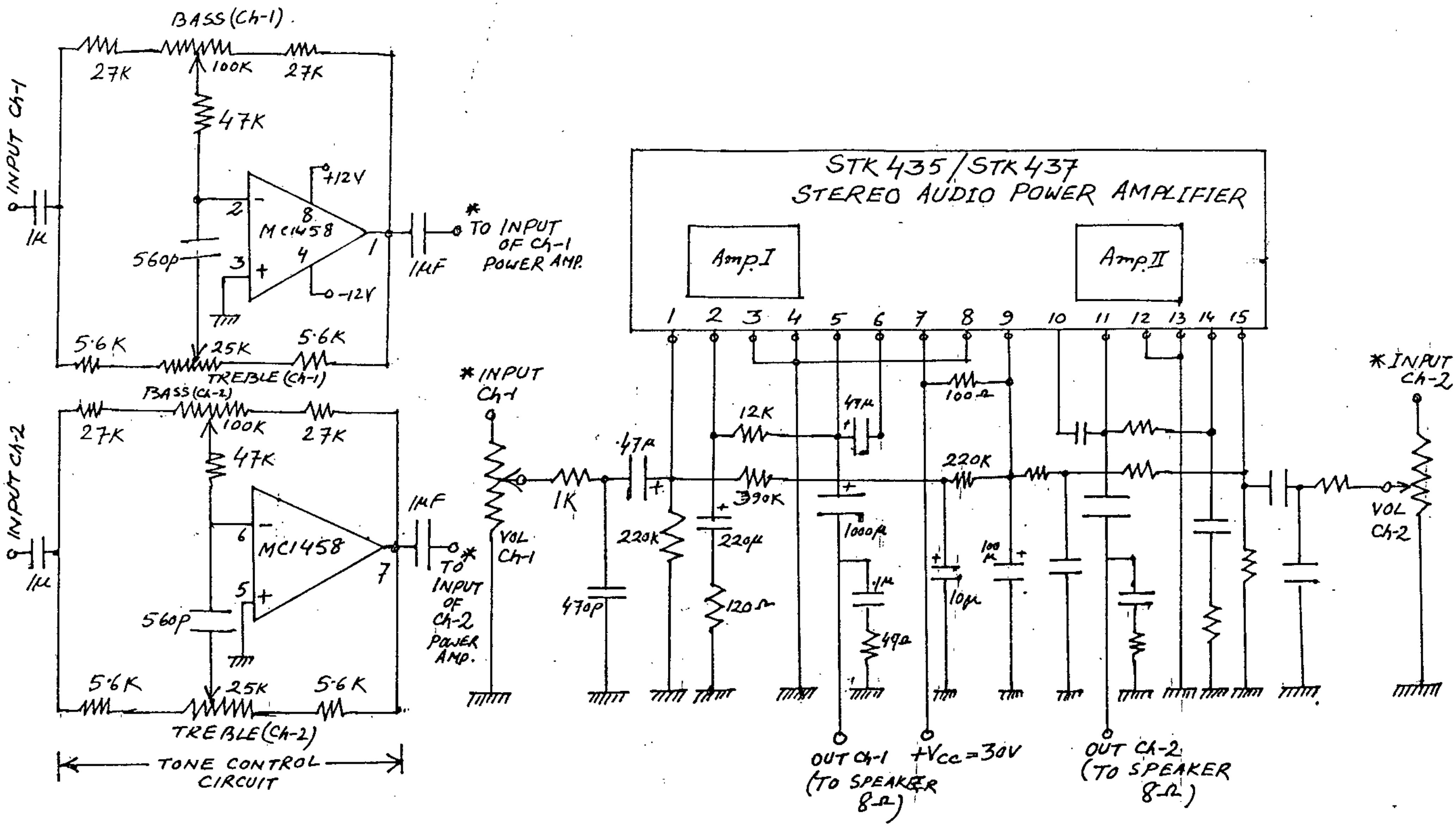
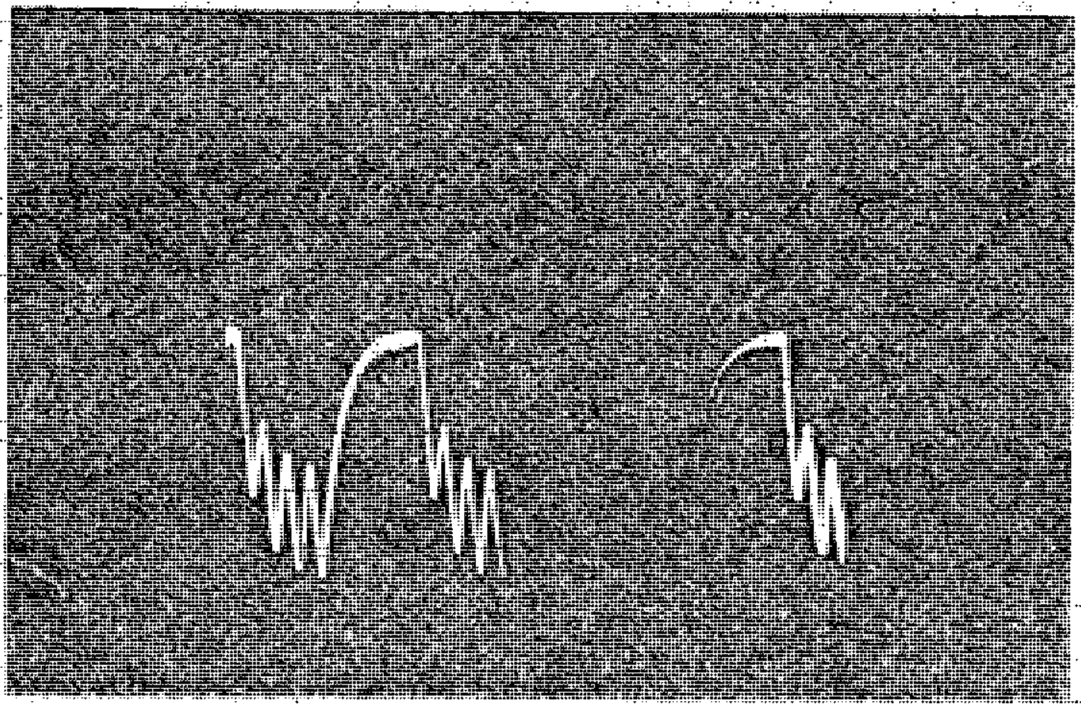
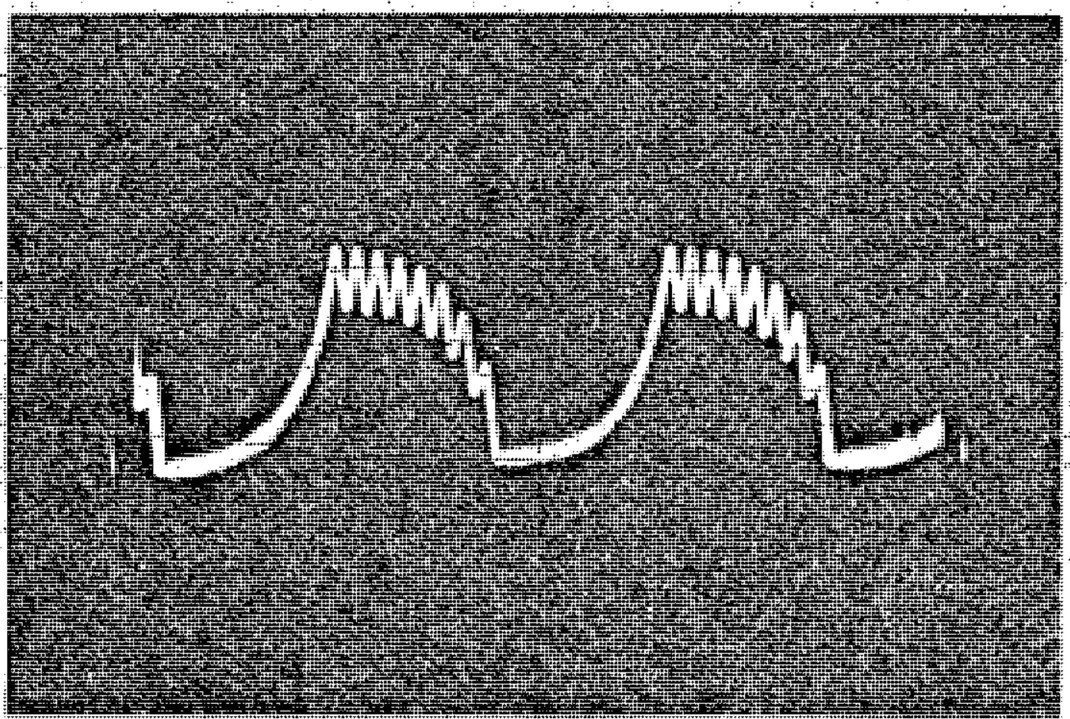


FIG. 6.3 AUDIO POWER AMPLIFIER AND TONE CONTROL CIRCUIT

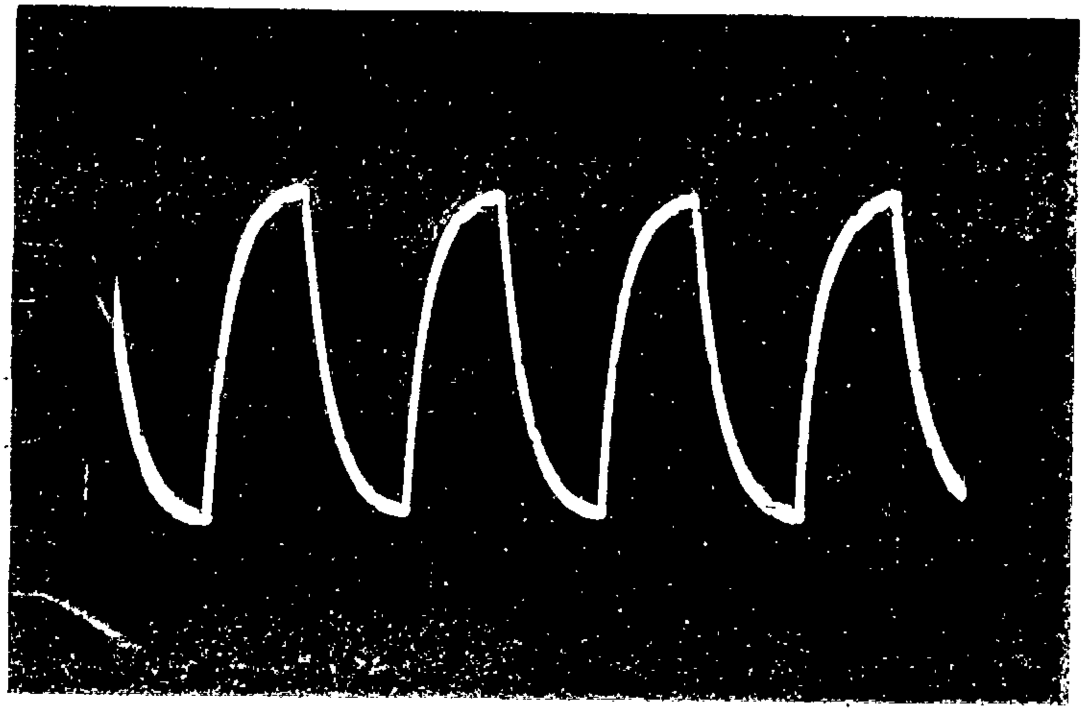
FIG. 6.2 WAVE SHAPES OF DIFFERENT INSTRUMENTS



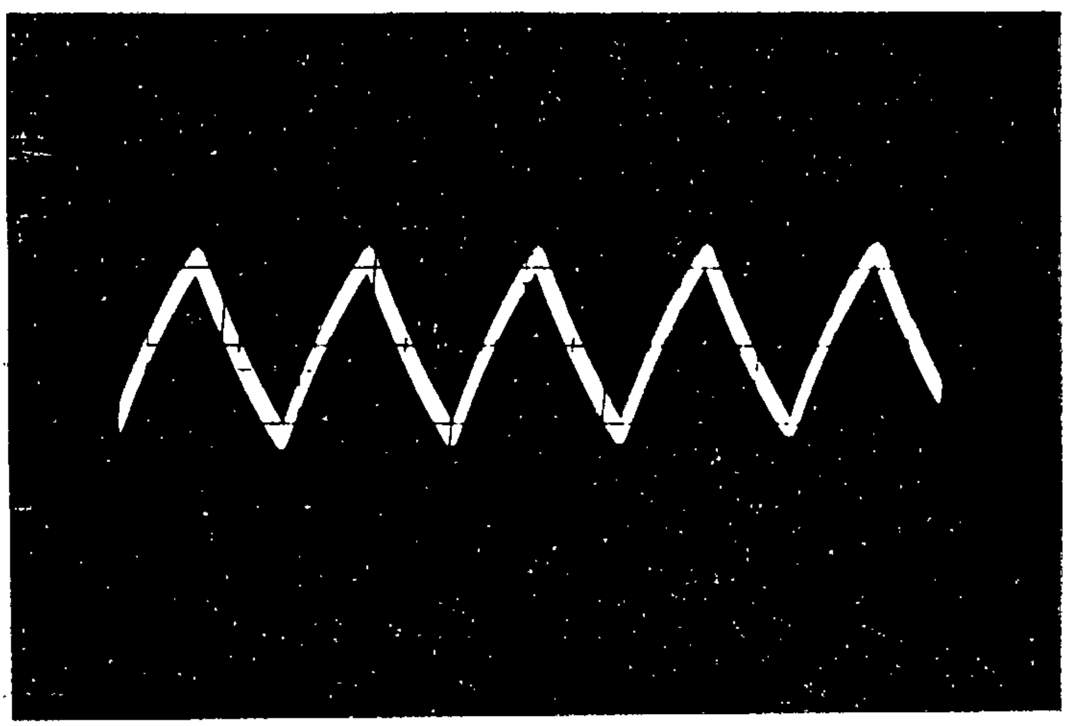
(A) ORGAN



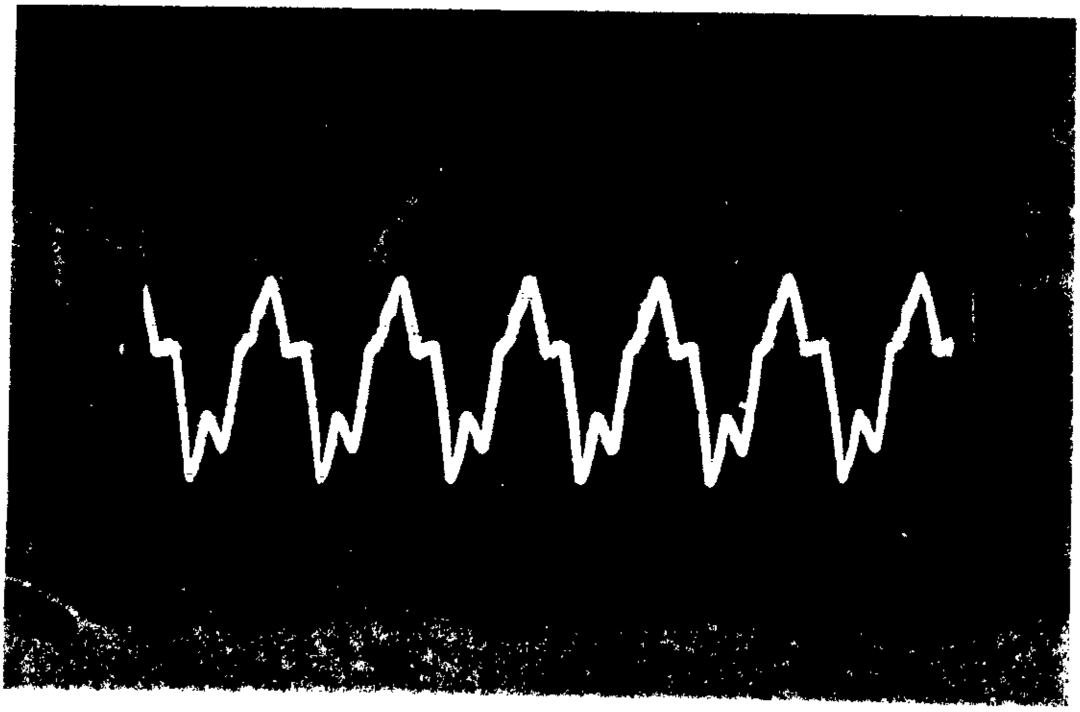
(B) VIOLIN



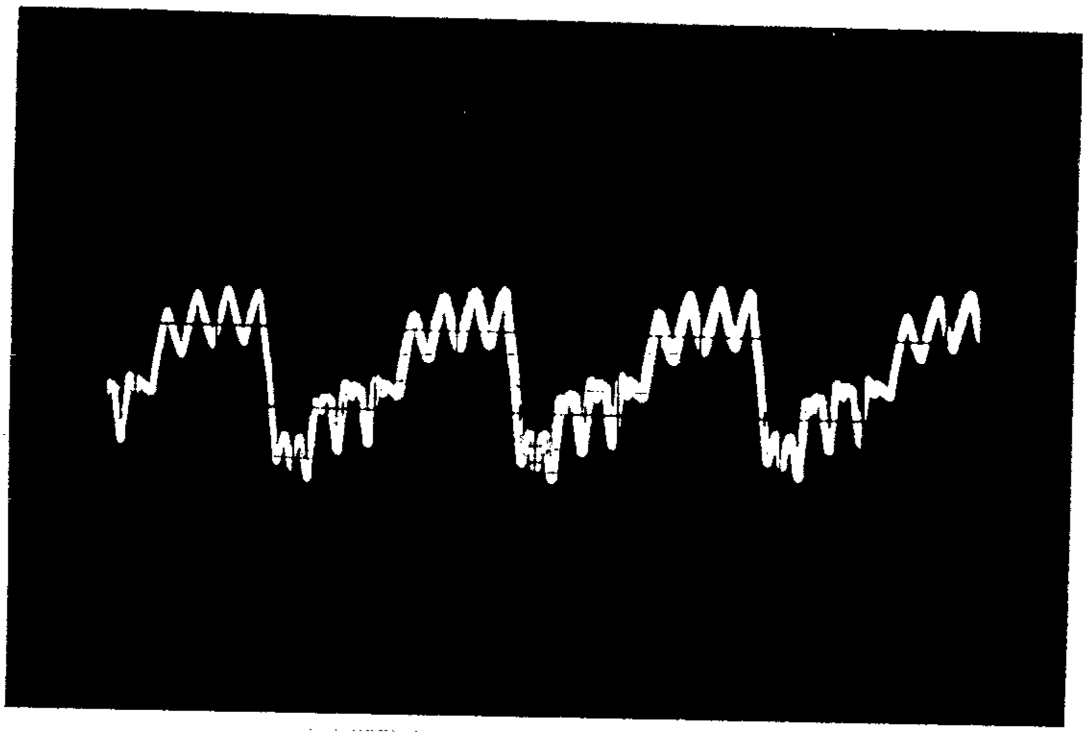
(C) FLUTE



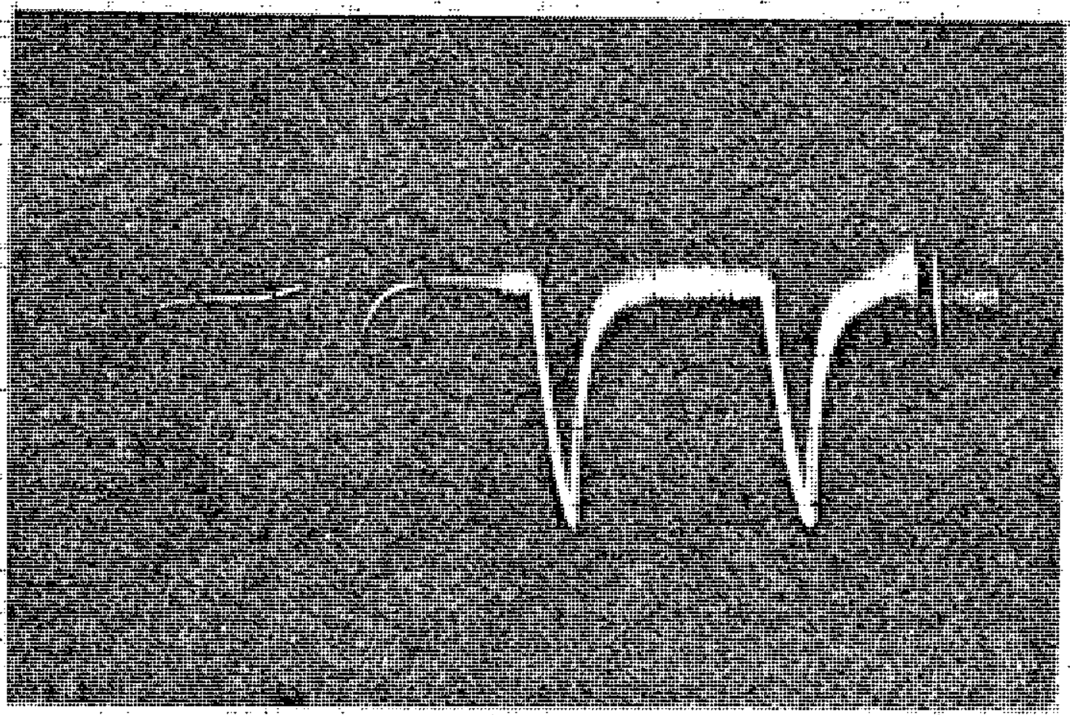
(D) FANTASY



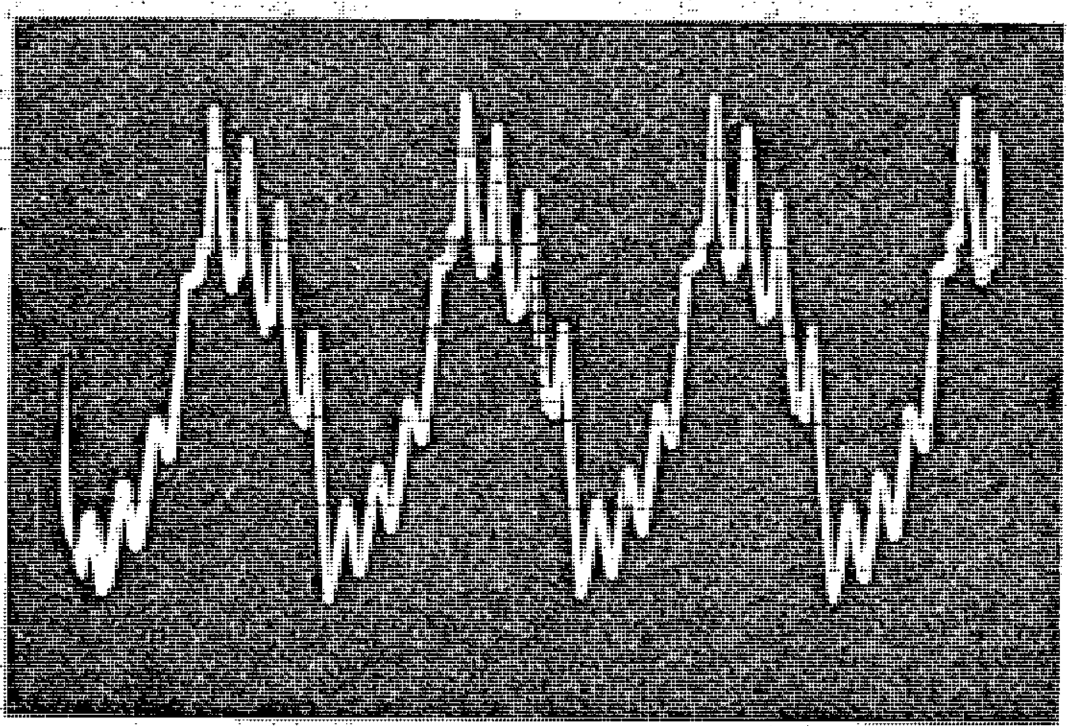
(E) ORGAN, FLUTE & HORN



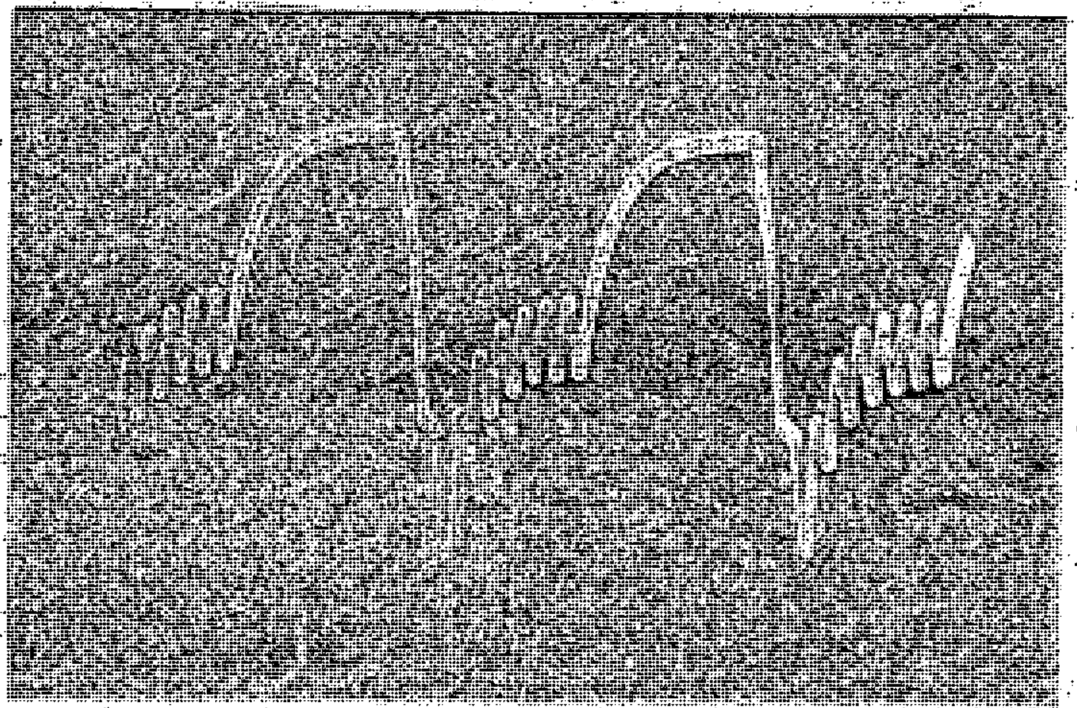
(F) ORGAN. FANTASY, VIOLIN, HORN



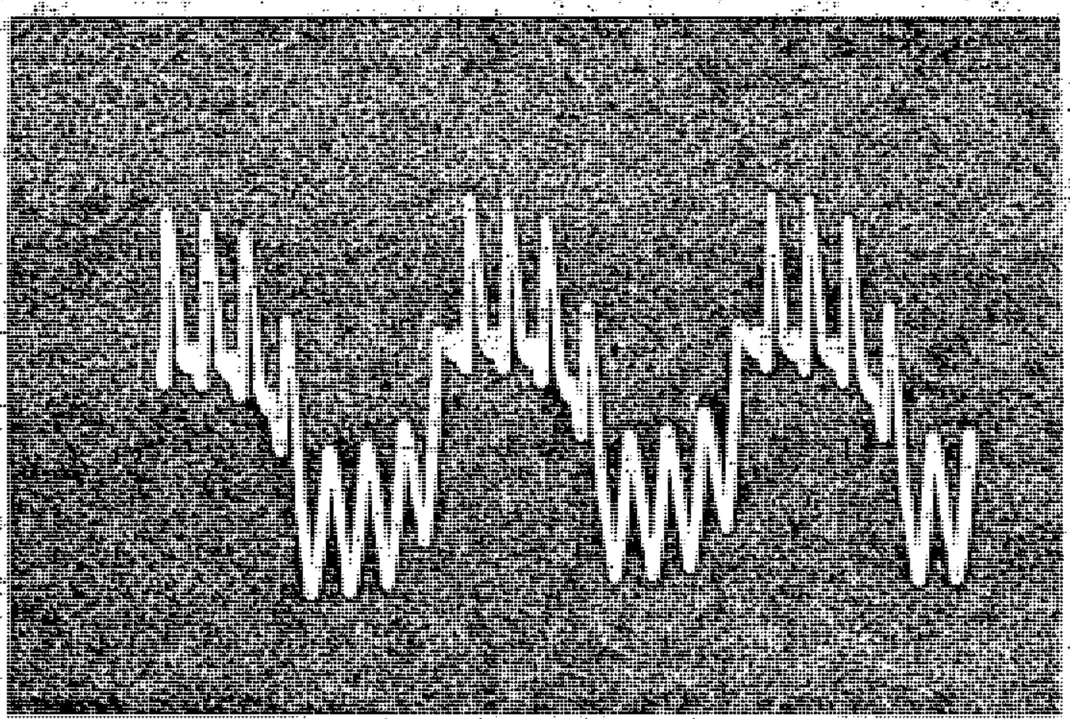
(G) HORN



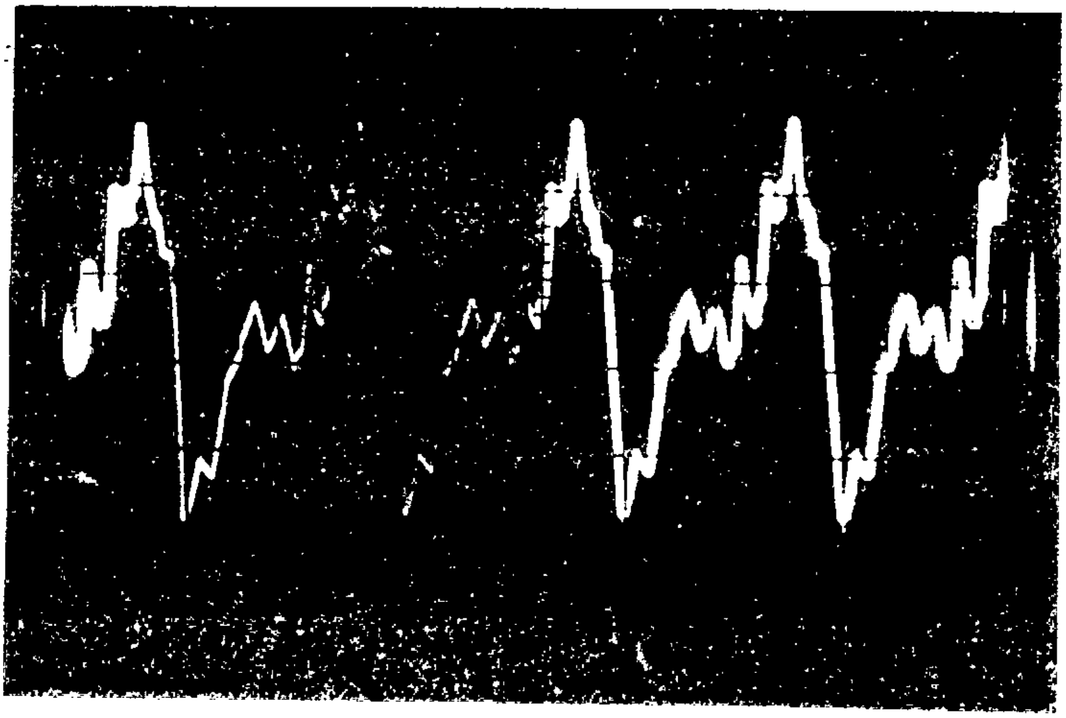
(H) FANTASY, VIOLIN, HORN



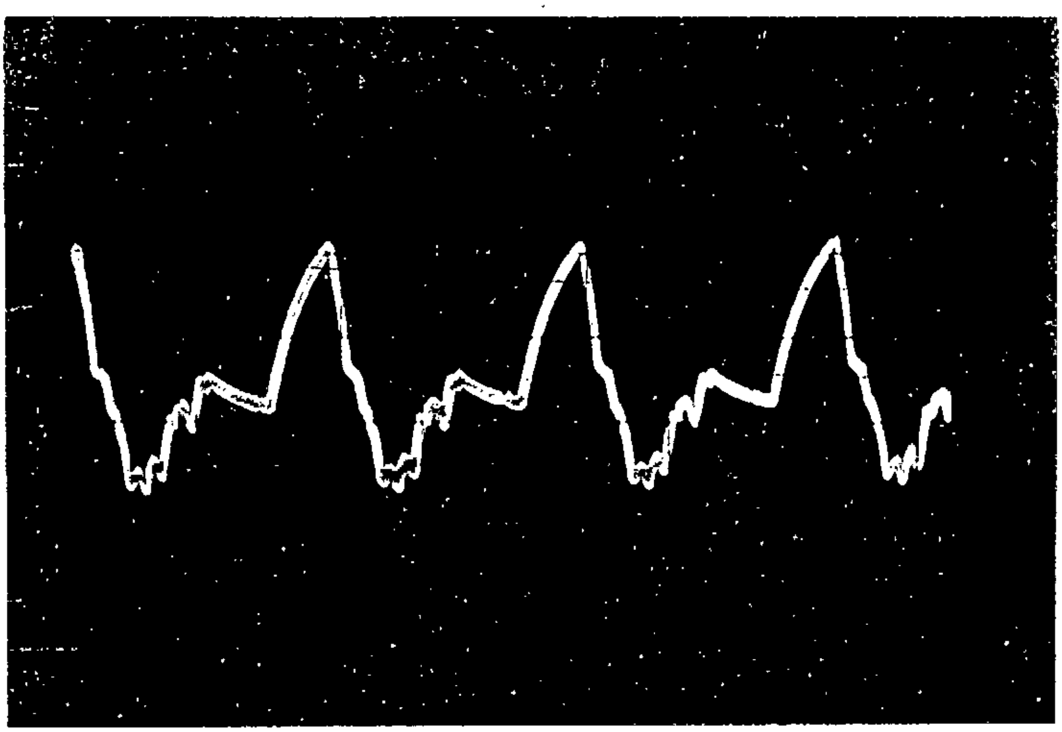
(I) HORN - VIOLIN



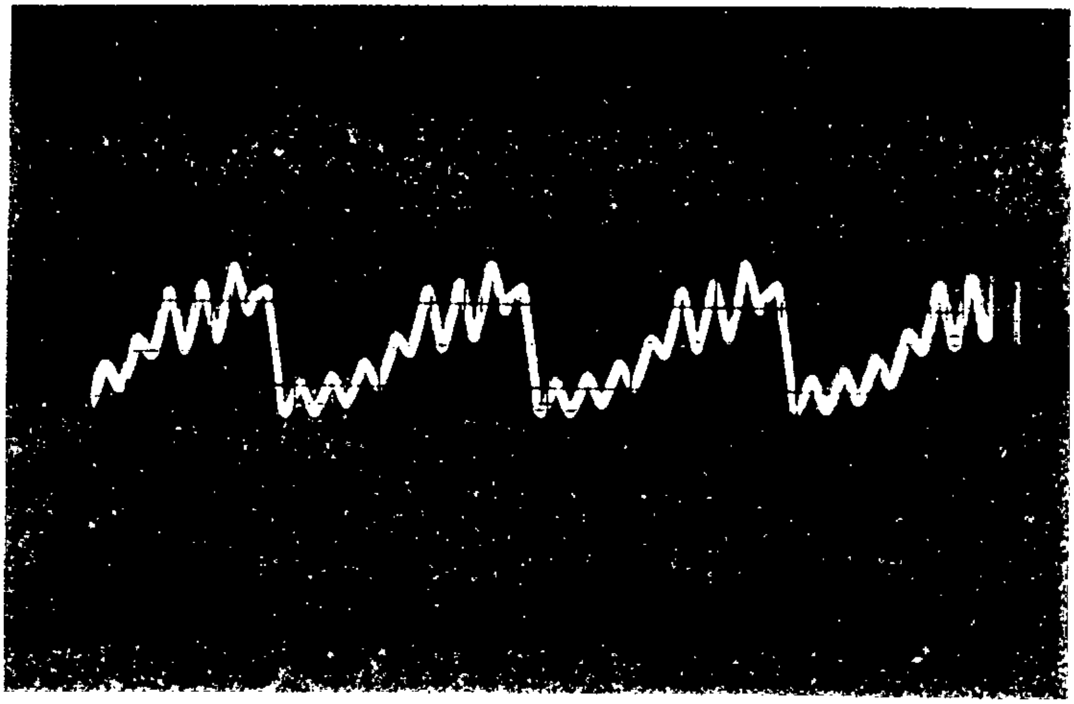
(J) FANTASY - VIOLIN



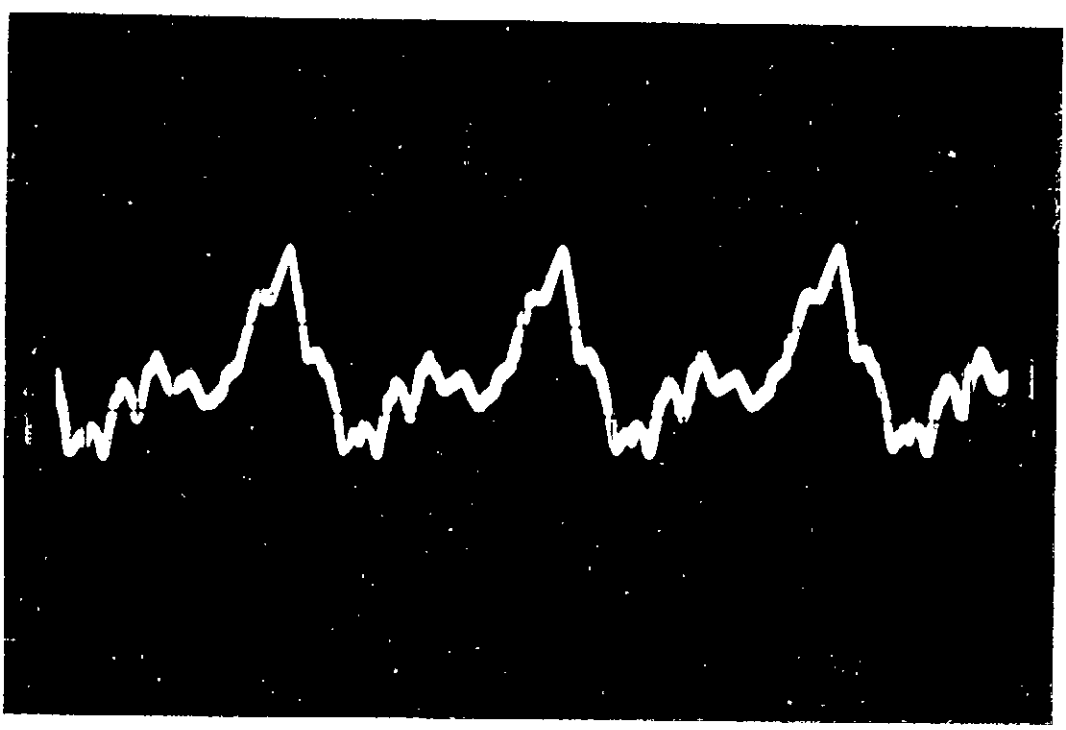
(K) ORGAN, FLUTE, VIOLIN, HORN & FANTASY



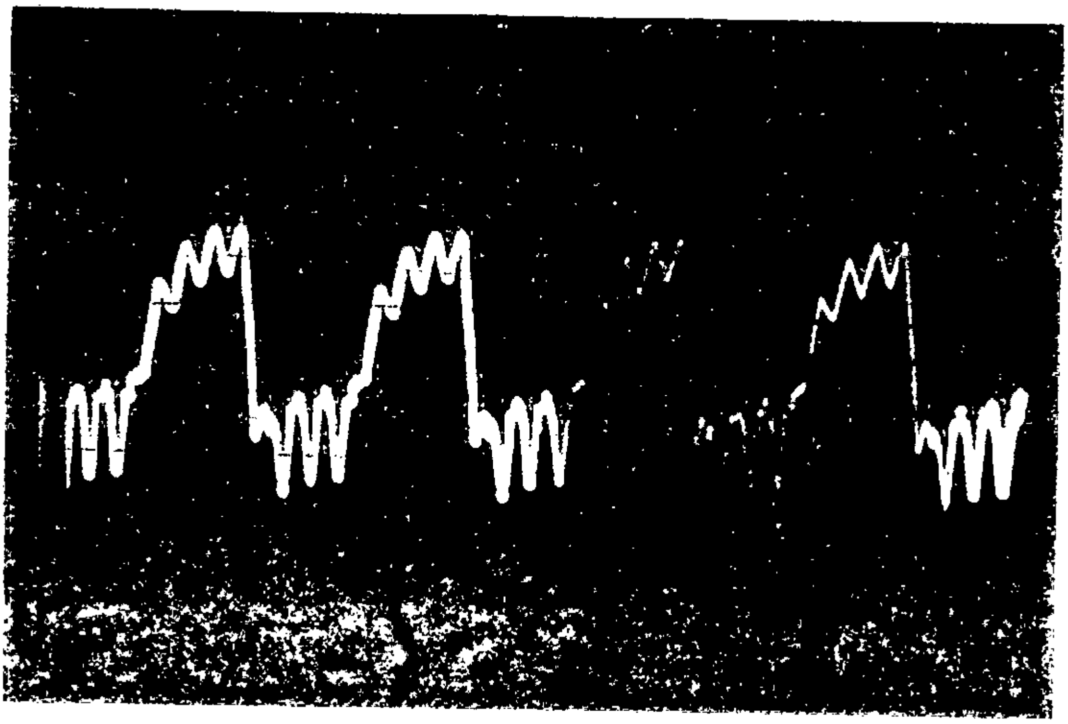
(L) ORGAN, FLUTE, VIOLIN & HORN



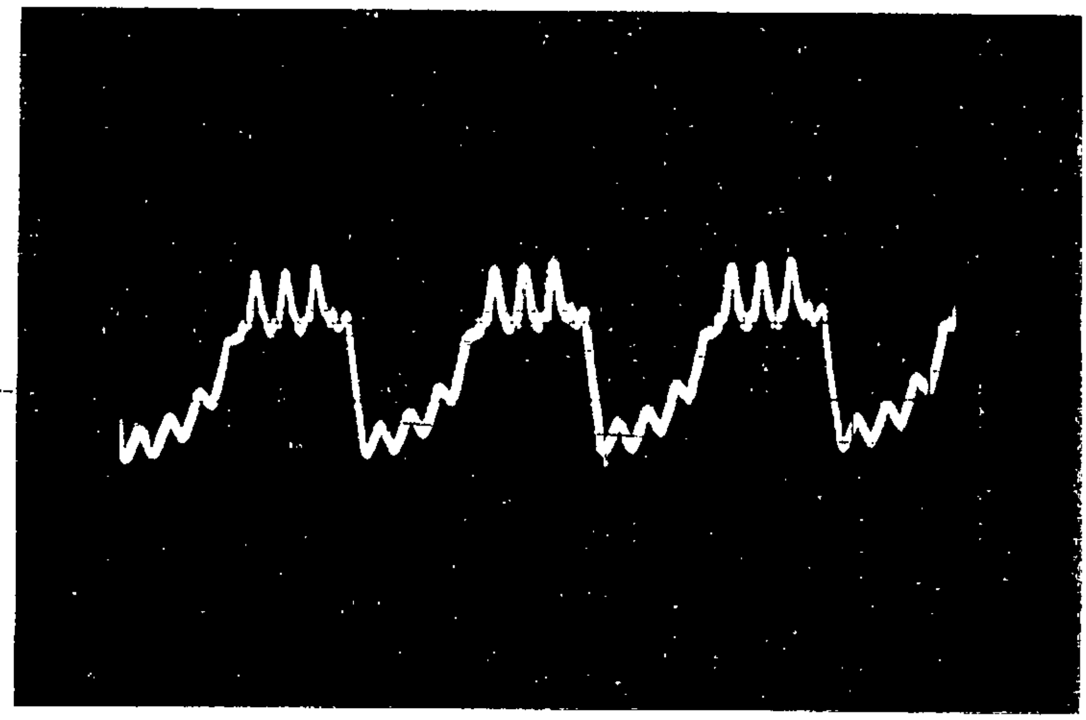
(M) FANTASY, HORN



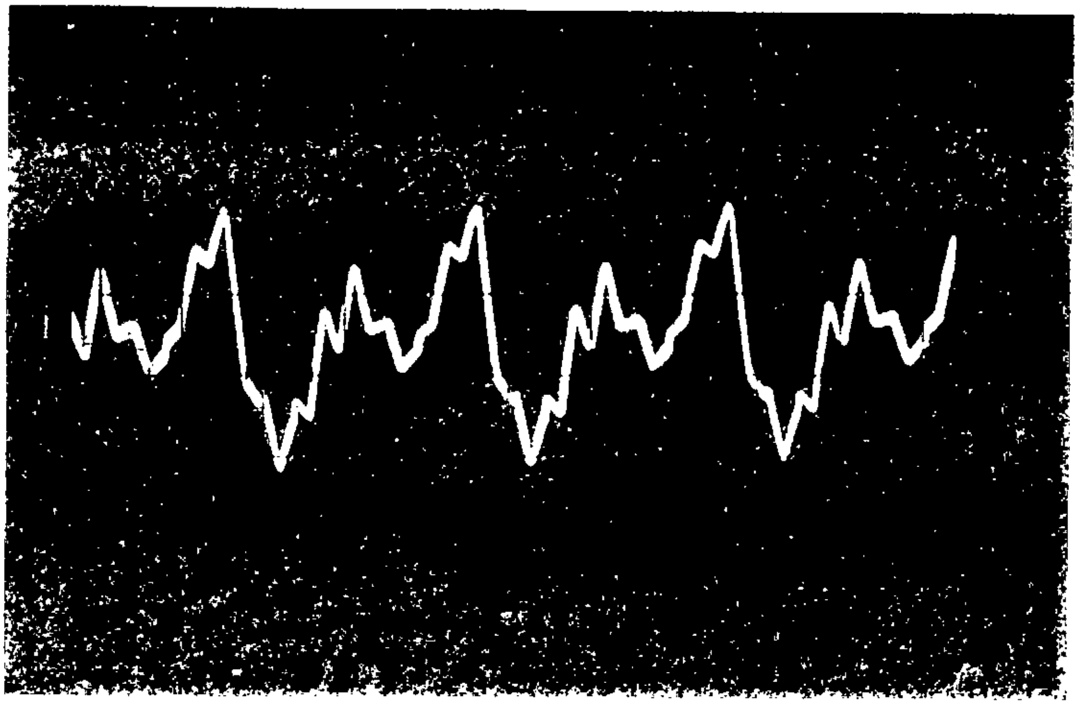
(N) FLUTE, FANTASY, HORN, VIOLIN



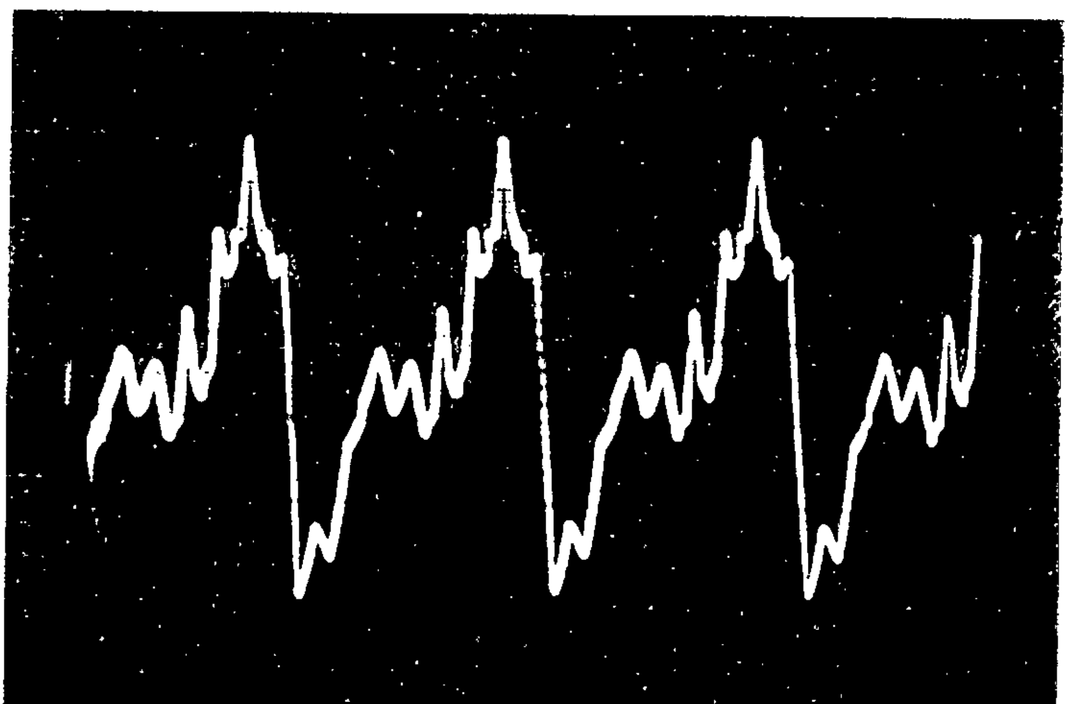
(O) ORGAN, FLUTE



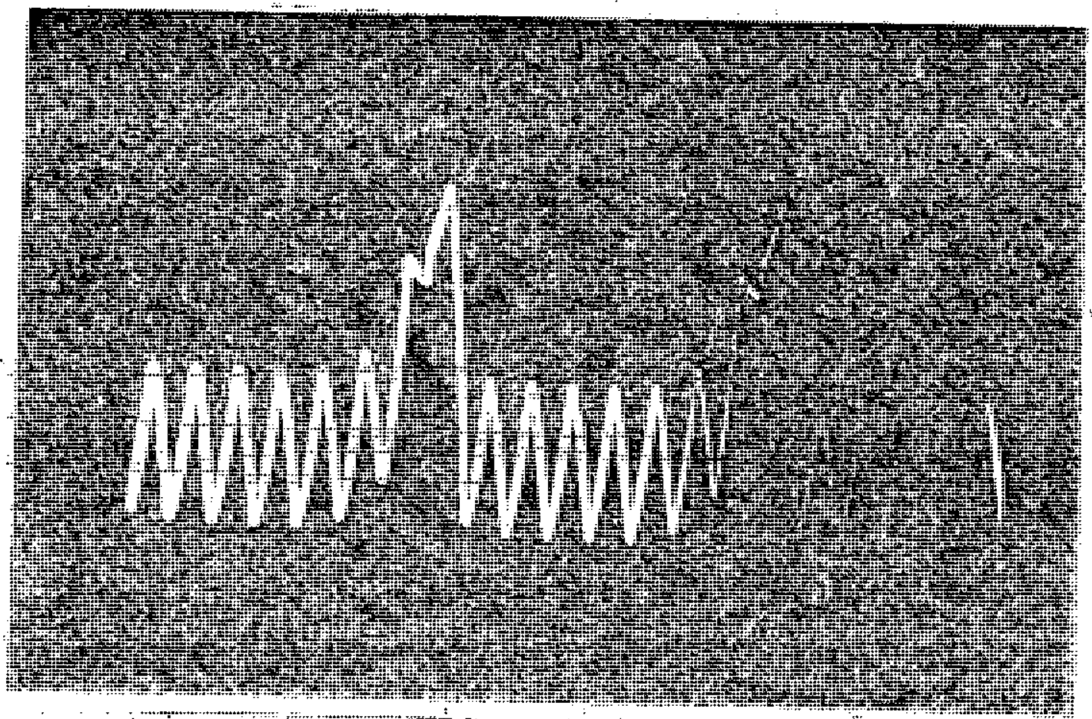
(P) ORGAN, FANTASY, VIOLIN, HORN



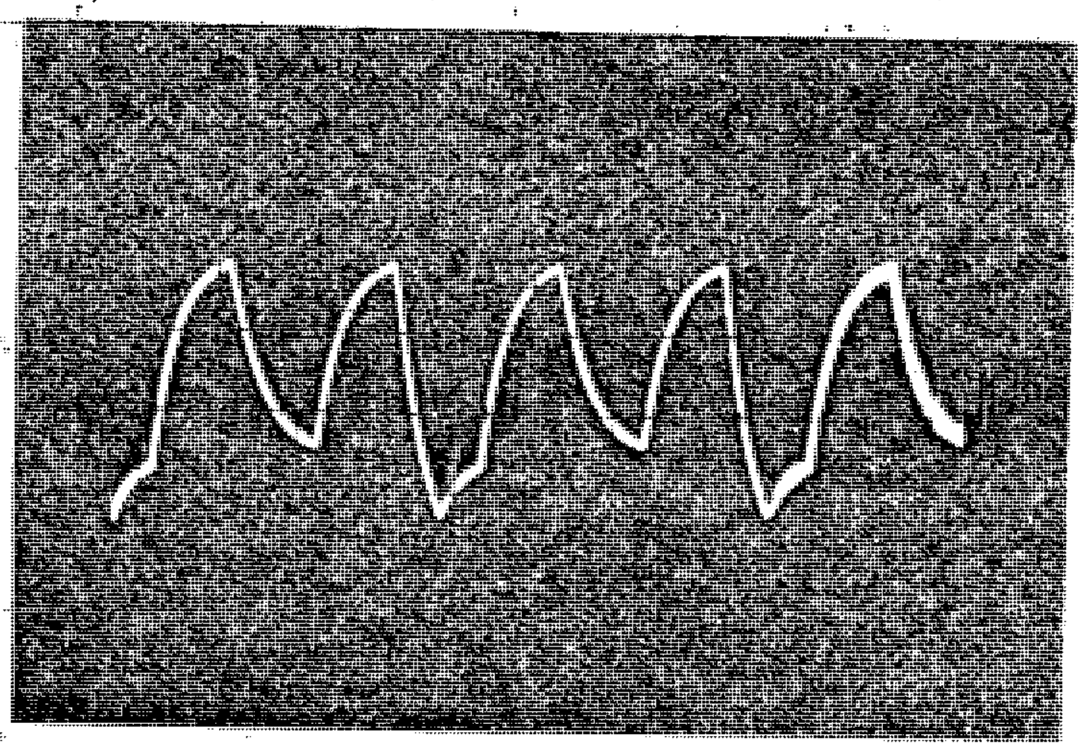
(Q) ORGAN, FLUTE, FANTASY, HORN



(R) FLUTE, FANTASY, VIOLIN, HORN



(S) FANTASY, HORN



(T) HORN, FLUTE

CHAPTER - VII

OPERATING INSTRUCTIONS

The users should know some necessary instructions and informations to play the developed musical instrument. These instructions and informations are given below. User's operating informations are also available on the display screen.

7.1. MANUAL PLAY

1. Press the character 'I' of key board for immediate stop operation OR the character 'S' for sustained stop operation.
2. Select the desired Octave by pressing the digit 0-5 of the key board.
3. Press the character 'M' of the key board for monophonic operation OR the character 'X' for X-phonic operation.
4. Select any instrument among the five - ORGAN, FLUTE,

FANTASY, VIOLIN, HORN by pressing the corresponding key on the musical key board. Any combination can be selected by pressing corresponding keys simultaneously.

7.2 RECORDING

1. Press 'R' of key-board to set the instrument in recording mode.

2. Give a song name of maximum8character long using the key board.

3. Press 'CR' of key board.

4. Now, start recording.

5. Recording mode can be terminated by pressing (SHIFT) (4)/($) when desired.

7.3. AUTOPLAY

1. Press 'A' of keyboard to set the instrument in autoplay mode.

2. Give any one of the song name among the recorded song by using the keyboard.

3. Press 'CR' of keyboard to start auto-playing.

4. Stop auto-playing by pressing SHIFT 4 ($).

7.4. SPECIFICATIONS OF THE DEVELOPED MICROPROCESSOR CONTROLLED MUSICAL INSTRUMENT

Number of Octaves	:	6
Number of keys	:	37 keys
Preset sounds	:	5 sounds ; Organ, Flute, Fantasy, Violin and Horn. Or any combination of them.
Memory play function	:	*Manual Memory, Storage Capacity = Max. 5 songs. * Autoplay.
Built in stereo audio power amplifier	:	* Maximum output power of 7 W per channel. * Bass and Treble controls.

CHAPTER --VIII

CONCLUSION

A practical microprocessor controlled musical instrument has been constructed in our laboratory by using Intel 8085 microprocessor. The complete hard-ware and software design for this musical instrument has been described in the previous chapters. This instrument can be compared with conventional musical instruments such as flute, violin, organ etc. It has got some distinctions and additional facilities over the conventional musical instruments.The developed musical instrument is able to generate tones of organ, flute, fantasy, violin and horn or tones of any composition of these five instruments. Because the developed musical system is controlled by soft-ware program, one can easily change any parameter of this system as desired. The developed musical instrument is able to record a song in its RAM, which is a distinct advantage over the conventional musical instrument. The recorded song can also be replayed when desired. The maximum memory capacity of this system is five songs with their names. The mamory capacity can be extended by introducing additional RAM chips. The developed musical system will be helpful for musicians and composers of contemporary music who are interested in the application of microprocessor technology to the arts. This system can be further developed by introducing additional hard-ware and soft-ware. The developed system has got tremendous flexibility over any conventional musical system because it is programmable.

REFERENCES

1. Bateman, Wayne, "Introduction to Computer Music", Wiley-Interscience publication, John Wiley and Sons, Inc., New-York, U.S.A.

2. Eargle, John, "Sound Recording", published by Von Nostrand Reinhold Company, New -York, U.S.A.

3. Ryerson, Michael, "Acoustic troubleshooting of audio system", Reston Publishing Company, Inc., Virginia, U.S.A.

4. Herrick, Clyde N., "Electronic Troubleshooting", Reston Publishing Company, INC, Virginia, U.S.A.

5. Intel MCS-80/85 Family User's Manual, October 1979, published by Intel Corporation.

APPENDIX A

COMPLETE SOFT-WARE PROGRAM FOR
THE DEVELOPED MUSICAL SYSTEM

COMPLETE SOFT-WARE PROGRAM

```
ASSM 1800

  1AA0                 0020 KB EQU 1AA0H
  F830                 0040 NMTBP EQU 0F830H
  F818                 0060 FRQTB EQU 0F818H
  F812                 0080 CNTRL EQU 0F812H
  F813                 0100 CNTRH EQU 0F813H
  F814                 0120 CKEN EQU 0F814H
  F815                 0140 DBCNT EQU 0F815H
  F816                 0160 PKEY EQU 0F816H
  F817                 0180 ZERO EQU 0F817H
  F810                 0200 MTAB EQU 0F810H
  F800                 0220 NTAB EQU 0F800H
  F806                 0240 ACTK EQU 0F806H
  F807                 0260 COM EQU 0F807H
  F808                 0280 INATB EQU 0F808H
  F80C                 0300 INTBE EQU 0F80CH
  F80D                 0320 INSTN EQU 0F80DH
  F80E                 0340 PRCNT EQU 0F80EH


  1800 3E10            0360 INIT MVI A,10H
  1802 3207F8          0380  STA COM
  1805 3E01            0400  MVI A,01
  1807 320DF8          0420  STA INSTN
  180A 1118F8          0440  LXI D,FRQTB
  180D 11341A          0460  LXI D,FRQT
  1810 CD8A1A          0480  CALL MOVN
  1813 3E02            0500  MVI A,2
  1815 D3C0            0520  OUT 0C0H
  1817 3EFF            0540  MVI A,0FFH
  1819 320CF8          0560  STA INTBE
  181C 3C              0580  INR A
  181D 3206F8          0600  STA ACTK
  1820 3230F8          0620  STA NMTBP
```

```
1823 CD3E18     Ø64Ø MAINP CALL REFR   MAIN  PROGRAM
1826 CD8C18     Ø66Ø  CALL INACT
1829 CDBF18     Ø68Ø  CALL ACTKC
182C CDF318     Ø7ØØ  CALL KEYP
182F CD4919     Ø72Ø  CALL FRKC
1832 CD6B19     Ø74Ø  CALL STOP
1835 CDAØ1A     Ø76Ø  CALL KB
1838 CD8Ø1A     Ø78Ø  CALL STIME
183B C32318     Ø8ØØ  JMP MAINP
183E C5         Ø82Ø REFR PUSH B   REFR  SUB-PROGRAM
183F E5         Ø84Ø  PUSH H
184Ø D5         Ø86Ø  PUSH D
1841 ØE8Ø       Ø88Ø  MVI C,8ØH
1843 21ØØF8     Ø9ØØ  LXI H,NTAB
1846 16Ø4       Ø92Ø  MVI D,4
1848 79         Ø94Ø REFR2 MOV A,C
1849 Ø7         Ø96Ø  RLC
184A 4F         Ø98Ø  MOV C,A
184B FE2Ø       1ØØØ  CPI 2ØH
184D CA7618     1Ø2Ø  JZ REFR1
185Ø 79         1Ø4Ø  MOV A,C
1851 D3C2       1Ø6Ø  OUT ØC2H
1853 3EØØ       1Ø8Ø  MVI A,Ø
1855 3214F8     11ØØ  STA CKEN
1858 DBC1       112Ø  IN ØC1H
185A CD7D18     114Ø REFR3 CALL ENCD
185D D24318     116Ø  JNC REFR2
186Ø 5F         118Ø  MOV E,A
1861 3A14F8     12ØØ  LDA CKEN
1864 8Ø         122Ø  ADD B
1865 47         124Ø  MOV B,A
1866 3C         126Ø  INR A
1867 3214F8     128Ø  STA CKEN
186A 79         13ØØ  MOV A,C
186B Ø7         132Ø  RLC
186C Ø7         134Ø  RLC
186D Ø7         136Ø  RLC
186E BØ         138Ø  ORA B
186F 77         14ØØ  MOV M,A
187Ø 23         142Ø  INX H
1871 7B         144Ø  MOV A,E
1872 15         146Ø  DCR D
1873 C25A18     148Ø  JNZ REFR3
1876 3EFF       15ØØ REFR1 MVI A,ØFFH
1878 77         152Ø  MOV M,A
1879 D1         154Ø  POP D
187A E1         156Ø  POP H
187B C1         158Ø  POP B
```

```
187C C9          1600  RET
187D FEFF        1620 ENCD CPI 0FFH
187F 37          1640  STC
1880 3F          1660  CMC
1881 C8          1680  RZ
1882 06FF        1700  MVI B,0FFH
1884 04          1720 DEC1 INR B
1885 37          1740  STC
1886 1F          1760  RAR
1887 DA8418      1780  JC DEC1
188A 37          1800  STC
188B C9          1820  RET
188C E5          1840 INACT PUSH H  INACTIVE  KEY  SUB-PROGRAM
188D D5          1860  PUSH D
188E 2108F8      1880  LXI H,INATB
1891 7E          1900 INAC2 MOV A,M
1892 FEFF        1920  CPI 0FFH
1894 CAAF18      1940  JZ SAVE
1897 1100F8      1960  LXI D,NTAB
189A CDB218      1980  CALL TABS
189D DAA718      2000  JC INAC1
18A0 3E00        2020  MVI A,0
18A2 77          2040  MOV M,A
18A3 23          2060  INX H
18A4 C39118      2080  JMP INAC2
18A7 1B          2100 INAC1 DCX D
18A8 3E00        2120  MVI A,0
18AA 12          2140  STAX D
18AB 23          2160  INX H
18AC C39118      2180  JMP INAC2
18AF D1          2200 SAVE POP D
18B0 E1          2220  POP H
18B1 C9          2240  RET
18B2 1A          2260 TABS LDAX D
18B3 FEFF        2280  CPI 0FFH
18B5 37          2300  STC
18B6 3F          2320  CMC
18B7 C8          2340  RZ
18B8 BE          2360  CMP M
18B9 13          2380  INX D
18BA C2B218      2400  JNZ TABS
18BD 37          2420  STC
18BE C9          2440  RET
18BF 3A06F8      2460 ACTKC LDA ACTK  ACTIVE  KEY  SUB-PROGRAM
18C2 FE00        2480  CPI 0
18C4 C8          2500  RZ
18C5 D5          2520  PUSH D
18C6 E5          2540  PUSH H
18C7 2106F8      2560  LXI H,ACTK
18CA 1100F8      2580  LXI D,NTAB
18CD CDB218      2600  CALL TABS
```

```
18D0 DAEC18    2620  JC ACTK1
18D3 CD5A1A    2640  CALL STRK
18D6 3E00      2660  MVI A,0
18D8 3206F8    2680  STA ACTK
18DB 3A07F8    2700  LDA COM
18DE F601      2720  ORI 1
18E0 3207F8    2740  STA COM
18E3 21FF00    2760  LXI H,0FFH
18E6 220EF8    2780  SHLD PRCNT
18E9 E1        2800  POP H
18EA D1        2820  POP D
18EB C9        2840  RET
18EC 1B        2860 ACTK1 DCX D
18ED 3E00      2880  MVI A,0
18EF 12        2900  STAX D
18F0 E1        2920  POP H
18F1 D1        2940  POP D
18F2 C9        2960  RET
18F3 3A16F8    2980 KEYP LDA PKEY KEY ON PROCESS SUB-PROGRAM
18F6 FE00      3000  CPI 0
18F8 C8        3020  RZ
18F9 D5        3040  PUSH D
18FA E5        3060  PUSH H
18FB 2116F8    3080  LXI H,PKEY
18FE 1100F8    3100  LXI D,NTAB
1901 CDB218    3120  CALL TABS
1904 D24119    3140  JNC KEYP1
1907 1B        3160  DCX D
1908 3E00      3180  MVI A,0
190A 12        3200  STAX D
190B 3A15F8    3220  LDA DBCNT
190E 3C        3240  INR A
190F 3215F8    3260  STA DBCNT
1912 FEFF      3280  CPI 0FFH
1914 D24619    3300  JNC KEYP3
1917 CD5A1A    3320  CALL STRK
191A 3E00      3340  MVI A,0
191C 3217F8    3360  STA ZERO
191F 2117F8    3380  LXI H,ZERO
1922 1108F8    3400  LXI D,INATB
1925 CDB218    3420  CALL TABS
1928 D23019    3440  JNC KEYP2
192B 1B        3460  DCX D
192C 3A06F8    3480  LDA ACTK
192F 12        3500  STAX D
1930 3A16F8    3520 KEYP2 LDA PKEY
1933 3206F8    3540  STA ACTK
1936 CD9619    3560  CALL SEND
1939 3A07F8    3580  LDA COM
193C E6FE      3600  ANI 0FEH
193E 3207F8    3620  STA COM
```

```
1941 3E00      3640 KEYP1 MVI A,0
1943 3216F8    3660  STA PKEY
1946 E1        3680 KEYP3 POP H
1947 D1        3700  POP D
1948 C9        3720  RET
1949 E5        3740 FRKC PUSH H   FRESH KEY SUB-PROGRAM
194A 2100F8    3760  LXI H,NTAB
194D 3EFF      3780 FRKC1 MVI A,0FFH
194F BE        3800  CMP M
1950 CA6919    3820  JZ FRKC2
1953 3E00      3840  MVI A,0
1955 BE        3860  CMP M
1956 23        3880  INX H
1957 CA4D19    3900  JZ FRKC1
195A 2B        3920  DCX H
195B 7E        3940  MOV A,M
195C 3216F8    3960  STA PKEY
195F 3E00      3980  MVI A,0
1961 77        4000  MOV M,A
1962 3215F8    4020  STA DBCNT
1965 23        4040  INX H
1966 C34D19    4060  JMP FRKC1
1969 E1        4080 FRKC2 POP H
196A C9        4100  RET
196B 3A07F8    4120 STOP LDA COM   STOP SUB-PROGRAM
196E 1F        4140  RAR
196F D0        4160  RNC
1970 E608      4180  ANI 8
1972 CA8119    4200  JZ STP1
1975 E5        4220  PUSH H
1976 2A0EF8    4240  LHLD PRCNT
1979 2B        4260  DCX H
197A 220EF8    4280  SHLD PRCNT
197D 7C        4300  MOV A,H
197E B5        4320  ORA L
197F E1        4340  POP H
1980 C0        4360  RNZ
1981 3E82      4380 STP1 MVI A,82H
1983 D3C0      4400  OUT 0C0H
1985 3A07F8    4420  LDA COM
1988 E6FE      4440  ANI 0FEH
198A 3207F8    4460  STA COM
198D 3E20      4480  MVI A,20H
198F 3279E9    4500  STA 0E979H
1992 327AE9    4520  STA 0E97AH
1995 C9        4540  RET
1996 E5        4560 SEND PUSH H   SEND SUB-PROGRAM
1997 C5        4580  PUSH B
1998 3A07F8    4600  LDA COM
199B E6FE      4620  ANI 0FEH
```

```
199D 3207F8    STA COM
19A0 CDDA19    CALL STAB
19A3 2B        DCX H
19A4 7E        MOV A,M
19A5 47        MOV B,A
19A6 E60F      ANI 0FH
19A8 87        ADD A
19A9 2118F8    LXI H,FRQTB
19AC 85        ADD L
19AD 6F        MOV L,A
19AE D2B219    JNC SEND1
19B1 24        INR H
19B2 4E       SEND1 MOV C,M
19B3 23        INX H
19B4 66        MOV H,M
19B5 69        MOV L,C
19B6 78        MOV A,B
19B7 E6F0      ANI 0F0H
19B9 0F        RRC
19BA 0F        RRC
19BB 0F        RRC
19BC 0F        RRC
19BD 4F        MOV C,A
19BE 3A0DF8    LDA INSTN
19C1 81        ADD C
19C2 4F        MOV C,A
19C3 0D       AGAIN DCR C
19C4 CACB19    JZ SEND2
19C7 29        DAD H
19C8 C3C319    JMP AGAIN
19CB 7D       SEND2 MOV A,L
19CC D3C4      OUT 0C4H
19CE 7C        MOV A,H
19CF F640      ORI 40H
19D1 D3C5      OUT 0C5H
19D3 3EC2      MVI A,0C2H
19D5 D3C0      OUT 0C0H
19D7 C1        POP B
19D8 E1        POP H
19D9 C9        RET
19DA 21EA19   STAB LXI H,KEYTB
19DD 3A06F8    LDA ACTK
19E0 CD4D1A    CALL DPK
19E3 23       STAB1 INX H
19E4 BE        CMP M
19E5 C8        RZ
19E6 23        INX H
19E7 C3E319    JMP STAB1
19EA 1025     KEYTB DW 2510H
19EC 1124      DW 2411H
19EE 1286      DW 8612H
```

```
19F0 1323        5660  DW 2313H
19F2 1485        5680  DW 8514H
19F4 1522        5700  DW 2215H
19F6 1684        5720  DW 8416H
19F8 1721        5740  DW 2117H
19FA 1820        5760  DW 2018H
19FC 1983        5780  DW 8319H
19FE 1A17        5800  DW 171AH
1A00 1B82        5820  DW 821BH
1A02 2016        5840  DW 1620H
1A04 2115        5860  DW 1521H
1A06 2281        5880  DW 8122H
1A08 2314        5900  DW 1423H
1A0A 2480        5920  DW 8024H
1A0C 2513        5940  DW 1325H
1A0E 2647        5960  DW 4726H
1A10 2712        5980  DW 1227H
1A12 2811        6000  DW 1128H
1A14 2946        6020  DW 4629H
1A16 2A10        6040  DW 102AH
1A18 2B45        6060  DW 452BH
1A1A 300F        6080  DW 0F30H
1A1C 310E        6100  DW 0E31H
1A1E 3244        6120  DW 4432H
1A20 330D        6140  DW 0D33H
1A22 3443        6160  DW 4334H
1A24 350C        6180  DW 0C35H
1A26 3642        6200  DW 4236H
1A28 370B        6220  DW 0B37H
1A2A 380A        6240  DW 0A38H
1A2C 3941        6260  DW 4139H
1A2E 3A09        6280  DW 093AH
1A30 3B40        6300  DW 403BH
1A32 4008        6320  DW 0840H
1A34 8700        6340 FRQT DW 0087H
1A36 8E00        6360  DW 008EH
1A38 9700        6380  DW 0097H
1A3A A000        6400  DW 00A0H
1A3C A900        6420  DW 00A9H
1A3E B300        6440  DW 00B3H
1A40 BE00        6460  DW 00BEH
1A42 CA00        6480  DW 00CAH
1A44 D500        6500  DW 00D5H
1A46 E200        6520  DW 00E2H
1A48 EF00        6540  DW 00EFH
1A4A FD00        6560  DW 00FDH
1A4C FF          6580  DB 0FFH
1A4D F5          6600 DPK PUSH P
1A4E E5          6620  PUSH H
1A4F C5          6640  PUSH B
```

```
1A50 2179E9    6660   LXI H,0E979H
1A53 CD8902    6680   CALL 289H
1A56 C1        6700   POP B
1A57 E1        6720   POP H
1A58 F1        6740   POP P
1A59 C9        6760   RET
1A5A 3A07F8    6780  STRK LDA COM   STORE  KEY  SUB-PROGRAM
1A5D E604      6800   ANI 4
1A5F C8        6820   RZ
1A60 E5        6840   PUSH H
1A61 2A10F8    6860   LHLD MTAB
1A64 3A06F8    6880   LDA ACTK
1A67 77        6900   MOV M,A
1A68 23        6920   INX H
1A69 3A12F8    6940   LDA CNTRL
1A6C 77        6960   MOV M,A
1A6D 23        6980   INX H
1A6E 3A13F8    7000   LDA CNTRH
1A71 77        7020   MOV M,A
1A72 23        7040   INX H
1A73 36FF      7060   MVI M,0FFH
1A75 2210F8    7080   SHLD MTAB
1A78 210000    7100   LXI H,0
1A7B 2212F8    7120   SHLD CNTRL
1A7E E1        7140   POP H
1A7F C9        7160   RET
1A80 E5        7180  STIME PUSH H  STORE  TIME  SUB-PROGRAM
1A81 2A12F8    7200   LHLD CNTRL
1A84 23        7220   INX H
1A85 2212F8    7240   SHLD CNTRL
1A88 E1        7260   POP H
1A89 C9        7280   RET
1A8A 1A        7300  MOVN LDAX D
1A8B FEFF      7320   CPI 0FFH
1A8D C8        7340   RZ
1A8E 77        7360   MOV M,A
1A8F 23        7380   INX H
1A90 13        7400   INX D
1A91 C38A1A    7420   JMP MOVN
```

```
ASSM 1AA0

1800             0020 INIT EQU 1800H
1A8A             0040 MOVN EQU 1A8AH
25A7             0060 GRP EQU 25A7H
F87D             0080 INSTD EQU 0F87DH
F87A             0100 AUTOC EQU 0F87AH
F875             0120 ASPD EQU 0F875H
F874             0140 INSTS EQU 0F874H
1996             0160 SEND EQU 1996H
F866             0180 NMATB EQU 0F866H
F864             0200 PTAB EQU 0F864H
F830             0220 NMTBP EQU 0F830H
F831             0240 NMTAB EQU 0F831H
F810             0260 MTAB EQU 0F810H
F870             0280 PNTR EQU 0F870H
F872             0300 WORKA EQU 0F872H
027A             0320 SCRN EQU 27AH
F806             0340 ACTK EQU 0F806H
F807             0360 COM EQU 0F807H
F808             0380 INATB EQU 0F808H
F80C             0400 INTBE EQU 0F80CH
F80D             0420 INSTN EQU 0F80DH
F503             0440 COMSR EQU 0F503H
1AA0 CD7DF8      0460 KB CALL INSTD      KB SUB-PROGRAM
1AA3 CDEB19      0480  CALL KEYB
1AA6 D0          0500  RNC
1AA7 FE30        0520  CPI 30H
1AA9 D8          0540  RC
1AAA FE36        0560  CPI 36H
1AAC D2CB1A      0580  JNC KB1
1AAF E60F        0600  ANI 0FH
1AB1 320DF8      0620  STA INSTN
1AB4 21C4EE      0640 OCT LXI H,0EEC4H
1AB7 11C41A      0660  LXI D,OCTAV
1ABA CD8A1A      0680  CALL MOVN
1ABD 3A0DF8      0700  LDA INSTN
1AC0 F630        0720  ORI 30H
1AC2 77          0740  MOV M,A
1AC3 C9          0760  RET
1AC4 0F03        0780 OCTAV DW 030FH
1AC6 1401        0800  DW 0114H
1AC8 162E        0820  DW 2E16H
1ACA FF          0840  DB 0FFH
1ACB E5          0860 KB1 PUSH H
1ACC C5          0880  PUSH B
1ACD 215B1C      0900  LXI H,COMTB
1AD0 47          0920  MOV B,A
1AD1 CD03F5      0940  CALL COMSR
1AD4 DADA1A      0960  JC FOUND
1AD7 C1          0980  POP B
```

```
1AD8 E1          POP H
1AD9 C9          RET
1ADA E9         FOUND PCHL
1ADB DBF1       KCALL IN 0F1H
1ADD E607        ANI 7
1ADF C8          RZ
1AE0 3E40        MVI A,40H
1AE2 D3F1        OUT 0F1H
1AE4 DBF0        IN 0F0H
1AE6 CD2BF6      CALL 0F62BH
1AE9 37          STC
1AEA C9          RET
1AEB CDDB1A     KEYB CALL KCALL
1AEE D0          RNC
1AEF FE24        CPI 24H
1AF1 C2151B      JNZ KEYB1
1AF4 3A07F8     SPLAY LDA COM
1AF7 E6FA        ANI 0FAH
1AF9 3207F8      STA COM
1AFC CDC31B     PLAY CALL MODE
1AFF 110A1B      LXI D,PL
1B02 CD8A1A      CALL MOVN
1B05 13          INX D
1B06 CD8A1A      CALL MOVN
1B09 C9          RET
1B0A 100C       PL DW 0C10H
1B0C 0119        DW 1901H
1B0E FF          DB 0FFH
1B0F 2020       SP DW 2020H
1B11 2020        DW 2020H
1B13 20FF        DW 0FF20H
1B15 37         KEYB1 STC
1B16 C9          RET
1B17 3A07F8     SUSTN LDA COM
1B1A F610        ORI 10H
1B1C 3207F8      STA COM
1B1F 21D9EE      LXI H,0EED9H
1B22 112B1B      LXI D,PROL
1B25 CD8A1A     MOVET CALL MOVN
1B28 C1         SAVE POP B
1B29 E1          POP H
1B2A C9          RET
1B2B 1315       PROL DW 1513H
1B2D 1314        DW 1413H
1B2F 0109        DW 0901H
1B31 0E05        DW 050EH
1B33 04FF        DW 0FF04H
1B35 3A07F8     IMMD LDA COM
1B38 E6EF        ANI 0EFH
1B3A 3207F8      STA COM
```

```
1B3D 21D9EE    LXI H,0EED9H
1B40 11461B    LXI D,IM
1B43 C3251B    JMP MOVET
1B46 090D     IM DW 0D09H
1B48 0D05      DW 050DH
1B4A 0409      DW 0904H
1B4C 0114      DW 1401H
1B4E 05FF      DW 0FF05H
1B50 3EFF     MPHON MVI A,0FFH
1B52 320CF8    STA INTBE
1B55 3C        INR A
1B56 3208F8    STA INATB
1B59 21CEEE    LXI H,0EECEH
1B5C 360D      MVI M,0DH
1B5E 11721B   MPH1 LXI D,MP
1B61 23        INX H
1B62 C3251B    JMP MOVET
1B65 3EFF     XPHON MVI A,0FFH
1B67 3208F8    STA INATB
1B6A 21CEEE    LXI H,0EECEH
1B6D 3618      MVI M,18H
1B6F C35E1B    JMP MPH1
1B72 1008     MP DW 0810H
1B74 0F0E      DW 0E0FH
1B76 0903      DW 0309H
1B78 FF        DB 0FFH
1B79 3A30F8   RECRD LDA NMTBP
1B7C FE05      CPI 5
1B7E CAA91B    JZ REC1
1B81 CD801C    CALL NREAD
1B84 CA281B    JZ SAVE
1B87 CDF81C    CALL NSTOR
1B8A CD731D    CALL NMDSP
1B8D CD2E1D    CALL ADSTR
1B90 2A10F8    LHLD MTAB
1B93 36FF      MVI M,0FFH
1B95 3A07F8   NONMR LDA COM
1B98 F604      ORI 4
1B9A 3207F8    STA COM
1B9D CDC31B    CALL MODE
1BA0 21EAEE    LXI H,0EEEAH
1BA3 11BA1B    LXI D,REC
1BA6 C3251B    JMP MOVET
1BA9 CDEF1C   REC1 CALL USRCL
1BAC 2184EF    LXI H,0EF84H
1BAF 11B51B    LXI D,FULL
1BB2 C3251B    JMP MOVET
1BB5 0615     FULL DW 1506H
1BB7 0C0C      DW 0C0CH
1BB9 FF        DB 0FFH
```

```
1BBA 1205      3000 REC DW 0512H
1BBC 030F      3020  DW 0F03H
1BBE 1204      3040  DW 0412H
1BC0 2020      3060  DW 2020H
1BC2 FF        3080  DB 0FFH
1BC3 21E5EE    3100 MODE LXI H,0EEE5H
1BC6 11CD1B    3120  LXI D,MO
1BC9 CD8A1A    3140  CALL MOVN
1BCC C9        3160  RET
1BCD 0D0F      3180 MO DW 0F0DH
1BCF 0405      3200  DW 0504H
1BD1 2EFF      3220  DW 0FF2EH
1BD3 D5        3240 APLAY PUSH D   AUTO PLAY SUB-PROGRAM
1BD4 CD801C    3260  CALL NREAD
1BD7 CA3C1C    3280  JZ NIL
1BDA CD521D    3300  CALL MATCH
1BDD D2471C    3320  JNC NFND
1BE0 2264F8    3340 NONA1 SHLD PTAB
1BE3 7E        3360  MOV A,M
1BE4 FEFF      3380  CPI 0FFH
1BE6 CA331C    3400  JZ APLA4
1BE9 3E60      3420  MVI A,60H
1BEB 3278E9    3440  STA 0E978H
1BEE 23        3460  INX H
1BEF 23        3480  INX H
1BF0 23        3500  INX H
1BF1 7E        3520 APLA1 MOV A,M
1BF2 FEFF      3540  CPI 0FFH
1BF4 CA331C    3560  JZ APLA4
1BF7 FE00      3580  CPI 0
1BF9 CA401C    3600  JZ NOKEY
1BFC E5        3620  PUSH H
1BFD CDDB1A    3640  CALL KCALL
1C00 E1        3660  POP H
1C01 D2091C    3680  JNC APLA3
1C04 FE24      3700  CPI 24H
1C06 CA331C    3720  JZ APLA4
1C09 7E        3740 APLA3 MOV A,M
1C0A 3206F8    3760  STA ACTK
1C0D CD9619    3780  CALL SEND
1C10 23        3800 APLA2 INX H
1C11 E5        3820  PUSH H
1C12 2A75F8    3840  LHLD ASPD
1C15 44        3860  MOV B,H
1C16 4D        3880  MOV C,L
1C17 E1        3900  POP H
1C18 5E        3920 DELY2 MOV E,M
1C19 23        3940  INX H
1C1A 56        3960  MOV D,M
1C1B 2B        3980  DCX H
```

```
1C1C 1B       DELY1 DCX D
1C1D 7B        MOV A,E
1C1E B2        ORA D
1C1F C22D1C    JNZ CHKEY
1C22 0B        DCX B
1C23 78        MOV A,B
1C24 B1        ORA C
1C25 C2181C    JNZ DELY2
1C28 23        INX H
1C29 23        INX H
1C2A C3F11B    JMP APLA1
1C2D CD7AF8   CHKEY CALL AUTOC
1C30 C31C1C    JMP DELY1
1C33 3E82     APLA4 MVI A,82H
1C35 D3C0      OUT 0C0H
1C37 3E20      MVI A,20H
1C39 3278E9    STA 0E978H
1C3C D1       NIL POP D
1C3D C1        POP B
1C3E E1        POP H
1C3F C9        RET
1C40 3E82     NOKEY MVI A,82H
1C42 D3C0      OUT 0C0H
1C44 C3101C    JMP APLA2
1C47 2184EF   NFND LXI H,0EF84H
1C4A 11531C    LXI D,NTF
1C4D CD8A1A    CALL MOVN
1C50 C33C1C    JMP NIL
1C53 0E0F     NTF DW 0F0EH
1C55 200E      DW 0E20H
1C57 010D      DW 0D01H
1C59 05FF      DW 0FF05H
1C5B 41       COMTB DB 41H
1C5C D31B      DW APLAY
1C5E 52        DB 52H
1C5F 791B      DW RECRD
1C61 53        DB 53H
1C62 171B      DW SUSTN
1C64 49        DB 49H
1C65 351B      DW IMMD
1C67 4D        DB 4DH
1C68 501B      DW MPHON
1C6A 58        DB 58H
1C6B 651B      DW XPHON
1C6D 47        DB 47H
1C6E 951B      DW NONMR
1C70 48        DB 48H
1C71 8A1D      DW NONMA
1C73 50        DB 50H
1C74 86F8      DW 0F886H
```

```
1C76 54         5000  DB 54H
1C77 83F8       5020  DW 0F883H
1C79 43         5040  DB 43H
1C7A 911D       5060  DW CLEAR
1C7C 4C         5080  DB 4CH
1C7D 77F8       5100  DW 0F877H
1C7F FF         5120  DB 0FFH
1C80 E5         5140 NREAD PUSH H   NAME READ SUB-PROGRAM
1C81 2184EF     5160  LXI H,0EF84H
1C84 22FEFF     5180  SHLD 0FFFEH
1C87 21101D     5200  LXI H,SNGNM
1C8A CD7A02     5220  CALL SCRN
1C8D E1         5240  POP H
1C8E E5         5260 READ PUSH H
1C8F 21C740     5280  LXI H,40C7H
1C92 CDF600     5300 NEXT CALL 0F6H
1C95 FE24       5320  CPI 24H
1C97 C2B31C     5340  JNZ CR
1C9A 21C740     5360  LXI H,40C7H
1C9D 36FF       5380 READ1 MVI M,0FFH
1C9F CDEF1C     5400  CALL USRCL
1CA2 21C740     5420  LXI H,40C7H
1CA5 7E         5440 READ2 MOV A,M
1CA6 2270F8     5460  SHLD PNTR
1CA9 23         5480  INX H
1CAA FE20       5500  CPI 20H
1CAC CAA51C     5520  JZ READ2
1CAF FEFF       5540  CPI 0FFH
1CB1 E1         5560  POP H
1CB2 C9         5580  RET
1CB3 FE0D       5600 CR CPI 0DH
1CB5 C2C01C     5620  JNZ DEL
1CB8 3ED0       5640  MVI A,0D0H
1CBA CDE100     5660  CALL 0E1H
1CBD C39D1C     5680  JMP READ1
1CC0 FE08       5700 DEL CPI 8
1CC2 C2D41C     5720  JNZ CHAR
1CC5 3EC7       5740  MVI A,0C7H
1CC7 BD         5760  CMP L
1CC8 CA921C     5780  JZ NEXT
1CCB 2B         5800  DCX H
1CCC 0608       5820 BSPA MVI B,8
1CCE CD0301     5840  CALL 103H
1CD1 C3921C     5860  JMP NEXT
1CD4 FE20       5880 CHAR CPI 20H
1CD6 DA921C     5900  JC NEXT
1CD9 FE5B       5920  CPI 5BH
1CDB D2921C     5940  JNC NEXT
1CDE 47         5960  MOV B,A
1CDF E6BF       5980  ANI 0BFH
```

```
1CE1 77          MOV M,A
1CE2 CD0301      CALL 103H
1CE5 3ECF        MVI A,0CFH
1CE7 BD          CMP L
1CE8 CACC1C      JZ BSPA
1CEB 23          INX H
1CEC C3921C      JMP NEXT
1CEF 2182EF     USRCL LXI H,0EF82H
1CF2 3EBE        MVI A,0BEH
1CF4 CDE100      CALL 0E1H
1CF7 C9          RET
1CF8 E5         NSTOR PUSH H   NAME STORE SUB-PROGRAM
1CF9 D5          PUSH D
1CFA C5          PUSH B
1CFB 2A70F8      LHLD PNTR
1CFE EB          XCHG
1CFF CD1B1D      CALL SETHL
1D02 CD8A1A      CALL MOVN
1D05 3E63        MVI A,63H
1D07 CDE100      CALL 0E1H
1D0A 36FF        MVI M,0FFH
1D0C C1          POP B
1D0D D1          POP D
1D0E E1          POP H
1D0F C9          RET
1D10 534F       SNGNM DW 4F53H
1D12 4E47        DW 474EH
1D14 204E        DW 4E20H
1D16 414D        DW 4D41H
1D18 452E        DW 2E45H
1D1A 0D          DB 0DH
1D1B 3A30F8     SETHL LDA NMTBP
1D1E 3C          INR A
1D1F 3230F8      STA NMTBP
1D22 2131F8     INCNM LXI H,NMTAB
1D25 010A00     INCHL LXI B,0AH
1D28 3D         INCH1 DCR A
1D29 C8          RZ
1D2A 09          DAD B
1D2B C3281D      JMP INCH1
1D2E E5         ADSTR PUSH H
1D2F D5          PUSH D
1D30 2A10F8      LHLD MTAB
1D33 23          INX H
1D34 2210F8      SHLD MTAB
1D37 EB          XCHG
1D38 CD7D1D      CALL INHL2
1D3B 73          MOV M,E
1D3C 23          INX H
1D3D 72          MOV M,D
```

```
1D3E D1         7000  POP D
1D3F E1         7020  POP H
1D40 C9         7040  RET
1D41 3A72F8     7060 NXTA LDA WORKA
1D44 3C         7080  INR A
1D45 3272F8     7100  STA WORKA
1D48 FE05       7120  CPI 5
1D4A 37         7140  STC
1D4B 3F         7160  CMC
1D4C C8         7180  RZ
1D4D CD221D     7200  CALL INCNM
1D50 37         7220  STC
1D51 C9         7240  RET
1D52 3E00       7260 MATCH MVI A,0
1D54 3272F8     7280  STA WORKA
1D57 CD411D     7300 MATC1 CALL NXTA
1D5A D0         7320  RNC
1D5B EB         7340  XCHG
1D5C 2A70F8     7360  LHLD PNTR
1D5F 0E08       7380  MVI C,8
1D61 CD5301     7400  CALL 0153H
1D64 C2571D     7420  JNZ MATC1
1D67 3A72F8     7440  LDA WORKA
1D6A CD801D     7460  CALL INH3
1D6D 7E         7480  MOV A,M
1D6E 23         7500  INX H
1D6F 66         7520  MOV H,M
1D70 6F         7540  MOV L,A
1D71 37         7560  STC
1D72 C9         7580  RET
1D73 2144EE     7600 NMDSP LXI H,0EE44H   NAME DISPLAY SUB-PROGRAM
1D76 1131F8     7620  LXI D,NMTAB
1D79 CD8A1A     7640  CALL MOVN
1D7C C9         7660  RET
1D7D 3A30F8     7680 INHL2 LDA NMTBP
1D80 2166F8     7700 INH3 LXI H,NMATB
1D83 3D         7720 INH2 DCR A
1D84 C8         7740  RZ
1D85 23         7760  INX H
1D86 23         7780  INX H
1D87 C3831D     7800  JMP INH2
1D8A D5         7820 NONMA PUSH D
1D8B 210044     7840  LXI H,4400H
1D8E C3E01B     7860  JMP NONA1
1D91 2144EE     7880 CLEAR LXI H,0EE44H
1D94 3E7B       7900  MVI A,7BH
1D96 CDE100     7920  CALL 0E1H
1D99 3E00       7940  MVI A,0
1D9B 3230F8     7960  STA NMTBP
1D9E C1         7980  POP B
```

```
1D9F E1          POP H
1DA0 C9          RET

1DA1 3EC9       MNIT MVI A,0C9H
1DA3 327DF8      STA 0F87DH
1DA6 3EC9        MVI A,0C9H
1DA8 327AF8      STA 0F87AH
1DAB 3277F8      STA 0F877H
1DAE 3280F8      STA 0F880H
1DB1 3283F8      STA 0F883H
1DB4 3286F8      STA 0F886H
1DB7 210041     INIT1 LXI H,4100H
1DBA 2210F8      SHLD MTAB
1DBD CDA725      CALL GRP
1DC0 218000      LXI H,80H
1DC3 2275F8      SHLD ASPD
1DC6 CDD11D      CALL INDP
1DC9 3E00        MVI A,0
1DCB 3274F8      STA INSTS
1DCE C30018      JMP INIT
1DD1 CDB41A     INDP CALL OCT
1DD4 CDF41A      CALL SPLAY
1DD7 CDDE1D      CALL SMPH
1DDA CDE31D      CALL SSUST
1DDD C9          RET
1DDE E5         SMPH PUSH H
1DDF C5          PUSH B
1DE0 C3501B      JMP MPHON
1DE3 E5         SSUST PUSH H
1DE4 C5          PUSH B
1DE5 C3171B      JMP SUSTN
```

PROGRAM FOR GRAPHIC DISPLAY.

```
ASSM 25A7

   25A7 061F      GRP MVI B,1FH   GRAPH SUB-PROGRAM
   25A9 CD0301     CALL 103H
   25AC CDD125     CALL VLDL
   25AF CDE525     CALL VLDR
   25B2 CDEE25     CALL FRSTL
   25B5 CDFC25     CALL HLD
   25B8 110020     LXI D,2000H
   25BB 017FEB     LXI B,0EB7FH
   25BE 2140E8     LXI H,0E840H
   25C1 CDA9F2     CALL 0F2A9H
   25C4 21C0EB     LXI H,0EBC0H
   25C7 114023     LXI D,2340H
   25CA 01FFED     LXI B,0EDFFH
   25CD CDA9F2     CALL 0F2A9H
   25D0 C9         RET
   25D1 2100E8    VLDL LXI H,0E800H
   25D4 06EA       MVI B,0EAH
   25D6 70        VLD1 MOV M,B
   25D7 23         INX H
   25D8 70         MOV M,B
   25D9 2B         DCX H
   25DA 114000     LXI D,40H
   25DD 19         DAD D
   25DE 7C         MOV A,H
   25DF FEF0       CPI 0F0H
   25E1 DAD625     JC VLD1
   25E4 C9         RET
   25E5 213EE8    VLDR LXI H,0E83EH
   25E8 06D5       MVI B,0D5H
   25EA CDD625     CALL VLD1
   25ED C9         RET
   25EE 2100E8    FRSTL LXI H,0E800H
   25F1 36FA      FRST2 MVI M,0FAH
   25F3 16F3      FRST1 MVI D,0F3H
   25F5 23        FRST3 INX H
   25F6 CD2C26     CALL MOVD
   25F9 36F5       MVI M,0F5H
   25FB C9         RET
   25FC 21C0EF    HLD LXI H,0EFC0H
   25FF 36EB       MVI M,0EBH
   2601 CDF325     CALL FRST1
   2604 36D7       MVI M,0D7H
   2606 2181EE    LSTM2 LXI H,0EE81H
   2609 CD1F26     CALL STLN
   260C 2101EF    LSTM1 LXI H,0EF01H
   260F CD1F26     CALL STLN
   2612 2181EB    LSTM4 LXI H,0EB81H
```

```
2615 36CE      0960  MVI M,0CEH
2617 CD2126    0980  CALL STLN1
261A 36CD      1000  MVI M,0CDH
261C 2101EE    1020 LSTM3 LXI H,0EE01H
261F 36EE      1040 STLN MVI M,0EEH
2621 16CC      1060 STLN1 MVI D,0CCH
2623 23        1080  INX H
2624 1E3C      1100  MVI E,3CH
2626 CD2E26    1120  CALL MOVD1
2629 36DD      1140  MVI M,0DDH
262B C9        1160  RET
262C 1E3E      1180 MOVD MVI E,3EH
262E 72        1200 MOVD1 MOV M,D
262F 23        1220  INX H
2630 1D        1240  DCR E
2631 C8        1260  RZ
2632 C32E26    1280  JMP MOVD1
```

DATA FOR GRAPH PROGRAM

```
DUMP 2000 233F
EA EA 85 82 87 84 81 8C 89 85 8C 89 81 8C 89 80
8E 8F 8E 8E 8F 8E 8E 8C 8C 8E 8F 8C 8D 8A 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 D5 D5
EA EA 95 94 95 95 95 80 80 95 8D 98 95 80 95 80
AA 9B 8A AA BF AA AB B3 B2 AA AA BF AE AA 80 04
05 16 05 0C 0F 10 05 04 80 01 14 80 02 15 05 14
80 05 05 05 80 04 05 10 14 80 80 80 80 80 D5 D5
EA EA 94 80 94 94 90 8C 98 94 80 A4 90 8C 98 80
AC BC AC BC 8C 9C AC 8C 8C AC BC 8C 9C A8 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 D5 D5
EA FA B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0
B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0
B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0
B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 F5 D5
EA EA 80 80 80 84 BA BE BC BD B5 A8 A8 80 15 80
80 16 80 80 17 80 80 19 80 80 1A 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 D5 D5
EA EA 80 83 B3 B3 AA BB B3 B7 95 B3 A2 C5 CC CA
C5 CC CA C5 CC CA C5 CC CA C5 CC CA 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80
80 80 80 80 80 80 80 80 80 80 80 80 80 80 D5 D5
EA EA 84 8C 8C 8C AA AE 8C 9D 95 8C 88 D1 F3 E3
D3 F3 E3 D3 F3 E3 D3 F3 E3 D3 F3 E3 C3 C3 C3 C3
C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3
C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 D7 D5
EA EA B3 B3 B3 B3 AA BB B3 B7 95 B3 A2 D5 C0 95
BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80 95 BF 80
D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80
95 BF 80 D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 D5
EA EA B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 B0 D5 C0 95
BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80 95 BF 80
D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80
95 BF 80 D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 D5
EA EA 80 80 80 80 80 80 80 80 80 80 80 D5 C0 95
BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80 95 BF 80
D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 C0 95 BF 80
95 BF 80 D5 C0 95 BF 80 95 BF 80 95 BF 80 D5 D5
EA EA C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 D5 C0 C0
EA C0 C0 EA C0 C0 EA C0 D5 C0 C0 EA C0 C0 EA C0
D5 C0 C0 EA C0 C0 EA C0 C0 EA C0 D5 C0 C0 EA C0
C0 EA C0 D5 C0 C0 EA C0 C0 EA C0 C0 EA C0 D5 D5
EA EA C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 D5 C0 C0
EA C0 C0 EA C0 C0 EA C0 D5 C0 C0 EA C0 C0 EA C0
D5 C0 C0 EA C0 C0 EA C0 C0 EA C0 D5 C0 C0 EA C0
C0 EA C0 D5 C0 C0 EA C0 C0 EA C0 C0 EA C0 D5 D5
EA EB C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 D7 C3 C3
```

```
EB C3 C3 EB C3 C3 EB C3 D7 C3 C3 EB C3 C3 EB C3
D7 C3 C3 EB C3 C3 EB C3 C3 EB C3 D7 C3 C3 EB C3
C3 EB C3 D7 C3 C3 EB C3 C3 EB C3 C3 EB C3 D7 D5
```

```
DUMP 2340 257F
EA EA 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 15 13 05 12 20 13 05 0C 05 03 14 09 0F 0E
13 2F 09 0E 06 0F 12 0D 01 14 09 0F 0E 13 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 D5 D5
EA EA 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 D5 D5
EA EA 20 20 20 20 20 0D 2D 10 08 0F 0E 09 03 20
3A 20 20 0D 20 20 20 FF 20 20 18 2D 10 08 0F 0E
09 03 20 3A 20 20 18 20 20 20 FF 20 20 20 0F 03
14 01 16 20 3A 20 30 2D 35 20 20 20 20 20 D5 D5
EA EA 20 20 20 20 20 09 0D 0D 05 04 09 01 14 05
3A 20 20 09 20 20 20 FF 20 20 13 15 13 14 01 09
0E 20 20 3A 20 20 13 20 20 20 FF 20 20 20 10 12
09 0E 14 20 3A 20 10 20 20 20 20 20 20 20 D5 D5
EA EA 20 20 20 20 20 12 05 03 0F 12 04 20 20 20
3A 20 20 12 20 20 20 FF 20 20 01 15 14 0F 10 0C
01 19 20 3A 20 20 01 20 20 20 FF 20 20 20 14 01
10 05 12 17 3A 20 14 20 20 20 20 20 20 20 D5 D5
EA EA 20 20 20 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 20 20 20 D5 D5
EA EA 20 20 20 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A
20 20 20 20 03 0F 0D 0D 01 0E 04 20 14 05 12 0D
09 0E 01 14 09 0F 0E 20 3A 20 24 20 20 20 2A 2A
2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 20 20 20 D5 D5
EA EA 20 20 20 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 20 20 20 D5 D5
EA EA 20 20 20 08 0F 12 0E 3A 15 20 FF 20 20 06
0C 15 14 05 3A 16 20 FF 20 20 0F 12 07 01 0E 3A
17 20 FF 20 20 16 09 0F 0C 09 0E 3A 19 20 FF 20
20 06 01 0E 14 01 13 13 19 3A 1A 20 20 20 D5 D5
```

APPENDIX B

I. 8085 INSTRUCTION SET SUMMARY BY FUNCTIONAL GROUPING
(FROM INTEL USER'S MANUAL)

II. FUNCTIONAL DESCRIPTION OF 8155 CHIP
(FROM INTEL USER'S MANUAL)

III. FUNCTIONAL DESCRIPTION OF STK-437 I.C. CHIP
(FROM SANYO USER'S MANUAL)

8085A INSTRUCTION SET SUMMARY BY FUNCTIONAL GROUPING
Table 6-1

Mnemonic*	Description	Instruction Code (1) D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Page
MOVE, LOAD, AND STORE										
MOVr1,r2	Move register to register	0	1	D	D	D	S	S	S	5-4
MOV M,r	Move register to memory	0	1	1	1	0	S	S	S	5-4
MOV r,M	Move memory to register	0	1	D	D	D	1	1	0	5-4
MVI r	Move immediate register	0	0	D	D	D	1	1	0	5-4
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	5-4
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	5-5
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	5-5
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	5-5
STAX B	Store A indirect	0	0	0	0	0	0	1	0	5-6
STAX D	Store A indirect	0	0	0	1	0	0	1	0	5-6
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	5-5
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	5-5
STA	Store A direct	0	0	1	1	0	0	1	0	5-5
LDA	Load A direct	0	0	1	1	1	0	1	0	5-5
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	5-5
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	5-5
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	5-6
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	5-15
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	5-15
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	5-15
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	5-15
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	5-15
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	5-15
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	5-15
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	5-15
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	5-16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5-16
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	5-5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5-9
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5-9
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	5-13
JC	Jump on carry	1	1	0	1	1	0	1	0	5-13
JNC	Jump on no carry	1	1	0	1	0	0	1	0	5-13
JZ	Jump on zero	1	1	0	0	1	0	1	0	5-13
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	5-13
JP	Jump on positive	1	1	1	1	0	0	1	0	5-13
JM	Jump on minus	1	1	1	1	1	0	1	0	5-13
JPE	Jump on parity even	1	1	1	0	1	0	1	0	5-13
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	5-13
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5-15
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	5-13
CC	Call on carry	1	1	0	1	1	1	0	0	5-14
CNC	Call on no carry	1	1	0	1	0	1	0	0	5-14

Mnemonic	Description	Instruction Code (1) D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Page
CZ	Call on zero	1	1	0	0	1	1	0	0	5-14
CNZ	Call on no zero	1	1	0	0	0	1	0	0	5-14
CP	Call on positive	1	1	1	1	0	1	0	0	5-14
CM	Call on minus	1	1	1	1	1	1	0	0	5-14
CPE	Call on parity even	1	1	1	0	1	1	0	0	5-14
CPO	Call on parity odd	1	1	1	0	0	1	0	0	5-14
RETURN										
RET	Return	1	1	0	0	1	0	0	1	5-14
RC	Return on carry	1	1	0	1	1	0	0	0	5-14
RNC	Return on no carry	1	1	0	1	0	0	0	0	5-14
RZ	Return on zero	1	1	0	0	1	0	0	0	5-14
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5-14
RP	Return on positive	1	1	1	1	0	0	0	0	5-14
RM	Return on minus	1	1	1	1	1	0	0	0	5-14
RPE	Return on parity even	1	1	1	0	1	0	0	0	5-14
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5-14
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	5-14
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	5-16
OUT	Output	1	1	0	1	0	0	1	1	5-16
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	5-8
DCR r	Decrement register	0	0	D	D	D	1	0	1	5-8
INR M	Increment memory	0	0	1	1	0	1	0	0	5-8
DCR M	Decrement memory	0	0	1	1	0	1	0	1	5-8
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5-9
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5-9
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5-9
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5-9
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5-9
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5-9
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	5-6
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	5-6
ADD M	Add memory to A	1	0	0	0	0	1	1	0	5-6
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	5-7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	5-6
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	5-7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	5-9
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	5-9
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	5-9
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	5-9
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	5-7
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	5-7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	5-7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	5-8
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	5-7

*All mnemonics copyrighted · Intel Corporation 1976.

8085A INSTRUCTION SET SUMMARY (Cont'd)

Table 6-1

Mnemonic	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Page
		Instruction Code (1)								
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	5-8
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	5-9
XRA r	Exclusive OR register with A	1	0	1	0	1	S	S	S	5-10
ORA r	OR register with A	1	0	1	1	0	S	S	S	5-10
CMP r	Compare register with A	1	0	1	1	1	S	S	S	5-11
ANA M	And memory with A	1	0	1	0	0	1	1	0	5-10
XRA M	Exclusive OR memory with A	1	0	1	0	1	1	1	0	5-10
ORA M	OR memory with A	1	0	1	1	0	1	1	0	5-11
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	5-11
ANI	And immediate with A	1	1	1	0	0	1	1	0	5-10
XRI	Exclusive OR immediate with A	1	1	1	0	1	1	1	0	5-10
ORI	OR immediate with A	1	1	1	1	0	1	1	0	5-11
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	5-11
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	5-11
RRC	Rotate A right	0	0	0	0	1	1	1	1	5-12
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	5-12
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	5-12
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	5-12
STC	Set carry	0	0	1	1	0	1	1	1	5-12
CMC	Complement carry	0	0	1	1	1	1	1	1	5-12
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	5-9
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	5-17
DI	Disable Interrupt	1	1	1	1	0	0	1	1	5-17
NOP	No-operation	0	0	0	0	0	0	0	0	5-17
HLT	Halt	0	1	1	1	0	1	1	0	5-17
NEW 8085A INSTRUCTIONS										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	5-17
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	5-18

NOTES: 1. DDS or SSS: B 000, C 001, D 010, E011, H 100, L 101, Memory 110, A 111.

2. Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.

intel

8155/8156/8155-2/8156-2
2048 BIT STATIC MOS RAM WITH I/O PORTS AND TIMER

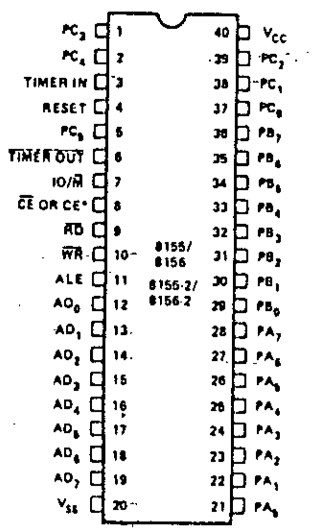
- 256 Word x 8 Bits
- Single +5V Power Supply
- Completely Static Operation
- Internal Address Latch
- 2 Programmable 8 Bit I/O Ports
- 1 Programmable 6-Bit I/O Port
- Programmable 14-Bit Binary Counter/Timer
- Compatible with 8085A and 8088 CPU
- Multiplexed Address and Data Bus
- 40 Pin DIP

The 8155 and 8156 are RAM and I/O chips to be used in the 8085A and 8088 microprocessor systems. The RAM portion is designed with 2048 static cells organized as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085A CPU. The 8155-2 and 8156-2 have maximum access times of 330 ns for use with the 8085A-2 and the full speed 5 MHz 8088 CPU.

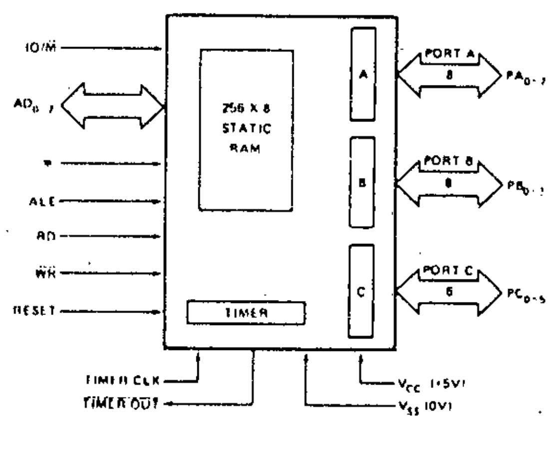
The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

PIN CONFIGURATION



BLOCK DIAGRAM



*: 8155/8155-2 = $\overline{CE}$, 8156/8156-2 = CE

8155/8156 PIN FUNCTIONS

Symbol	Function
RESET (input)	Pulse provided by the 8085A to initialize the system (connect to 8085A RESET OUT). Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two 8085A clock cycle times.
AD_{0-7} (input)	3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch inside the 8155/56 on the falling edge of ALE. The address can be either for the memory section or the I/O section depending on the IO/$\overline{M}$ input. The 8-bit data is either written into the chip or read from the chip, depending on the $\overline{WR}$ or $\overline{RD}$ input signal.
CE or $\overline{CE}$ (input)	Chip Enable: On the 8155, this pin is $\overline{CE}$ and is ACTIVE LOW. On the 8156, this pin is CE and is ACTIVE HIGH.
$\overline{RD}$ (input)	Read control: Input low on this line with the Chip Enable active enables and AD_{0-7} buffers. If IO/$\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/status registers will be read to the AD bus.
$\overline{WR}$ (input)	Write control: Input low on this line with the Chip Enable active causes the data on the Address/Data bus to be written to the RAM or I/O ports and command/status register depending on IO/$\overline{M}$.
ALE (input)	Address Latch Enable: This control signal latches both the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
IO/$\overline{M}$ (input)	Selects memory if low and I/O and command/status registers if high.
PA_{0-7} (8) (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PB_{0-7} (8) (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PC_{0-5} (6) (input/output)	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the command register. When PC_{0-5} are used as control signals, they will provide the following: PC_0 — A INTR (Port A Interrupt) PC_1 — ABF (Port A Buffer Full) PC_2 — $\overline{A\ STB}$ (Port A Strobe) PC_3 — B INTR (Port B Interrupt) PC_4 — $\overline{B\ BF}$ (Port B Buffer Full) PC_5 — B STB (Port B Strobe)
TIMER IN (input)	Input to the counter-timer.
$\overline{TIMER\ OUT}$ (output)	Timer output. This output can be either a square wave or a pulse depending on the timer mode.
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.

DESCRIPTION

The 8155/8156 contains the following:

- 2k Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
- 14-bit timer-counter

The IO/$\overline{M}$ (IO/Memory Select) pin selects either the five registers (Command, Status, PA_{0-7}, PB_{0-7}, PC_{0-5}) or the memory (RAM) portion. (See Figure 1.)

The 8-bit address on the Address/Data lines, Chip Enable input CE or $\overline{CE}$, and IO/$\overline{M}$ are all latched on-chip at the falling edge of ALE. (See Figure 2.)

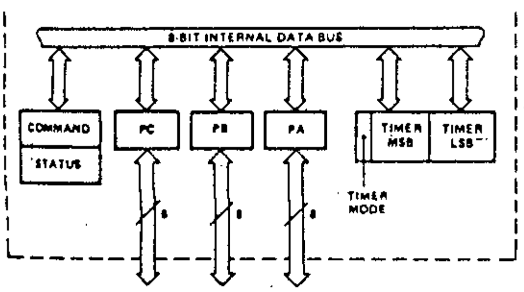
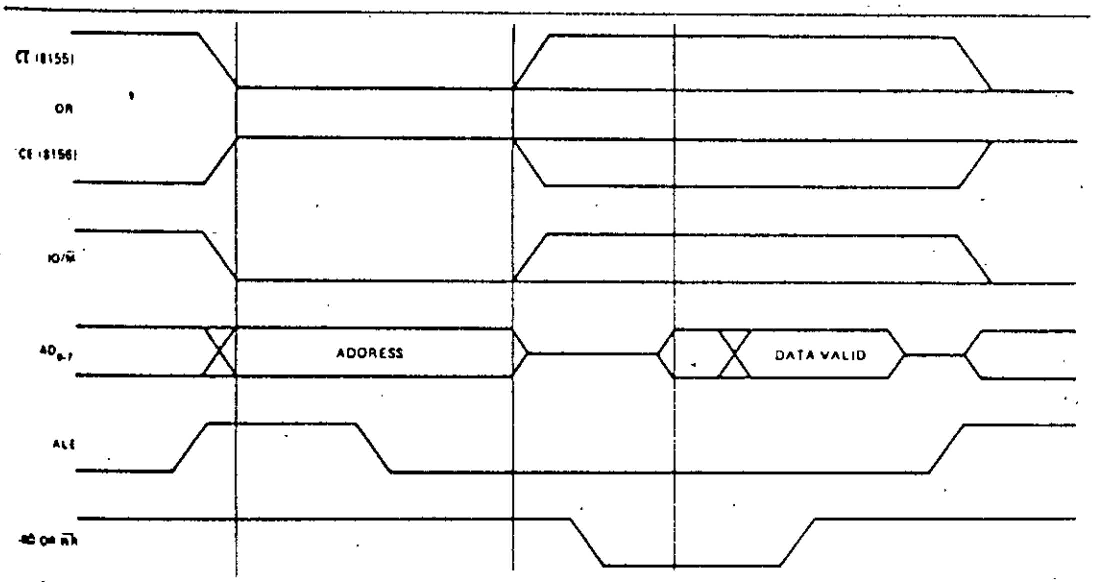


Figure 1. 8155/8156 Internal Registers



NOTE: FOR DETAILED TIMING INFORMATION, SEE FIGURE 12 AND A.C. CHARACTERISTICS.

Figure 2. 8155/8156 On-Board Memory Read/Write Cycle

PROGRAMMING OF THE COMMAND REGISTER

The command register consists of eight latches. Four bits (0-3) define the mode of the ports, two bits (4-5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6-7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation with the Chip Enable active and $IO/\overline{M} = 1$. The meaning of each bit of the command byte is defined in Figure 3. The contents of the command register may never be read.

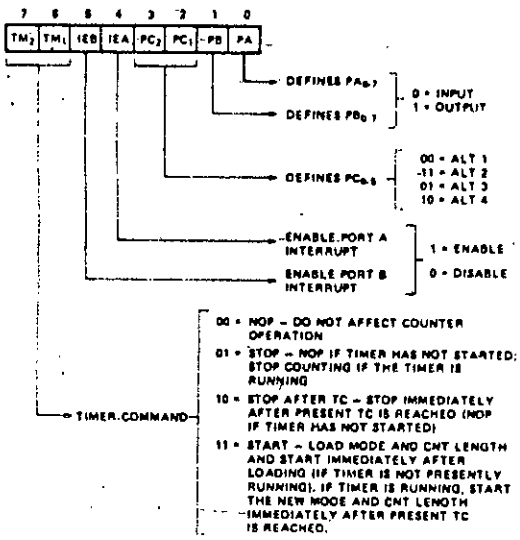


Figure 3. Command Register Bit Assignment

READING THE STATUS REGISTER

The status register consists of seven latches, one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in Figure 4. Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

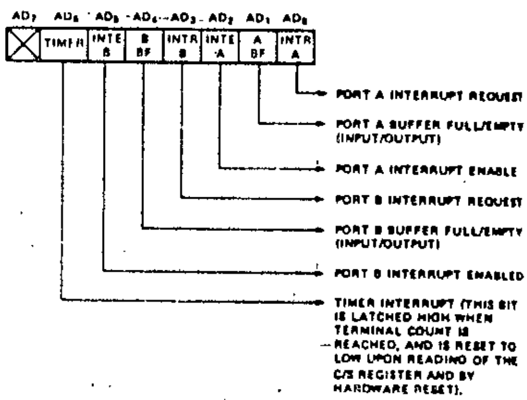


Figure 4. Status Register Bit Assignment

INPUT/OUTPUT SECTION

The I/O section of the 8155/8156 consists of five registers: (See Figure 5.)

- **Command/Status Register (C/S)** — Both registers are assigned the address XXXXX000. The C/S address serves the dual purpose.

  When the C/S registers are selected during WRITE operation, a command is written into the command register. The contents of this register are *not* accessible through the pins.

  When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.

- **PA Register** — This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (See timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.

- **PB Register** — This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.

- **PC Register** — This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.

  When PC_{0-5} is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155 sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. (See Table 1.)

When the 'C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

CONTROL	INPUT MODE	OUTPUT MODE
BF	Low	Low
INTR	Low	High
STB	Input Control	Input Control

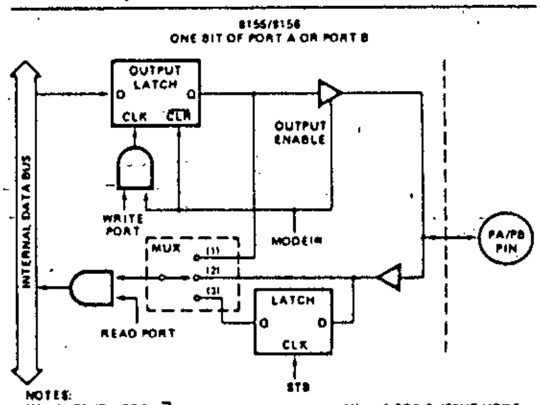
I/O ADDRESS†								SELECTION
A7	A6	A5	A4	A3	A2	A1	A0	
X	X	X	X	X	0	0	0	Interval Command Status Register
X	X	X	X	X	0	0	1	General Purpose I O Port A
X	X	X	X	X	0	1	0	General Purpose I O Port B
X	X	X	X	X	0	1	1	Port C - General Purpose I O or Control
X	X	X	X	X	1	0	0	Low-Order 8 bits of Timer Count
X	X	X	X	X	1	0	1	High 6 bits of Timer Count and 2 bits of Timer Mode

X: Don't Care

† I/O Address must be qualified by CE = 1 (8156) or $\overline{CE}$ = 0 (8155) and IO/$\overline{M}$ = 1 in order to select the appropriate register

Figure 5. I/O port and Timer Addressing Scheme

Figure 6 shows how I/O PORTS A and B are structured within the 8155 and 8156:



NOTES:
(1) OUTPUT MODE
(2) SIMPLE INPUT
(3) STROBED INPUT
— MULTIPLEXER CONTROL

(4) = 1 FOR OUTPUT MODE
= 0 FOR INPUT MODE

READ PORT = (IO/$\overline{M}$=1) • ($\overline{RD}$=0) • (CE ACTIVE) • (PORT ADDRESS SELECTED).
WRITE PORT = (IO/$\overline{M}$=1) • ($\overline{WR}$=0) • (CE ACTIVE) • (PORT ADDRESS SELECTED)

Figure 6. 8155/8156 Port Functions

TABLE 1. TABLE OF PORT CONTROL ASSIGNMENT.

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

The outputs of the 8155/8156 are "glitch-free" meaning that you can write a "1" to a bit position that was previously "1" and the level at the output pin will not change.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go low. When the 8155/56 is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of PORT C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Figure 7 shows how the 8155/8156 I/O ports might be configured in a typical MCS-85 system.

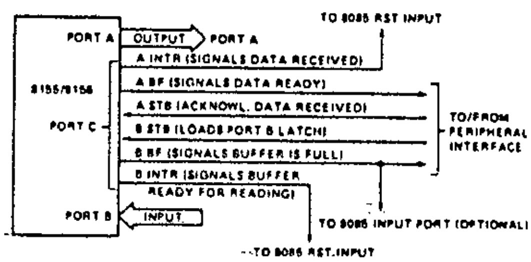


Figure 7. Example: Command Register = 00111001

TIMER SECTION

The timer is a 14-bit down-counter that counts the TIMER IN pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high-order byte of the register. (See Figure 5).

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0-13 of the high order count register will specify the length of the next count and bits 14-15 of the high order register will specify the timer output mode (see Figure 8). The value loaded into the count length register can have any value from 2H through 3FFH in Bits 0-13.

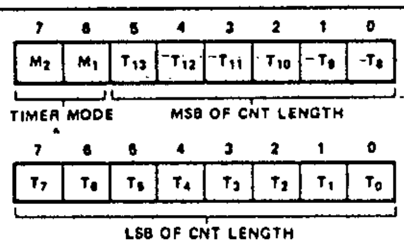


Figure 8. Timer Format

There are four modes to choose from: M2 and M1 define the timer mode, as shown in Figure 9.

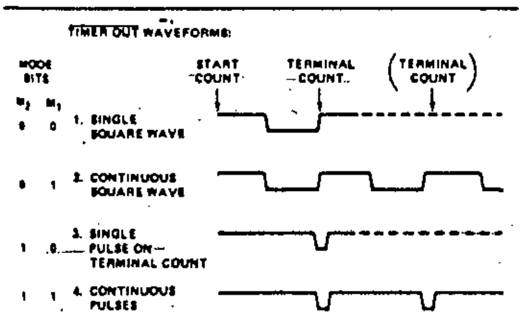


Figure 9. Timer Modes

Bits 6-7 (TM_2 and TM_1) of command register contents are used to start and stop the counter. There are four commands to choose from:

TM_2	TM_1	
0	0	NOP — Do not affect counter operation.
0	1	STOP — NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC — Stop immediately after present TC is reached (NOP if timer has not started)
1	1	START — Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

Note that while the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be used by the counter, you must issue a START command to the counter. This applies even though you may only want to change the count and use the previous mode.

In case of an odd-numbered count, the first half-cycle of the squarewave output, which is high, is one count longer than the second (low) half-cycle, as shown in Figure 10.

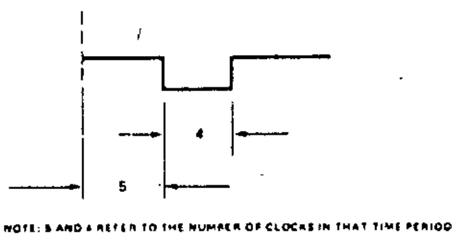


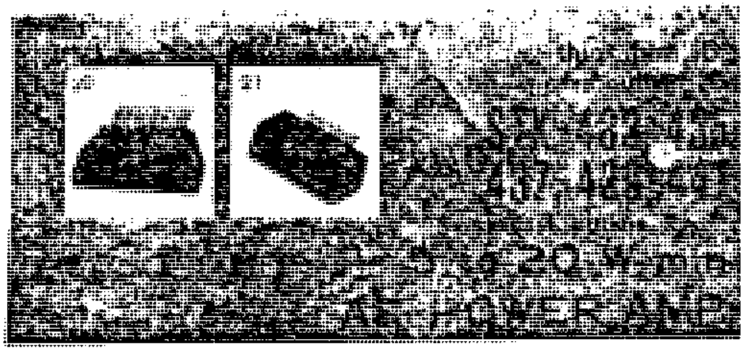
Figure 10. Asymmetrical Square-Wave Output Resulting from Count of 9

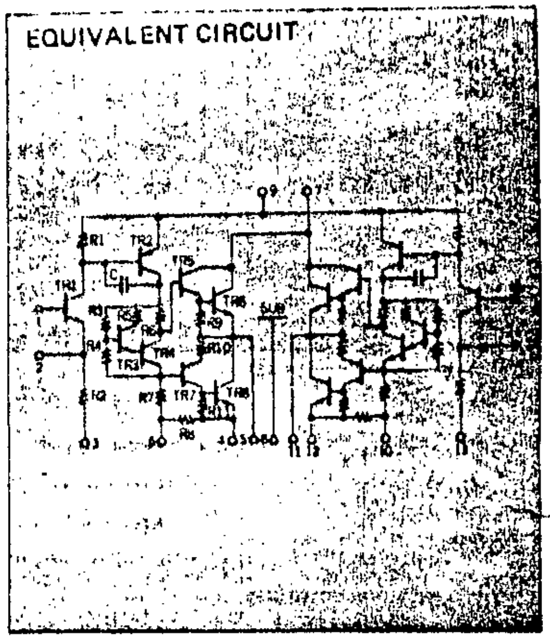
The counter in the 8155 is not initialized to any particular mode or count when hardware RESET occurs, but RESET does *stop* the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

Please note that the timer circuit on the 8155/8156 chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary) or 2 (decimal). (For the detection of single pulses, it is suggested that one of the hardware interrupt pins on the 8085A be used.) After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count
2. Read in the 16-bit value from the count length registers
3. Reset the upper two mode bits
4. Reset the carry and rotate right one position all 16 bits through carry
5. If carry is set, add 1/2 of the full original count (1/2 full count — 1 if full count is odd).

Note: If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the 8155/56 always counts out the right number of pulses in generating the TIMER OUT waveforms.





Features
- ⊙IMST, 2 Channels by 1 Power Supply
- Small shock noise because of direct coupling emitter feedbacked
- AF output power STK-433: 5W min., STK-435: 7W min., STK-437: 10W min., STK-439: 15W min., STK-411: 20W min.

ABSOLUTE MAXIMUM RATINGS / Ta=25°C

		STK-433	STK-435	STK-437	STK-439	STK-441	
Maximum Supply Voltage (pin 7 to 4 or 12)	Vcc max	32	39	50	56	63	V
Operating Case Temperature	T_C	90	90	90	85	85	°C
Storage Temperature	Tstg	→	→	→	→	-30 to +100	°C
Allowable Load Shorting Time (in appointed condition)	ts	→	→	→	→	2	sec

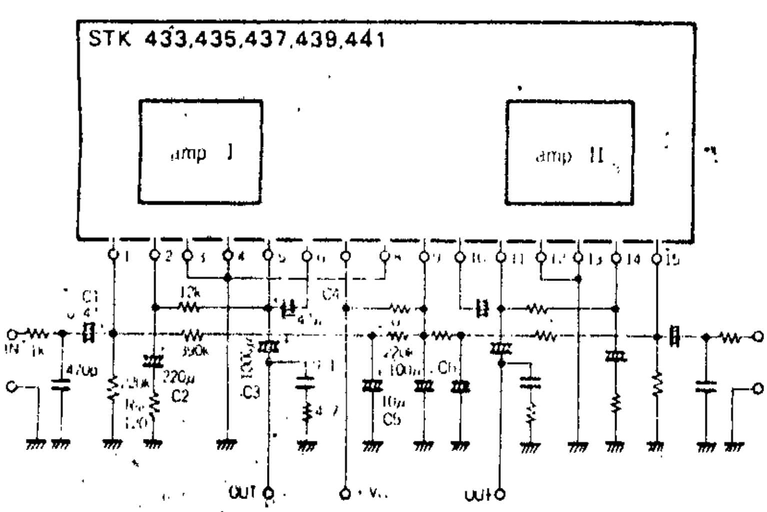
RECOMMENDED OPERATION CONDITION / Ta=25°C

		STK-433	STK-435	STK-437	STK-439	STK-441	
Supply Voltage	Vcc	23	27	33	39	44	V
Load Resistance	R_L	→	→	→	→	8	Ω

OPERATION CHARACTERISTICS / Ta=25°C, recommended condition, Pg=600Ω, VG=40dB

			STK-433	STK-435	STK-437	STK-439	STK-441	unit
Quiescent Current	Icco	recommended Vcc	→	→	→	→	120	mA max
Output Power	Po	THD=1%, f=1kHz	5	7	10	15	20	Wmin
Distortion	THD	Po=0.1W, f=1kHz	0.5	0.5	0.2	0.2	0.3	%max
Frequency Response	f	Po=0.1W	→	→	→	→	110k	

APPLICATION: AF Power Amp.



品 名	STK-433	STK-435	STK-437	STK-439	STK-441
C1	16V	25V	35V	35V	30V
C2	10V	25V	25V	25V	25V
C3	16V	25V	35V	35V	50V
C4	16V	16V	25V	25V	35V
C5	25V	25V	35V	35V	35V
C6	25V	25V	50V	63V	63V

See the operation characteristics on this specification.