PRL

TECHNICAL NOTE

TN-91-71

FOUR CHANNEL AUDIO CASSETTE BASED DATA RECORDER

R. N. MISRA

JANUARY 1991

PHYSICAL RESEARCH LABORATORY
NAVRANGPURA
AHMEDABAD 380 009

DOCUMENT CONTROL AND DATA SHEET

1.	Report No.	:	PRL-TN-91-71
2.	Title and subtitle Four channel Audio Cassette based Data Recorder		
3.	Report Date	:	January 1991
4.	Type of Report:		Technical Report
5.	Pages	:	58
6.	Price	:	Unpriced
7.	No.of References	:	10
8.	Author	:	R.N. Misra
9.	A. Purpose	:	Hardware and software design details of the instrument
	B. Useful for	:	Operation and maintenance of the instrument
10.	Originating Unit	:	Planetary Atmospheres & Aeronomy Area, Physical Research Laboratory, Navrangpura, Ahmedabad 380 009
11.	Sponsoring Agency	:	None
12.	Abstract	:	The audio cassette is a cost effective medium for storage of scientific data. The instrument described here can store about 1.5 M bytes data on a C-60 cassette. It utilizes 8085A microprocessor for controlling the tape recorder, multiplexing of input signals and their sampling as well as digitization.
13.	Key words	:	Data Recorder, Audio Cassette, Microprocessor 8085A, Instrumentation.
14.	Distribution statement	:	Distribution unlimited
15.	Security Classification	:	Unclassified

CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

FOUR CHANNEL AUDIO CASSETTE BASED DATA RECORDER

R.N.Misra

ABSTRACT

A four channel data recording system based on audio cassette recorder has been designed.The channels have sampling rate of 5 per second and can be paralleled to provide two channels of 1Ø per second or one channel of 2Ø per second .The system has been designed around a 8Ø85A microprocessor and has facilities for software debugging as well.The current date ,time and three character station code can be entered through the keyboard, which gets displayed on the seven segment display during the execution of the main recording program.Time is updated by software and station and time codes are recorded along with the data in the beginning of the data block.The instrument has facility to read the data from the cassette tape for verification and transfer to floppy or hard disk of a PC, if so desired.

Keywords:Data Recorder,Audio Cassette,Microprocessor,
8Ø85A,Instrumentation.

1. INTRODUCTION

Audio cassette recorder is a cost effective device for unattended recording of data. It can be easily driven under microprocessor control to realise a data recording instrument that can replace a pen and ink type recorder. A four channel data recorder based on 8Ø85A microprocessor has been designed for such applications. The microprocessor controls the multiplexer, data aquisition, storage and subsequent recording on the audio cassette in block mode. It also updates a real time clock under software control. The current time, day of the year, year and station code are recorded in the beginning of every block to facilitate data analysis. The instrument has a keyboard, fifteen nos of 7 segment displays and a set of eight LED's which facilitate control of the instrument. The keyboard is used for entering time and station code and initiating the data recording. The seven segment displays indicate memory address and contents during the initialisation as in common microprocessor kits and station code , year, day of the year and current time during the operation of the instrument. The eight LED's display recording status. The keyboard and the displays have been located on the front panel, which has a gentle slope to provide comfort to the operator. The system reset is provided on the keyboard and the power on-off, as well as record /play switches have been located on the rear panel of the instrument. It can also be used as a microprocessor kit similar to SDK-85 on power-up .An additional switch located on the rear panel selects RS232C port for this purpose. The baud rate has been fixed at 12ØØ by software.

2.Ø DESIGN

The block diagram of the instrument has been given in fig.1.It consists of four distinct parts ,the microcomputer ,keyboard/display, data conversion /tape interface electronics and power supply.These have been covered in the following sections.

2.1 THE MICROCOMPUTER

The microcomputer has been designed around 8Ø85A microprocessor.It is a small ,single board computer designed to be software compatible to SDK-85 of Intel.The block diagram of the microcomputer has been given in the fig.2.The board has been covered in detail elsewhere(Ref.1),and only its salient features would be discussed here.

The microcomputer uses 8212 latch for generation of stable low address lines AØ-A7 and 8155 as general I/O device.The 8253 and 8251 have been used for generation of various timing signals and sending serial data to the cassette tape write electronics.MR*,MW*,IOW* and IOR* signals have been generated from RD*,WR*and IO/M* signals of the 8Ø85A with the help of three to eight decoder type 74LS138.Another 74LS138 has been used for generation of chip select signals for memory and I/O.A dual 2 to 4 decoder type 74LS155 has been used for generation of chip select signals for 4K EPROM type 2732 and the two 8K RAM's type 6264.While the 74LS155 has been wired on a piggyback board,The two 6264's have been accomodated in one socket of 6116 by mounting those one over the other.Chip select lines of 2732 and 6264 as well as A12,Vcc and WE*(MW*) have been wired point to point.The memory and I/O maps and decoding logic has been given in fig.3.

2.2 KEYBOARD AND DISPLAY

A keyboard consisting of 24 single contact key switches has been used for controlling the system.A set of eight LED's are used for displaying the status information and fifteen 7 segment LED displays for station code and time.The display codes for alphanumeric data stored in ASCII and segment assignments of 7 segment displays have been given in appendix C and D .The keyboard and display have been accomodated on a single pcb which has been mounted just behind the front panel.The circuit diagram of the keyboard display unit has been given in fig.6 .It is based on the keyboard display controller type 8279 of INTEL.The scan lines SCØ-SC2 of 8279 are decoded in a 3 to 8 decoder type 74LS138 , the first three outputs OØ,O1,O2 of which have been used for scanning the key matrix.Two additional decoders of the same type have been used for decoding all the four scan lines SCØ-SC3 for scanning the common anodes of the 7 segment displays as well as eight status LED's.The high current requirement is fulfilled by the use of 16 driver transistors.The return lines RLØ-RL7 are connected to the key matrix.The cathodes of the 7 segment displays as well as eight status LED's are driven in parallel by eight transistors controlled by the eight output lines AØ-A3 and BØ-B3 of the 8279.The leftmost digit is driven by the output OØ of 74LS138 while the successive digits are driven by other outputs of this decoder and the next one.The output O7 of the second 74LS138 controls the eight status LED's.As shown in the memory map the keyboard controller IC is located at 18ØØh.The register select line AØ of the 8279 is connected to the address line A8 and therefore the command/status register is mapped at 19ØØh and the data register at 18ØØh.

2.3 DATA CONVERSION AND TAPE INTERFACE

The circuit diagram of the data pcb has been given in the fig.4.One D/A converter type AD7533 has been connected to the port A of 8155 to generate the analog output corresponding to the digital input.The conversion of the input analog signal into digital form is accomplished by comparing the same to the output of the D/A converter referred above,in a comparator type LM311,and testing the output of the latter at port line PBØ of the 8155.The successive approximation algorithm has been implemented in software.The two output lines of the port C of 8155 have been used for multiplexing four analog inputs into one and also demultiplexing output of the D/A convertor into four lines.A dual four to one multiplexer/demultiplexer type CD4052 has been used for this purpose.It is operated at +5 volts and can therefore handle signals of 0-5 volt range. The third line of 8155 is used for enabling the S/H circuit realised by four sections of the quad op-amp type LM324.The input lines have been buffered by another LM324 and protected by a set of diode clamps.

The cassette tape recorder interface includes data write as well as data read interface.The data write interface has been realised by a pair of JK flip-flops of CD4027,one binary counter type CD4040 and a few gates.The USART 8251 has been programmed to operate in X16 clock mode.Accordingly the counter chain of 8253 is also initialised to generate 76800 Hz clock to get a data rate of 4800 from 8251.The dividing factors of the counter in 8155 as well as those of 8253 have been given in fig .1.The X16 clock of 76800 Hz also drives a counter type CD4040 to generate X2 clock.The set input of the first JK flip-flop is driven by the serial data line of 8251 and its J and K inputs are held at logical ONE.The X2 clock is connected to the clock input of the JK flip-flop and therefore Q output of the latter remains at state ONE for logical ONE output from 8251 and toggles at every positive transition of the clock if the output of 8251 is logical ZERO.The Q output of the first JK FF drives the J&K inputs of the second JK FF,the clock of which is also two times the bit rate.The output of the second JK FF is the required Biphase-Mark signal. It is attenuated and shaped in a R-C network to limit its upper end of frequency spectrum. Biphase-Mark coding is especially suitable for use in tape recorder applications as it is not affected by the signal inversion in record reproduce electronics internal to the cassette tape recorder. Biphase-Mark is characterised by two zero crossings for one bit period for logical ONE and only one for logical ZERO. The waveforms have been given in fig.5 . This scheme has been described in detail in Ref.7.

Coming to the data recovery, it is necessary to point out that the phase shifts inherent in the record reproduce process have no effect on so far as audio applications are concerned, however, the phase equalisation is needed for proper decoding of the recorded waveform. This has been accomplished in the circuit by an op-amp type 741 or a section of 324. The output of the phase equaliser in turn drives a zero crossing detector implemented by a comparator LM311 to generate a nearly square wave. The rising and falling edges of this signal are made to produce narrow pulses of about 5 microsecond width with the help of an XOR gate CD4030. Time period between these pulses gives estimate of ONE and ZERO in the recorded data. Pulses separated by about one bit period represent ZERO and two pulses separated by half bit period indicate logical ONE. This signal can be decoded by hardware using phase lock loop as explained in Ref.7 ,or else software technique can be used to decipher data. The latter has been utilized in the present system to reduce hardware.

Some of the cassette tape recorders require isolated or floating contacts for the operation of the Remote function. A pcb type relay has therefore been used for the purpose.It is driven by a transistor which in turn is controlled by port line PC3 of 8155.

2.4 POWER SUPPLY I

Power supply is of straight forward design and needs no special treatment.The circuit diagram of the power supply has been given in fig 7 .An alternative power supply with provision for battery backup has also been designed.

2.5 POWER SUPPLY II

The instrument draws about one ampere at 5 volts ,Ø.5 amperes at +12 Volts and Ø.1 amperes at -12 Volts.Previous models of the instrument derived this power from a linear regulator standard power supply energised by 23Ø V AC power source (power supply I).The instrument requires initialization at every power on sequence and therfore frequent power failures defeat the very purpose of unattended operation.A power supply system normally running on 23Ø V ac power source and switching over to

12V lead acid car battery has therefore been designed.A raw dc of +15 Volts is generated from 230 V ac ,which in turn drives a switched mode power supply through a power diode OR gate.The battery is normally kept at trickle charge at C/1000 rate and supplies power during power failures.A battery of 10 Ah capacity can run the instrument for about ten hours.The SMPS is based on the design described in Ref.8 .It uses SG3524 IC as control element and utilizes a transformer wound on the U-core of the horizontal deflection transformer of 12/14 in.TV.It generates +12V,-12V and +5V, out of which +5V is regulated through feedback.12V supplies ,both positive as well as negative,have only about 2 percent regulation ,which is however sufficient for the various circuits.The SMPS is adequately shielded by enclosing it in an aluminium box to reduce the interference to the other circuits. The circuit diagram of the power supply II has been given in Fig.8.

2.6 CONSIDERATIONS FOR THE CASSETTE TAPE RECORDER

Virtually any model of standard monophonic audio cassette tape recorder can be used with the present system.However a separate tape recorder requires extensive cabling apart from power connection which adds to the confusion.One of the models of the instrument have therefore been designed to include a built in tape recorder.Car cassette player mechanisms are fairly compact and can easily be converted into a mono recorder by adding erase head and replacing the read/write head by mono one.The car cassette deck has only one lever for tape forward and reverse motion as well as eject function.The tape starts moving just after insertion of the cassette but REMO function can be implemented by stopping the motor .The record/reproduce switch has been installed on the tape pcb and is accessible from the outside.It is of the push button type having six poles and two way connections. The BEL1895 has been used for audio amplifier section.The circuit is of standard type and dc bias and dc erase has been used (Ref.9 and 1Ø).

2.7 OPERATION AND SOFTWARE DETAILS

The first 4K bytes of memory is earmarked for the monitor program.It is updated version of the one included in the SDK-85 kit and can be used in the similar way.It is used for entry of date ,time and station code as well as data read and debugging purposes.The operational software of the instrument is located at 28ØØh-2FFFh and it is to be copied into the 1ØØØh-17FFh area of RAM prior to use.This can be conveniently done with the help of program located at 3C8Øh.The memory buffer of 16K bytes has been split in to two parts of 8K each located at 4ØØØh and 6ØØØh.These are designated as buffer I and buffer II respectively.While one of the buffers is used for storage of data being currently acquired, the other one is used for transfering the data acquired in the previous cycle, to the cassette.The data collection rate totals to 2Ø samples per second for all the four channels which may be distributed to realise two channels of 10 per sec rate ,four channels of 5 per sec rate or just a single channel of 20 per second.Data is stored in the buffer in the sequence of channels 1,2,3,4.

First 16 bytes of each of the buffers costitute the identification information which includes time of acquiring the present data and station code and the rest of the memory of this block contains the collected data. The order of the identification bytes has been given in the appendix G .The recording rate is 4800 bits per second and therefore 8k bytes of data would require only about 20 seconds to record on to cassette.A leader consisting of continuous 4800 Hz is recorded for two seconds after the tape is set in motion .A sequence of 128 ZERO's and 128 ONES is recorded after this tone for synchronization purposes.It is followed by a sync byte A5 ,two bytes of byte count and a blank.The data is recorded after this leader.The checksum of the data is recorded in the last byte which is followed by 128 ZERO's and 128 ONES's once again.The tape is stopped after recording a small length of tape containing 4800 Hz tone.Thus the tape recorder runs for about 25 seconds with a gap of 400 seconds.Thus a C-60 cassette would last for about 8 hours per side.The C-90 cassette can also be used to increase the time but as the tape is thinner ,it is more likely to get stuck.

The software consists of the following modules:

2.7.1.Initialization; It loads the vartious addresses in different pointers ,initializes the I/O devices,displays time and enables interrupts to start data collection operation.The clock starts ticking and thus confirms the operation of the system.The multiplexer control LED's also flicker.The flow chart of this module along with that of the A/D converter has been given in the appendix A.The interrupt line RST7.5 is used for all the operations ,clock update, display update as well as data acquisition.The interrupt vector is modified by each of the software modules to point to the next module ,to enable the operation of the following module at the occurrance of the next interrupt.

2.7.2.Clock module; It updates the current time at the 6Ø pps time base generated in the timer chain.The time is stored in the RAM as well as displayed on the seven segment displays.Clock is of 24 hour type and day of the year is counted upto 3 digits.Year is not updated as it is not needed.The flow chart of the clock and display modules have been given in the appendix B .The interrupt vector is altered to point to the display module at the end of this subroutine.

2.7.3.Display module;This module updates the display on receipt of the interrupt.The 6Ø pps pulse referred above is used to interrupt the microprocessor.the clock update,display update and the data acquisition procedure take place on the ocurrance of interrupt sequentially.Thus each of these operations take place 20 times per second.The display subroutine modifies the interrupt vector to point to the data acquisition module.

2.7.4.Data conversion;This subroutine is the basic one for this system.It stores one byte of data in RAM,advances channel count ,converts the analog signal to digital by successive approximation method and updates various pointers.It also checks for buffer full to initiate recording process and switchover the buffers.Data recording is a simple affair, data is presented to the USART 8251 byte by byte in software loop,which serialises and sends it to tape write electronics.

As it has been mentioned earlier, only the RST7.5 interrupt line has been used for the system.As there is ample time between interrupts,data acquisition and recording can take place simultaneously.

2.7.5.Data read or verify;The data read operation has been realised by software.The data read operation is usually performed after data collection and hence microprocessor can be made use of for this purpose.The recorded data from the tape is fed to an edge detector which generates an interrupt signal .This signal is routed to RST7.5 by means of a logic gate under software control.A software counter provides the count between each edge which is read through interrupt subroutine.The counter is driven by a 300 Khz pulse train ,which in turn is taken from the output of the 8155 timer out terminal.The interrupt subroutine reads the counter and again resets and restarts the counter.A hardware counter implemented in 8253 with interrupt was used because software counters using timing loops could provide very small counts for such a fast data rate.The decision level therefore could not be precisely fixed,resulting in poor reliability of the decoder.With a dedicated counter and high frequency clock,the tolerance on the decision level became sufficiently high to enable error free decoding.The decoding scheme has been shown in appendix E with the help of a flow chart.It decodes longer intervals between pulses as ZERO and two pulses of smaller interval as ONE.This method could even be used for a data rate of 72ØØ bps with success.

3.Ø CONCLUSION

A four channel data recording system utilizing the standard audio cassette medium has been designed.It has been successfully used for the recording of Ionospheric Scintillations at PRL Ahmedabad and Department of Physics, Saurashtra university, Rajkot.It can easily be used to record any other information.Marginal software changes can be made in the software to increase the recording time at the cost of channel bandwidth.It would prove to be quite convenient for unattended data gathering.

ACKNOWLEDGEMENTS

This instrument is the upgraded version of the Ionospheric Scintillation Recorder designed by the author earlier and described in Ref.2 .The hardware and software refinements were implemented by Mr.Akhilesh Kumar during his tenure at PRL during 1988-89.Messrs. M.B.Dadhania,R.I.Patel,and Mr.Dineh Patel of A.V.P.T. Rajkot contributed toward the design and fabrication of SMPS and Tape Recorder electronics.Mr.H.D.Parikh did all the electronic assembly of the instrument and the diagrams included in the document were drawn by Mr.G.A.Panchal.The instrument was designed at the suggestion of Dr.Harish Chandra and Dr.G.D.Vyas took all the pains to check the proper operation in field.Facilities provided by Prof.K.N.Iyer of the Department of Physics ,Saurashtra University ,Rajkot during the field trial of this instrument and help made available during that period from his collegues is thankfully acknowledged.

REFERENCES

1.A LOW COST TOUCH KEYBOARD CRT TERMINAL,R. N. Misra and I.T. Kripalani,PRL Technical note TN-86-53,March 1986.

2.A MICROPROCESSOR BASED IONOSPHERIC SCINTILLATION DATA RECORDER,R.N.Misra and M.B.Dadhania,PRL Technical Note TN-86-56,October 1986.

3.MCS-8Ø/85Family users manual ,1972,Intel Corporation,3Ø65 Bowers Avenue Santa Clara, California 95Ø51,USA.

4.SDK-85 System Design Kit users manual,1978,Intel Corporation,3Ø65 Bowers Avenue,Santa Clara,California 95Ø51,USA.

5.Components Data Catalog 198Ø,Intel Corpn.

6.Data Acquisition Components and Subsystems 198Ø,Analog Devices,Route One Industrial Park,P.O.Box 28Ø,Norwood,Massachusetts,Ø2Ø62,USA.

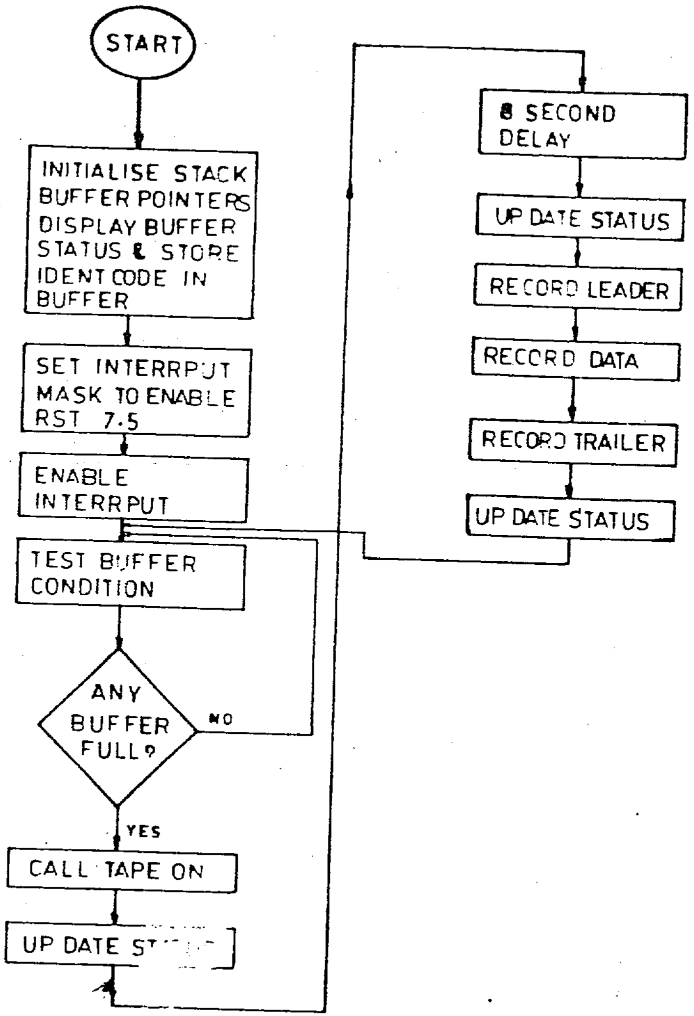
7.Smith,P. and Zokoczy,P.I. 'DATA RECORDING ON AUDIO CASSETTE',Wireless World,Feb.1982,pp5Ø-52 and 62.

8.CONTROL IC's FOR SMPS APPLICATIONS, A.K.Maini ,Electronics For You,November 1989,pp1Ø5-114.

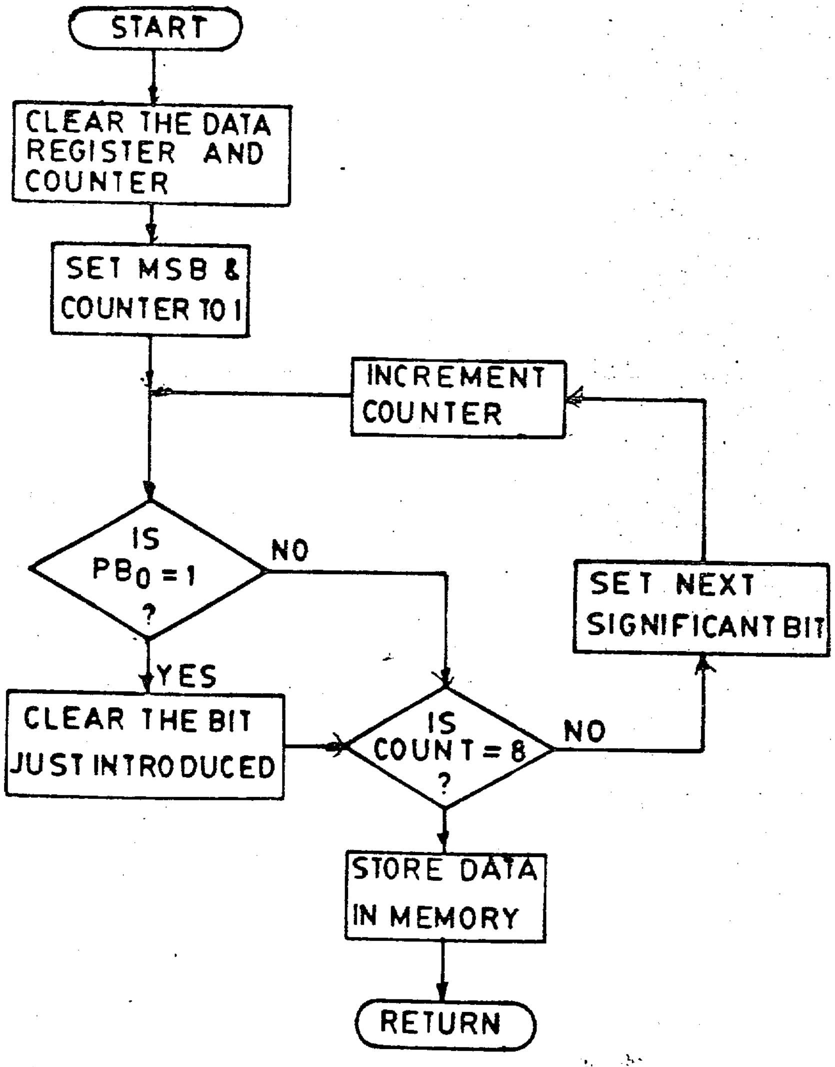
9.DATA SHEET BEL 1895,Bharat Electronics Limited Bangalore.

1Ø.RADIO CASSETTE RECORDER,S.Das Gupta, Electronics For You,April 1986,pp79-86

APPENDIX. A

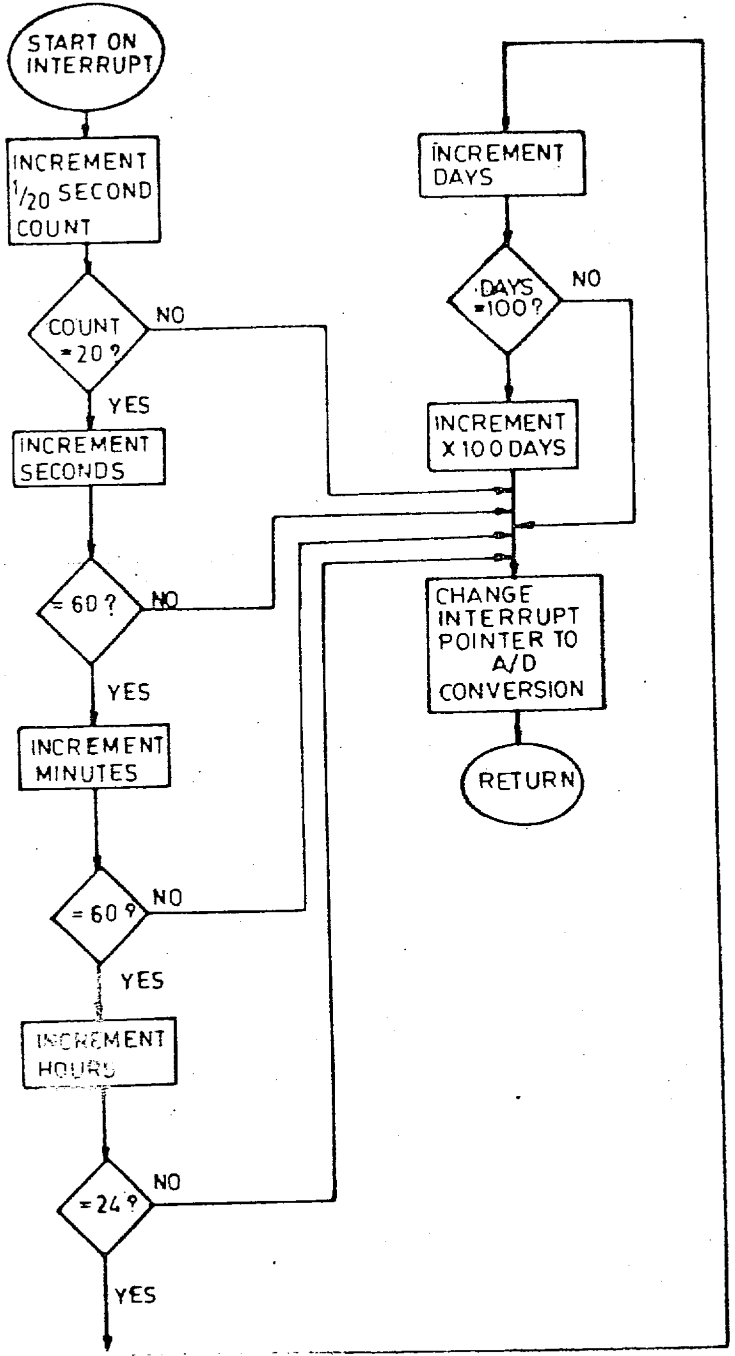


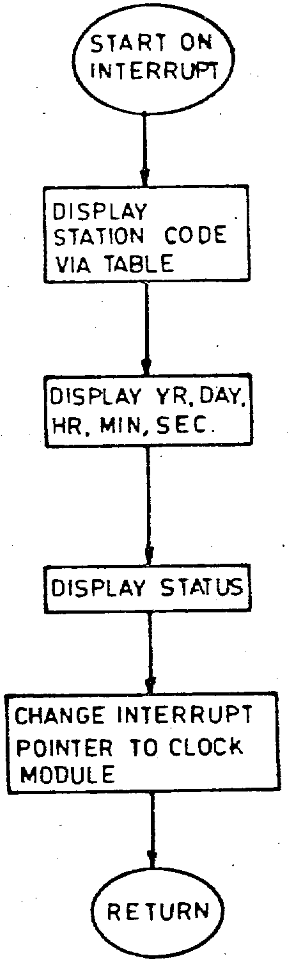
MAIN MODULE



A/D CONVERTER

APPENDIX B


START ON INTERRUPT
INCREMENT 1/20 SECOND COUNT
COUNT =20?
NO
YES
INCREMENT SECONDS
=60?
NO
YES
INCREMENT MINUTES
=60?
NO
YES
INCREMENT HOURS
=24?
NO
YES
INCREMENT DAYS
DAYS =100?
NO
INCREMENT X 100 DAYS
CHANGE INTERRUPT POINTER TO A/D CONVERSION
RETURN


START ON INTERRUPT
DISPLAY STATION CODE VIA TABLE
DISPLAY YR, DAY, HR, MIN, SEC.
DISPLAY STATUS
CHANGE INTERRUPT POINTER TO CLOCK MODULE
RETURN

APPENDIX C

SEVEN SEGMENT ASCII DISPLAY CODES

1.	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
20	00		BL	30	F3	0	0	40	35	@	⌐	50	37	P	P
21	28	!	!	31	60	1	1	41	77	A	A	51	CD	Q	o.
22	22	"	''	32	B5	2	2	42	C7	B	b	52	05	R	r
23	00	#	BL	33	F4	3	3	43	93	C	C	53	D6	S	S
24	00	$	BL	34	66	4	4	24	ES	D	d	54	87	T	t
25	00	%	BL	35	D6	5	5	45	B7	E	E	55	E3	U	U
26	00	&	BL	36	D7	6	6	46	17	F	F	56	C1	V	u
27	20	'	'	37	70	7	7	47	D3	G	G	57	C1	W	u
28	93	(	[	38	F7	8	8	48	47	H	h	58	25	X	⌐
29	F0	)	]	39	76	9	9	49	40	I	I	59	E6	Y	y
2A	00	*	BL	3A	77	A	A	4A	E1	J	J	5A	B5	Z	Z
2B	64	+	⊣	3B	C7	B	b	4B	67	K	H	5B	93	[	[
2C	08	,	.	3C	93	C	C	4C	83	L	L	5C	46	\	ч
2D	04	–	–	3D	E5	D	d	4D	45	M	∩	5D	F0	]	]
2E	08	.	.	3E	B7	E	E	4E	75	N	n	5E	32	⌒	∩
2F	21	/	⌐	3F	17	F	F	4F	C3	0	o	5F	80	'	'

1. ASCII CODE 2. DISPLAY CODE

3. ASCII CHARACTER 4. DISPLAYED CHARACTER

NOTE: INPUT CODES 3A TO 3F ARE USED FOR HEX [NON ASCII]

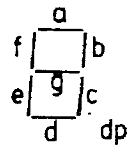
APPENDIX. D

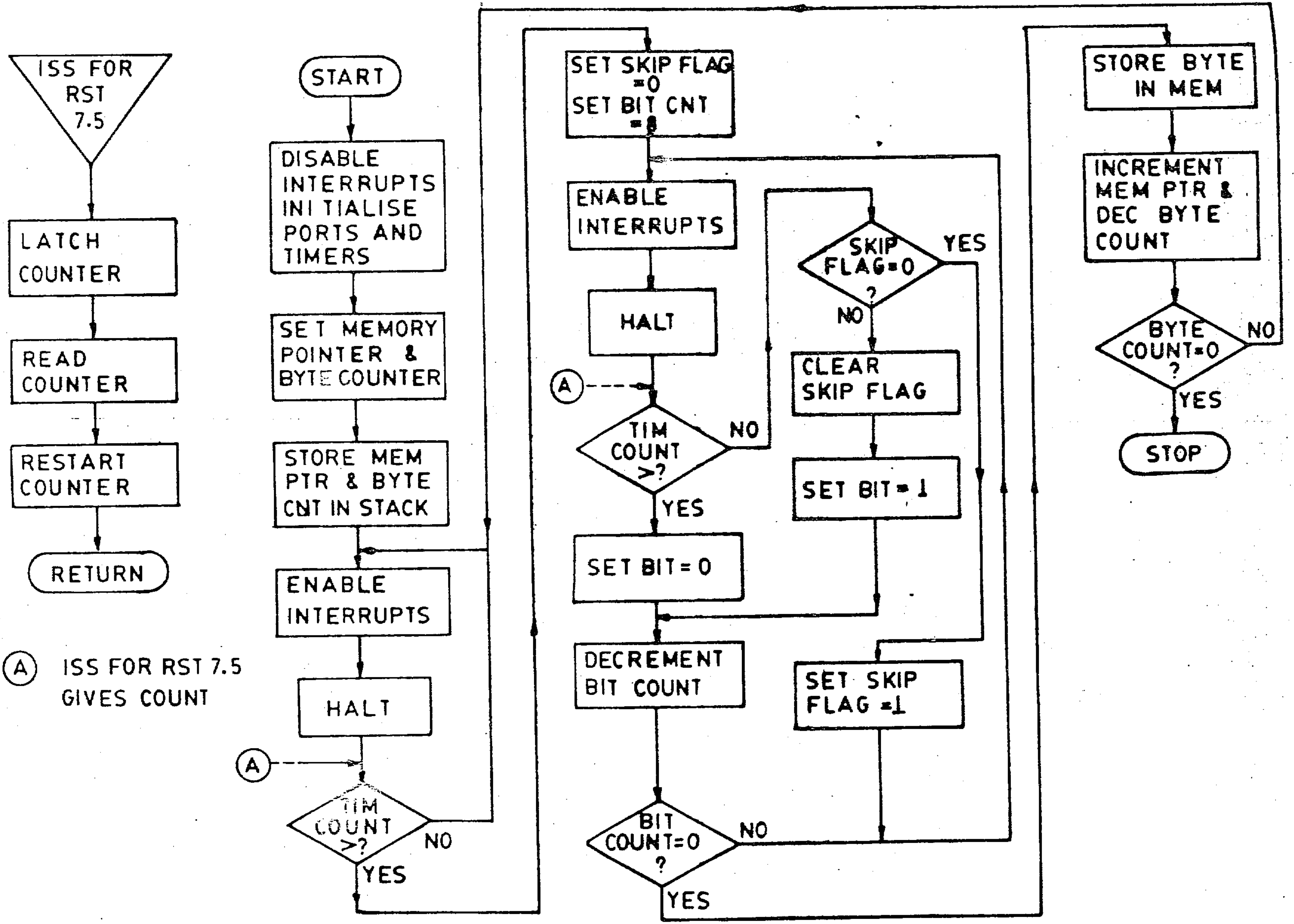
DISPLAY DATA FORMAT

8279 DISPLAY RAM OUTPUTS	A3	A2	A1	A0	B3	B2	B1	B0
DATA BITS	D7	D6	D5	D4	D3	D2	DI	D0
SEGMENTS	d	c	b	a	d.P.	g	f	e

'1' - LED is OFF

'0' - LED is ON





APPEDIX E BI Ø M DECODER FLOW CHART

APPENDIX F

STEP BY STEP OPERATION

Information regarding the current time ,and station code has to be initially loaded in the microprocessor memory starting from the address 20A2h.Seconds ,minutes,hours,units and tens of day of the year ,hundreds of the day of the year, a leading zero and lastly the last two digits of the year are stored in the RAM in this sequence.ASCII codes of the three prominent letters of the station name are entered after the year data.This information gets displayed on the seven segment displays after the program starts running.Clock is updated automatically by the program.All the information is lost on power down and initialisation needs to be done again all over.

Contents of Display RAM to be loaded at the start of the recording.

ADDRESS	CONTENTS
2ØA2H	SECONDS
2ØA3H	MINUTES
2ØA4H	HOURS
2ØA5H	UNITS AND TENS OF DAY OF THE YEAR
2ØA6H	HUNDREDS OF DAY OF THE YEAR WITH LEADING ZERO
2ØA7H	YEAR
2ØA8H	ASCII 1, STATION CODE

20A9H	ASCII 2, STATION CODE
20AAH	ASCII 3, STATION CODE

After loading this information in RAM ,copy the program by: GO 3C80.Now the instrument can start functioning by: GO 12A0.

DATA READ AND TRANSFER TO PC

The read subroutine is located at 1400H.One block of data can be read in the following way:

```
GO 3C80          ;copy the  software in RAM
GO 1400          ;REWIND THE TAPE: PRESS PLAY BUTTON
```

Now one block of data will get stored in RAM at 4000-5FFF and the header information i.e. time, station code etc. would be displayed .The control returns to monitor and more blocks can be read in the same way.

Transfer of a block of data to a floppy disk would require insertion of DOS diskette in the PC ,followed by terminal emulation software.As soon as terminal emulation software takes control,the baud rate of the COMM port is set to 1200 bps.Now the switch at the back of the instrument labelled TTY/KBD is flipped over to TTY and reset button pressed.The instrument responds by giving Signon message on the PC screen.

Now the tape recorder is prepared for data read operation ,tape is inserted and software copy as well as data read commands given as explained before.For data transfer to the PC, the program at 16C0 is used. GO 16C0 command is issued but the final CR is withheld.The file transfer is initiated by specifying the drive number filename etc.For example

a: scint.dat would store the block under filename scint1.

Now the CR should be given to start data transfer.The data is simultaneously displayed on the screen.Data bytes are split in two ASCII characters and stored in ASCII form.End of data is indicated by signon message.The transfer operation can be terminated at this stage.More blocks may be read and transferred to floppy or hard disk in the same way.

APPENDIX G

Temporary Storage;

Stack	2ØAØ	
1/2Ø seconds	2ØA1	
Seconds	2ØA2	
Minutes	2ØA3	
Hours	2ØA4	
Days	2ØA5	
Days*1ØØ	2ØA6	
Year	2ØA7	
Station code I	2ØA8	
Station code II	2ØA9	
Station code III	2ØAA	
Optional	2ØAB	
Data Pointer	2ØAC,2ØAD	(Number of bytes stored)
Memory Pointer	2ØAE,2ØAF	(Memory address where the new data is to be stored)
Status	2ØB3	
Lowaddress of Interrupt routine	2ØB4	

Header constitutes data from locations 2ØAB to 2ØA2 stored in reverse order followed by six ØØ's.

Memory Map;

ØØØØ-ØFFF	Monitor
1ØØØ-17FF	Run Location for Recorder Program
18ØØ-1FFF	Keyboard/Display

2ØØØ-2ØFF	8155 RAM
28ØØ-37FF	Not used
38ØØ-3FFF	Recorder Program in EPROM
4ØØØ-5FFF	Buffer I
6ØØØ-7FFF	Buffer II
8ØØØ-FFFF	Not used

I/O Map and port assignments;

ØØH		8251	
Ø8H		8253	
	counter Ø,	/1Ø	
	counter 1,	/256Ø	6Ø PPS:Rate
	counter 2,	/2	76.8 Khz:Square wave
2ØH		8155	
	port A	D/A Converter	
	port B,PBØ	Comparator, rest not used	
	port C:		
PCØ		Channel select	
PC1		Channel select,4 inputs	
PC2		Write/Read control,RST 7.5	
PC3		Motor On,Cassette Recorder	
PC4		MUX Enable	
PC5		Free	

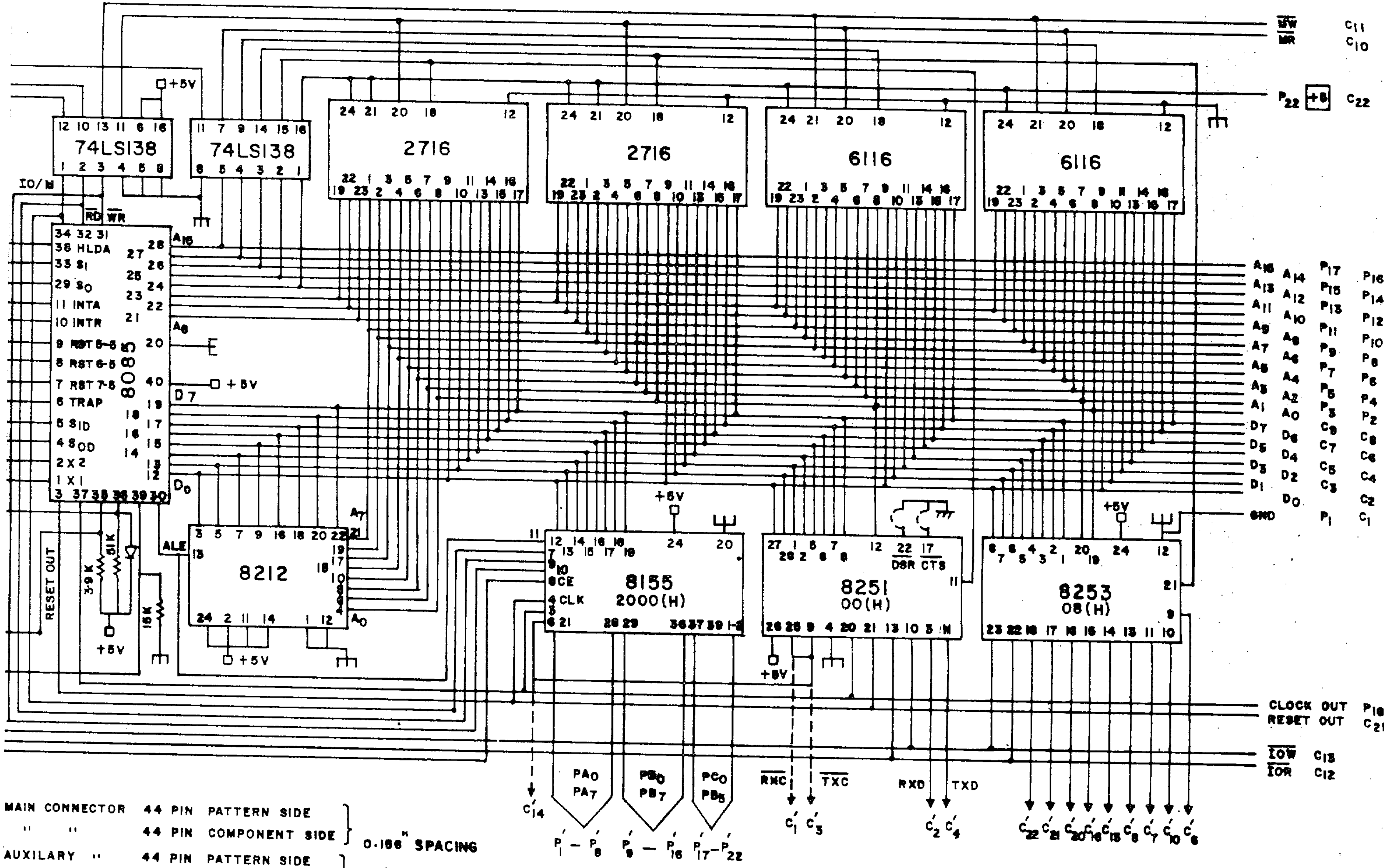
APPENDIX H

BIT Assignments of the STATUS byte

Bit Ø/Bit 1	Channel select
Bit 2	Write Buffer A, Ø Empty, 1 Full
Bit 3	Write Buffer B, do
Bit 4	T/R Motor ON,Ø OFF,1 ON
Bit 5	Data recording,Ø off ,1 on
Bit 6	Buffer A, Ø Empty,1 Full
Bit 7	Buffer B, Ø Empty,1 Full

STATUS LED'S

LSB							MSB
X	X	X	X	X	X	X	X
CHANNEL		WRITE	WRITE	T/R	DATA	BUFF A	BUFF B
SELECT		BUFF A	BUFF B	MOTOR	REC	FULL	FULL
				ON	ON		

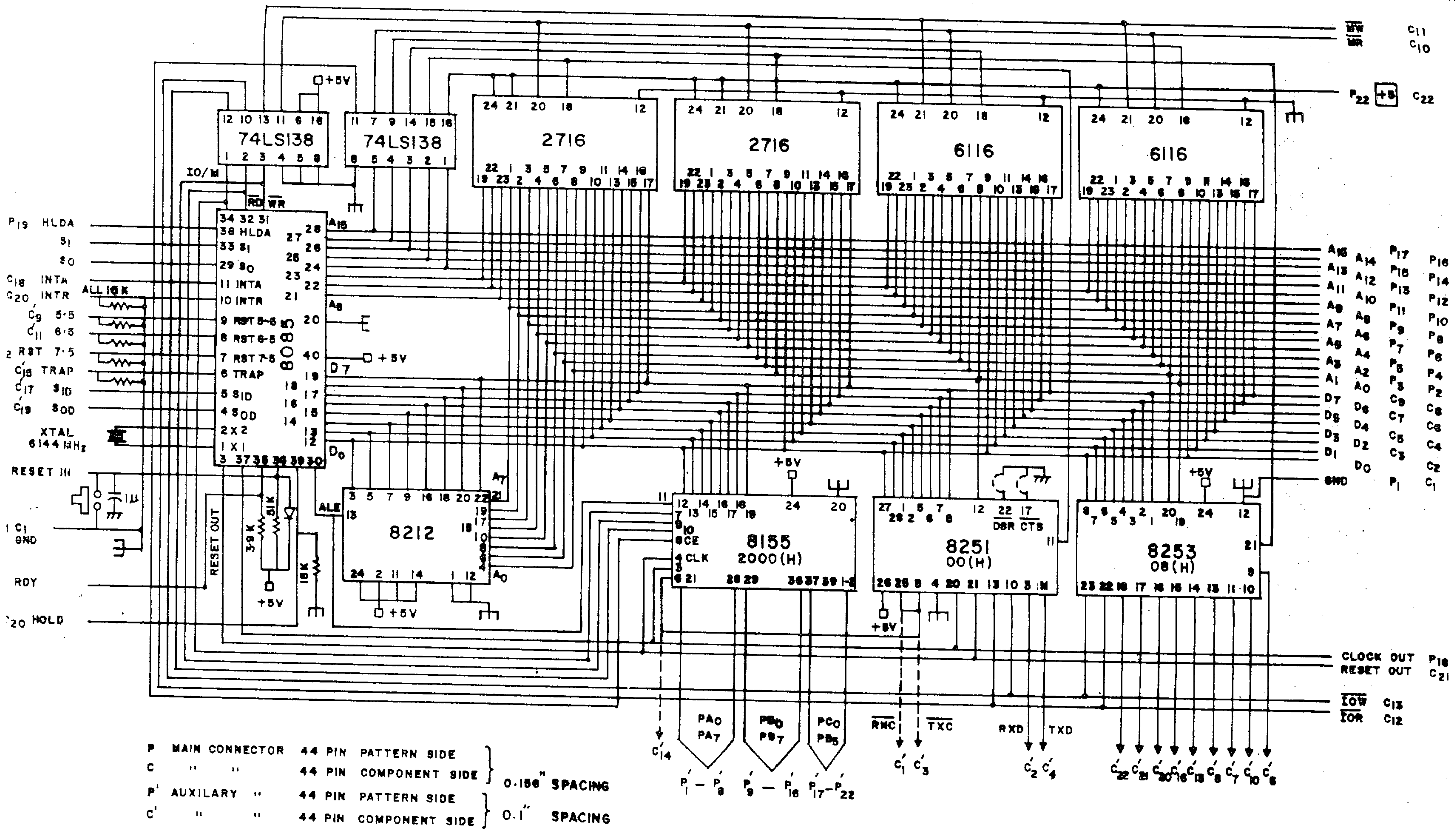


MAIN CONNECTOR 44 PIN PATTERN SIDE
" " 44 PIN COMPONENT SIDE } 0.156" SPACING

AUXILARY " 44 PIN PATTERN SIDE
" " 44 PIN COMPONENT SIDE } 0.1" SPACING

RNM / ITK - 9/85

8085 A BASED SINGLE CARD MICROCOMPUTER



P MAIN CONNECTOR 44 PIN PATTERN SIDE
C " " 44 PIN COMPONENT SIDE } 0.156" SPACING

P' AUXILARY " 44 PIN PATTERN SIDE
C' " " 44 PIN COMPONENT SIDE } 0.1" SPACING

RNM / ITK - 9/85

APPENDIX J

Addresses of important subroutines(hex)

1.Keyboard monitor	Ø04E
2.Serial monitor	Ø3FA
3.Time delay via DE	Ø5F1
4.Byte write	1Ø39/156Ø
5.Byte read	1Ø5Ø/15CØ
6.Block write ,HL mem pointer,BC byte count	1ØBØ
7.Leader/Trailer,128 ØØ's& 128 FF's	1ØCØ
8.Two second delay	1ØF8
9.Clock	11ØØ
1Ø.A/D converter calling location	116Ø
11.A/D converter subroutine	165Ø
12.Motor on	11EØ
13.Motor off	11FØ
14.Main program for recording	121Ø
15.RST 7.5 vector,A/D converter	127Ø
16. do ,clock	128Ø
17. do ,display	129Ø
18.Entry point for main program	12AØ
19.Copy header	12D8
2Ø.RST 7.5 ISS header	12F5
21.Disply packed BCD	13ØØ
22.Split nibble	131C
23.Display alphanumeric	134Ø
24.Initialize ports and timers	136Ø

25.Start record mode	1391
26.Start read mode	1398
27.Main call for initialisation	13AØ
28.Code table for alphanumeric data for display on seven segment LED's	13CØ-13FF
29.Read one block	14ØØ/16ØØ
3Ø.Copy header ,used by read	145Ø
31.Copy self to 1ØØØ ,runs at 3C8Ø	148Ø
32.Write sync and byte count,called by record	14AØ
33.Loop FF	14CØ
34.Checksum generation in Accumulator	14EØ
35.Start recording	15ØØ
36.Initialise 8251	153Ø
37.Wait char used by 8251	156Ø
38.Initialise for read	157Ø
39.RST 7.5 ISS location for read operation	15AØ
4Ø.Byte read	15CØ
41.Block read and display	16ØØ
42.A/D converter	165Ø
43.Test D/A converter, generate ramp	168Ø
44.Test count between pulses in read mode data stored at 4ØØØ, 2ØØØ values	169Ø
45.Data Transfer Program	16CØ

APPENDIX K

SYSTEM SPECIFICATIONS

1.No. of channels	one to four
2.Input range	0-5 Volts
3.Sampling rate (total)	20 per sec
4.Data recording medium	3.8 mm tape (audio cassette)
5.Recording density	230 char per in
6.Tape speed	47.6 mm per sec
7.Number of tracks	one per side
8.Interrecord gap	95 mm approx
9.Recording mode and format	1 start bit , 8 bit serial data, LSB first,1 stop bit , no parity
10.Recording rate	4800 bps
11.Data block size	8096 bytes
12,Data volume on one C-60 cassette	1.5 Mbytes
13.Power source	230V AC/12V Battery
14.Power consumption	15 Watts
15.Size WXDXH (mm)	305X380X76
16.Weight (excluding tape recorder)	3.3 Kg.

Note: The recording rate may be reduced to 2400 bps to reduce the number of errors.In that case the volume of data on C-60 cassette also drops to 800 Kbytes.It would require the following changes in the software:

address,	1364	3E08;divide by 8 to get 38.4 KHz
	15C5	3E56;detection threshold for 1
	15D3	3E56;or 0

APPENDIX L

PROGRAM LISTING

```
1Ø39     C36Ø15    JMP 156Ø     ;BYTE WRITE

1Ø5Ø     C3CØ15    JMP 15CØ     ;BYTE READ
```

WRITE BYTE IN E REG REPEAT AS IN D REG

```
1ØAØ     168Ø      MVI D,8Ø     ;128 TIMES
1ØA2     4B        MOV C,E      ;WRITE VIA C REG
1ØA3     D5        PUSH D
1ØA4     CD391Ø    CALL 1Ø39    ;BYTE WRITE FROM C REG
1ØA7     D1        POP D
1ØA8     15        DCR D
1ØA9     C2A21Ø    JNZ 1ØA2     ;WRITE ALL BYTES
1ØAC     C9        RET
```

BLOCK WRITE,HL MEM POINTER,BC BYTE COUNT

```
1ØBØ     C5        PUSH B
1ØB1     4E        MOV C,M
1ØB2     CD391Ø    CALL 1Ø39    ;CALL BYTE WRITE
1ØB5     23        INX H
1ØB6     C1        POP B
1ØB7     ØB        DCX B
1ØB8     78        MOV A,B
1ØB9     B1        ORA C
1ØBA     C2BØ1Ø    JNZ 1ØBØ     ;REPEAT TILL
1ØBD     C9        RET          ;ALL BYTES OVER
```

RECORD LEADER,128 ØØ'S AND 128 FF'S

```
1ØCØ     C5        PUSH B
1ØC1     1EØØ      MVI E,ØØ     ;ØØ ,128 TIMES
1ØC3     CDAØ1Ø    CALL 1ØAØ
1ØC6     1EFF      MVI E,FF     ;THEN FF ALSO
1ØC8     CDAØ1Ø    CALL 1ØAØ
1ØCB     C1        POP B
1ØCC     C9        RET
```

8 SECOND DELAY VIA DOUBLE LOOP

```
1ØDØ     D5        PUSH D
1ØD1     111ØØØ    LXI D,ØØ1Ø   ;OUTER COUNT LOOP
1ØD4     D5        PUSH D
1ØD5     11FFFF    LXI D,FFFF   ;INNER COUNT LOOP
1ØD8     CDF1Ø5    CALL Ø5F1    ;MONITOR UTILITY
1ØDB     D1        POP D
1ØDC     1B        DCX D
1ØDD     7A        MOV A,D
1ØDE     B3        ORA E
1ØDF     C2D41Ø    JNZ 1ØD4     ;OUTER LOOP
1ØE2     D1        POP D
1ØE3     C9        RET
```

BUFFER CHANGEOVER A TO B

```
1ØE4     3AB32Ø    LDA 2ØB3     ;STATUS BYTE
1ØE7     F68Ø      ORI 8Ø       ;BIT 7=1
1ØE9     32B32Ø    STA 2ØB3     ;STORE BACK
1ØEC     11FØ1F    LXI D,1FFØ   ;BYTE COUNT 2ØØØ-1ØH
1ØEF     CDDB12    CALL 12DB    ;COPY STATION CODE TIME
1ØF2     C9        RET
```

2 SECONDS DELAY

```
1ØF8     D5        PUSH D
1ØF9     11Ø4ØØ    LXI D,ØØØ4
1ØFC     C3D41Ø    JMP 1ØD4     ;REST IS PART OF 8 SEC
```

CLOCK SUBROUTINE

```
11ØØ     21AØ2Ø    LXI H,2ØAØ
11Ø3     CD2511    CALL 1125
11Ø6     FE2Ø      CPI 2Ø
11Ø8     CØ        RNZ
11Ø9     CD2511    CALL 1125
11ØC     FE6Ø      CPI 6Ø
11ØE     CØ        RNZ
11ØF     CD2511    CALL 1125
1112     FE6Ø      CPI 6Ø
1114     CØ        RNZ
1115     CD2511    CALL 1125
1118     FE24      CPI 24
111A     CØ        RNZ
111B     CD2511    CALL 1125
111E     E6FF      ANI FF
112Ø     CØ        RNZ
1121     CD2511    CALL 1125
1124     C9        RET
```

LOAD ZERO AND ADVANCE MEM POINTER ,USED BY CLOCK

```
1125     36ØØ      MVI M,ØØ
1127     23        INX H
1128     34        INR M
1129     7E        MOV A,M
112A     37        STC
112B     3F        CMC
112C     27        DAA
112D     77        MOV M,A
112E     C9        RET
```

DISPLAY SUBROUTINE

```
1130    E5        PUSH H
1131    C5        PUSH B
1132    F5        PUSH PSW
1133    21A820    LXI H,20A8   ;STN CODE
1136    Ø6Ø3      MVI B,Ø3     ;3 BYTES ASCII
1138    3E9Ø      MVI A,9Ø     ;CODE 827
113A    32ØØ19    STA 19ØØ     ;CNTRL 8279
113D    CD4Ø13    CALL 134Ø    ;CALL DISPLAY2
114Ø    3E93      MVI A,93     ;NUMERIC DATA
1142    32ØØ19    STA 19ØØ     ;
1145    21A72Ø    LXI H,20A7   ;YEAR DAY ETC
1148    Ø6Ø6      MVI B,Ø6     ;SIX BYTES
114A    CDØØ13    CALL 13ØØ    ;CALL DISPLAY1
114D    3E9F      MVI A,9F     ;STATUS,8 LED'S
114F    32ØØ19    STA 19ØØ     ;
1152    3AB32Ø    LDA 2ØB3     ;GET STATUS
1155    2F        CMA          ;COMPLEMENT
1156    32ØØ18    STA 18ØØ     ;DISPLAY
1159    F1        POP PSW
115A    C1        POP B
115B    E1        POP H
115C    C9        RET
```

A/D CONVERTER

```
116Ø    E5        PUSH H
1161    D5        PUSH D
1162    DB23      IN 23
1164    E6EF      ANI EF       ;ENABLE MUX
1166    D323      OUT 23
1168    2AAC2Ø    LHLD 2ØAC    ;BYTE COUNT
116B    EB        XCHG         ;IN DE
116C    2AAE2Ø    LHLD 2ØAE    ;BUFFER ADDR IN HL
116F    CD5Ø16    CALL 165Ø    ;CALL  CONVERT
1172    23        INX H        ;ADVANCE MEM POINTER
1173    1B        DCX D        ;BYTE COUNT
1174    7A        MOV A,D      ;TEST COUNT
1175    B3        ORA E        ;JUMP IF
1176    CC9311    CZ 1193      ;BUFFER FULL
1179    22AE2Ø    SHLD 2ØAE    ;ELSE STORE BACK
117C    EB        XCHG         ;COUNT AND MEM
117D    22AC2Ø    SHLD 2ØAC    ;POINTER
118Ø    D1        POP D
1181    E1        POP H
1182    CDBØ11    CALL 11BØ    ;CHANNEL CHANGEOVER
1185    CDC311    CALL 11C3    ;MUX CONTROL
1188    DB23      IN 23
118A    F61Ø      ORI 1Ø       ;DISABLE MUX
118C    D323      OUT 23 .
118E    C9        RET
```

BUFFER FULL

```
1193    7C        MOV A,H
1194    FE6Ø      CPI 6Ø       ;H =6ØH ? IF YES
1196    CAE41Ø    JZ 1ØE4      ;BUFF 1 FULL
1199    264Ø      MVI H,4Ø     ;ELSE BUFF 2 FULL
119B    3AB32Ø    LDA 2ØB3     ;CHANGE TO BUFF 1
119E    F64Ø      ORI 4Ø       ;STATUS BYTE UPDATE
11AØ    32B32Ø    STA 2ØB3
11A3    11FØ1F    LXI D,1FFØ   ;NEW COUNT 2K-16 BYTE
11A6    CDDB12    CALL 12DB    ;COPY HEADER
11A9    C9        RET
```

CHANNEL CHANGEOVER

```
11BØ    C5        PUSH B
11B1    3AB32Ø    LDA 2ØB3     ;STATUS BYTE
11B4    4F        MOV C,A      ;SAVE IN C
11B5    3C        INR A        ;NEXT CHANNEL
11B6    E6Ø3      ANI Ø3       ;RETAIN LO 2 BITS
11B8    47        MOV B,A      ;SAVE IN B
11B9    79        MOV A,C      ;RECALL OLD STATUS
11BA    E6FC      ANI FC       ;SUPPRESS LO 2
11BC    BØ        ORA B        ;PUT NEW IN PLACE
11BD    32B32Ø    STA 2ØB3     ;NEW STATUS
11CØ    C1        POP B
11C1    C9        RET
```

MUX CONTROL

```
11C3    C5        PUSH B
11C4    DB23      IN 23
11C6    E6FC      ANI FC       ;SUPPRESS OLD LO BITS
11C8    4F        MOV C,A      ;SAVE IN C
11C9    3AB32Ø    LDA 2ØB3     ;GET STATUS
11CC    E6Ø3      ANI Ø3       ;LO BITS FOR MUX
11CE    B1        ORA C        ;UPDATE
11CF    D323      OUT 23       ;NEW CHANNEL
11D1    C1        POP B
11D2    C9        RET
```

START PULSE

```
11D3    DB23      IN 23
11D5    F62Ø      ORI 2Ø
11D7    D323      OUT 23
11D9    DB23      IN 23
11DB    E61F      ANI 1F
11DD    D323      OUT 23
11DF    C9        RET
```

MOTOR ON

```
11EØ    DB23      IN 23
11E2    F6Ø8      ORI Ø8       ;PC3 MADE ONE
11E4    D323      OUT 23
11E6    3AB32Ø    LDA 2ØB3     ;UPDATE STATUS
11E9    F61Ø      ORI 1Ø
```

```
11EB     32B32Ø    STA 2ØB3
11EE     C9        RET
```

MOTOR OFF

```
11FØ     DB23      IN 23
11F2     E6F7      ANI F7       ;PC3 =ZERO
11F4     D323      OUT 23
11F6     3AB32Ø    LDA 2ØB3     ;UPDATE STATUS
11F9     E6EF      ANI EF
11FB     32B32Ø    STA 2ØB3
11FE     C9        RET
```

UPDATE AND DISPLAY MOTOR STATUS

```
12ØØ     3AB32Ø    LDA 2ØB3     ;STATUS BYTE
12Ø3     E61Ø      ANI 1Ø       ;UPDATE
12Ø5     CA3Ø11    JZ 113Ø      ;DISPLAY
12Ø8     C9        RET
```

MAIN PROGRAM FOR RECORDING

```
121Ø     3AB32Ø    LDA 2ØB3     ;STATUS BYTE
1213     F62Ø      ORI 2Ø       ;RECORDING ON
1215     32B32Ø    STA 2ØB3     ;STORE BACK
1218     E6Ø4      ANI Ø4       ;TST BIT2
121A     11ØØ2Ø    LXI D,2ØØØ   ;BLOCK SIZE
121D     CA2212    JZ 1222      ;BUF1 FULL THEN JMP
122Ø     16ØØ      MVI D,ØØ     ;
1222     21ØØ4Ø    LXI H,4ØØØ   ;MEM POINTER
1225     19        DAD D        ;ADD DE FOR NEW ADDR
1226     Ø1ØØ2Ø    LXI B,2ØØØ   ;BYTE COUNT
1229     CDØØ15    CALL 15ØØ    ;CALL RECORD BLOCK
122C     3AB32Ø    LDA 2ØB3     ;STATUS
122F     E6DF      ANI DF       ;BIT 6=Ø
1231     32B32Ø    STA 2ØB3     ;STORE BACK
1234     7C        MOV A,H      ;
1235     FE6Ø      CPI 6Ø       ;H=6Ø? HL =BUFF2
1237     C24412    JNZ 1244     ;ELSE JUMP OVER
123A     3AB32Ø    LDA 2ØB3     ;STATUS
123D     E673      ANI 73       ;BITS Ø,1,5,6,7
123F     F6Ø8      ORI Ø8       ;BIT 3 =1
1241     C35Ø12    JMP 125Ø     ;JUMPOVER
```

BUFFER 2IN USE

```
1244     FE8Ø      CPI 8Ø       ;H=8Ø
1246     C211Ø6    JNZ Ø611     ;IF NOT  SHOW ERR
1249     3AB32Ø    LDA 2ØB3     ;STATUS BYTE
124C     E6B3      ANI B3       ;BITS Ø,1,4,5,7
124E     F6Ø4      ORI Ø4       ;BIT 2=1
125Ø     32B32Ø    STA 2ØB3     ;STORE STATUS
1253     C3C312    JMP 12C3     ;JUMP TEST BUFF
                                 AND RECORD
```

RST 7.5 VECTOR ,A/D CONVERTER

```
127Ø     CD6Ø11    CALL 116Ø    ;CALL A/D CONVERT
1273     3E8Ø      MVI A,8Ø     ;LO ADDRESS FOR CLOCK
1275     32B42Ø    STA 2ØB4     ;STORED AT RST 7.5
1278     F1        POP PSW      ;VECTOR
1279     E1        POP H
127A     D1        POP D
127B     C1        POP B
127C     FB        EI
127D     C9        RET
```

RST 7.5 VECTOR FOR CLOCK

```
1280    CDØØ11    CALL 11ØØ     ;CALL CLOCK SUBROUTINE
1283    3E9Ø      MVI A,9Ø      ;LO ADDR FOR DISPLAY
1285    32B42Ø    STA 2ØB4      ;STORED AT RST 7.5
1288    F1        POP PSW       ;VECTOR
1289    E1        POP H
128A    D1        POP D
128B    C1        POP B
128C    FB        EI
128D    C9        RET
```

RST 7.5 VECTOR FOR DISPLAY

```
129Ø    CD3Ø11    CALL 113Ø     ;CALL DISPLAY
1293    3E7Ø      MVI A,7Ø      ;LO ADDR FOR A/D
1295    32B42Ø    STA 2ØB4      ;STORED AT RST 7.5
1298    F1        POP PSW       ;VECTOR
1299    E1        POP H
129A    D1        POP D
129B    C1        POP B
129C    FB        EI
129D    C9        RET
```

MAIN PROGRAM

```
12AØ    31AØ2Ø    LXI SP,2ØAØ ;INIT STACK
12A3    21Ø47Ø    LXI H,7ØØ4  ;INITIAL CONDITION
12A6    22B32Ø    SHLD 2ØB3   ;TO STATUS BYTE
12A9    21FØ1F    LXI H,1FFØ  ;DATA BYTE COUNT
12AC    22AC2Ø    SHLD 2ØAC   ;AT 2ØAC
12AF    CDAØ13    CALL 13AØ   ;INIT PORTS TIMER
12B2    3EØB      MVI A,ØB    ;INTERRUPTS
12B4    3Ø        SIM         ;SET MASK
12B5    3E1Ø      MVI A,1Ø    ;SET RECORD MODE
12B7    D323      OUT 23      ;
12B9    21ØØ4Ø    LXI H,4ØØØ  ;BUFF 1 START
12BC    CDDB12    CALL 12DB   ;COPY HEADER
12BF    22AE2Ø    SHLD 2ØAE   ;SAVE MEM POINTER
12C2    FB        EI          ;AT 2ØAE
12C3    3AB32Ø    LDA 2ØB3    ;GET STATUS BYTE
12C6    E6CØ      ANI CØ      ;TEST FOR ANY BUFFER
12C8    CAC312    JZ 12C3     ;FULL ,LOOP TILL FOUND
12CB    C31Ø12    JMP 121Ø    ;GO FOR RECORDING
```

BLOCK WRITE,HL MEM POINETR
DE BYTE COUNTER

```
12CF    4E        MOV C,M     ;DATA IN C
12DØ    CD391Ø    CALL 1Ø39   ;BYTE WRITE
12D3    23        INX H       ;NEXT LOC
12D4    1B        DCX D       ;BYTE COUNT
12D5    7A        MOV A,D
12D6    B3        ORA E
12D7    C2CF12    JNZ 12CF    ;REPEAT ALL BYTES
12DA    C9        RET
```

COPY HEADER

```
12DB    D5        PUSH D
12DC    C5        PUSH B
12DD    11AB2Ø    LXI D,2ØAB  ;HEADER HI END
12EØ    Ø1Ø6ØA    LXI B,ØAØ6  ;1Ø CHAR,6 ZEROES
12E3    1A        LDAX D      ;GET ONE CHAR
12E4    77        MOV M,A     ;STORE
12E5    23        INX H       ;NEXT
12E6    1B        DCX D       ;NEXT LOWER BYTE
12E7    Ø5        DCR B       ;COPY 1Ø BYTES
12E8    C2E312    JNZ 12E3    ;
12EB    36ØØ      MVI M,ØØ    ;ZEROES
12ED    23        INX H       ;AT NEXT 6
12EE    ØD        DCR C       ;PLACES
12EF    C2EB12    JNZ 12EB    ;
12F2    C1        POP B
12F3    D1        POP D
12F4    C9        RET
```

RST 7.5 ISS HEADER

```
12F5    C5        PUSH B
12F6    D5        PUSH D
12F7    E5        PUSH H
12F8    F5        PUSH PSW
12F9    3AB42Ø    LDA 2ØB4     ;GET LO ADDR OF RST 7.5
12FC    6F        MOV L,A      ;VECTOR ,IN L REG
12FD    2612      MVI H,12     ;
12FF    E9        PCHL         ;VECTOR IS 12XX
```

DISPLAY BCD PACKED IN ONE BYTE

```
13ØØ    ØEØ2      MVI C,Ø2     ;
13Ø2    CD1C13    CALL 131C    ;CALL SPLIT
13Ø5    E5        PUSH H       ;
13Ø6    21F92Ø    LXI H,2ØF9   ;GET DISPLAY CODE
13Ø9    7E        MOV A,M      ;
13ØA    CD2D13    CALL 132D    ;DISPLAY 2 DIGITS
13ØD    32ØØ18    STA 18ØØ
131Ø    23        INX H        ;NEXT DATA
1311    ØD        DCR C
1312    C2Ø913    JNZ 13Ø9     ;DISPLAY AGAIN
1315    E1        POP H        ;
1316    2B        DCX H
1317    Ø5        DCR B
1318    C2ØØ13    JNZ 13ØØ
131B    C9        RET
```

SPLIT NIBBLES

```
131C    7E       MOV A,M     ;GET BYTE
131D    ØF       RRC         ;THEN HI
131E    ØF       RRC         ;NIBBLE
131F    ØF       RRC
132Ø    ØF       RRC
1321    E6ØF     ANI ØF      ;ONLY LO NIBBLE
1323    32F92Ø   STA 2ØF9    ;SAVE
1326    7E       MOV A,M     ;NEXT BYTE
1327    E6ØF     ANI ØF      ;LO NIBBLE
1329    32FA2Ø   STA 2ØFA    ;AT NEXT PLACE
132C    C9       RET
```

DISPLAY ALPHANUMERIC DAT VIA TABLE

SYMBOL TABLE AT Ø384 COPIED AT 13DØ

```
132D    E5       PUSH H      ;
132E    2184Ø3   LXI H,Ø384  ;
1331    85       ADD L       ;ADD A TO L REG
1332    6F       MOV L,A     ;
1333    7E       MOV A,M     ;GET DISPLAY DATA
1334    E1       POP H       ;CORRESSPNDING TO BYTE
1335    2F       CMA         ;COMPLEMENT TO DISPLAY
1336    C9       RET
```

DISPLAY ALPHANUMERIC DATA

```
134Ø    3E9Ø     MVI A,9Ø    ;CODE FOR 8279
1342    32ØØ19   STA 19ØØ    ;
1345    ØØ       NOP
1346    ØØ       NOP
1347    ØØ       NOP
1348    D5       PUSH D
1349    5E       MOV E,M     ;DATA IN E REG
134A    16ØØ     MVI D,ØØ    ;INIT D
134C    E5       PUSH H      ;SAVE HL
134D    21AØ13   LXI H,13AØ  ;HL POINTS TO TABLE
135Ø    19       DAD D       ;FOR DISPLAY ON 7 SEG
1351    7E       MOV A,M     ;GET TABLE ENTRY FOR
1352    2F       CMA         ;ASCII CODE &
1353    E1       POP H       ;COMPLEMENT
1354    ØØ       NOP
1355    32ØØ18   STA 18ØØ    ;DISPLAY REG
1358    23       INX H       ;NEXT CHAR
1359    Ø5       DCR B       ;COUNT CHAR
135A    C24913   JNZ 1349    ;DISPLAY ALL CHAR
135D    D1       POP D
135E    C9       RET
```

INITIALISATION OF PORTS AND TIMERS

```
136Ø    3E3E     MVI A,3E    ;CNTRØ,MODE3,LSB FIRST
1362    D3ØB     OUT ØB      ;
1364    3EØ4     MVI A,Ø4    ;DIVIDE BY 4
1366    D3Ø8     OUT Ø8      ;OUT PUT 76.8 KHZ
1368    3EØØ     MVI A,ØØ
136A    D3Ø8     OUT Ø8
136C    3E7C     MVI A,7C    ;CNTR1,MODE2,LSB FIRST
136E    D3ØB     OUT ØB
137Ø    3EØØ     MVI A,ØØ
1372    D3Ø9     OUT Ø9      ;DIVIDE BY 14ØØH,512Ø
1374    3E14     MVI A,14    ; DECIMAL,OUTPUT 6Ø PPS
1376    D3Ø9     OUT Ø9      ;CNTR2,MODE3
1378    3EBE     MVI A,BE
137A    D3ØB     OUT ØB      ;DIVIDE BY 6ØØH
137C    3EØØ     MVI A,ØØ    ;1536 DECIMAL:NOT USED
137E    D3ØA     OUT ØA
138Ø    3EØ6     MVI A,Ø6
1382    D3ØA     OUT ØA      ;CNTR 8155
1384    3E4Ø     MVI A,4Ø    ;DIVIDE BY 1Ø,LOAD MSB ØØ
1386    D325     OUT 25      ;LSB OAH
1388    3EØA     MVI A,ØA
138A    D324     OUT 24      ;START
138C    3ECD     MVI A,CD
138E    D32Ø     OUT 2Ø
139Ø    C9       RET
```

RECORD MODE

```
1391    DB23     IN 23
1393    E63B     ANI 3B      ;BIT2 C PORT=Ø
1395    D323     OUT 23
1397    C9       RET
```

READ MODE

```
1398    DB23     IN 23
139A    F6Ø4     ORI Ø4      ;BIT2 C PORT=1
139C    D323     OUT 23
139E    C9       RET
```

INITIALISATION SUBROUTINES CALLED HERE

```
13AØ    CD6Ø13   CALL 136Ø   ;PORTS AND TIMERS
13A3    CD9113   CALL 1391   ;RECORD MODE
13A6    3EØ8     MVI A,Ø8    ;DISPLAY CODE
13A8    32ØØ19   STA 19ØØ    ;TO 8279
13AB    CD3Ø15   CALL 153Ø   ;INITIALISE 8251
13AE    C9       RET
```

CODE TABLE TO DISPLAY ALPHANUMERIC DATA ON
7 SEGMENT DISPLAYS

```
13CØ ØØ 28 22 ØØ ØØ ØØ ØØ 20 93 FØ ØØ 84 Ø8 Ø4 Ø8 21
13DØ F3 6Ø B5 F4 66 D6 D7 70 F7 76 77 C7 93 E5 B7 17
13EØ 35 77 C7 93 E5 97 17 D3 47 4Ø E1 67 83 45 45 C5
13FØ 37 CD Ø5 D6 87 E3 C1 C1 25 E6 B5 93 46 FØ 32 8Ø
```

READ SUBROUTINES

```
14ØØ      C3ØØ16     JMP 16ØØ     ;NEW READ AT 16ØØH
```

PART OF OLD RED PROGRAM,NOT USED

```
14Ø3      CDAØ13     CALL 13AØ
14Ø6      CDEØ11     CALL 11EØ
14Ø9      CDF81Ø     CALL 1ØF8
14ØC      21ØØ3Ø     LXI H,3ØØØ
14ØF      C5         PUSH B
141Ø      CDCØ14     CALL 14CØ
1413      FEA5       CPI A5
1415      CA2Ø14     JZ 142Ø
1418      CD5Ø1Ø     CALL 1Ø5Ø
141B      C31314     JMP 1413

141E      ØØ         NOP
141F      ØØ         NOP
142Ø      CD5Ø1Ø     CALL 1Ø5Ø
1423      4F         MOV C,A
1424      C5         PUSH B
1425      CD5Ø1Ø     CALL 1Ø5Ø
1428      C1         POP B
1429      47         MOV B,A
142A      C5         PUSH B
142B      CD5Ø1Ø     CALL 1Ø5Ø
142E      77         MOV M,A
142F      23         INX H
143Ø      C1         POP B
1431      ØB         DCX B
1432      78         MOV A,B
1433      B1         ORA C
1434      C22A14     JNZ 142A
1437      CD5Ø1Ø     CALL 1Ø5Ø
143A      32B62Ø     STA 2ØB6
143D      CDFØ11     CALL 11FØ
144Ø      CD5Ø14     CALL 145Ø
1443      CD3Ø11     CALL 113Ø
1446      CF         RST 1
```

COPY HEADER

```
145Ø      C5         PUSH B
1451      21ØØ4Ø     LXI H,4ØØØ   ;MEM POINTER
1454      11AB2Ø     LXI D,2ØAB   ;HEADER DESTINATION
1457      Ø6ØA       MVI B,ØA     ;BYTE COUNT
1459      7E         MOV A,M      ;MOVE BYTE FROM MEM
145A      12         STAX D       ;VIA HL TO DE
145B      23         INX H
145C      1B         DCX D
145D      Ø5         DCR B        ;TILL 1Ø BYTES OVER
145E      C25914     JNZ 1459
1461      C1         POP B
1462      C9         RET
```

READ BLOCK,NOT USED

```
147Ø      C5         PUSH B
1471      CD5Ø1Ø     CALL 1Ø5Ø
1474      77         MOV M,A
1475      23         INX H
1476      ØØ         NOP
1477      C1         POP B
1478      ØB         DCX B
1479      78         MOV A,B
147A      B1         ORA C
147B      C27Ø2Ø     JNZ 2Ø7Ø
147E      C9         RET
```

COPY PROGRAM TO 1ØØØH,IT WORKS AT 3C8ØH

```
148Ø      31AØ2Ø     LXI SP,2ØAØ ;INIT STACK
1483      21ØØ38     LXI H,38ØØ  ; SOURCE START
1486      11ØØ1Ø     LXI D,1ØØØ  ;DESTINATION
1489      Ø1FFØ7     LXI B,Ø7FF  ;BYTE COUNT
148C      7E         MOV A,M     ;MOVE
148D      12         STAX D      ;DATA TO NEW LOC
148E      23         INX H
148F      13         INX D
149Ø      ØB         DCX B
1491      78         MOV A,B
1492      B1         ORA C
1493      C28C3C     JNZ 3C8C    ;FOR ALL BYTES
1496      CF         RST 1       ;JUMP TO MONITOR
```

WRITE SYNC BYTE AND BYTE COUNT

```
14AØ      C5         PUSH B
14A1      ØEA5       MVI C,A5     ;SYNC BYTE=A5H
14A3      CD391Ø     CALL 1Ø39    ;BYTE WRITE
14A6      C1         POP B
14A7      C5         PUSH B       ;DATA IN BC IS BYTE
14A8      CD391Ø     CALL 1Ø39    ;COUNT;WRITE IT
14AB      C1         POP B        ;C FIRST THEN
14AC      78         MOV A,B      ;B
14AD      C5         PUSH B
```

```
14AE     4F          MOV C,A
14AF     CD3910      CALL 1039
14B2     C1          POP B
14B3     C9          RET

              LOOP FF,WAITS FOR 32 FF'S

14CØ     D5          PUSH D
14C1     162Ø        MVI D,2Ø     ;32 FF'S TO COUNT
14C3     D5          PUSH D
14C4     CD5Ø1Ø      CALL 1Ø5Ø    ;BYTE READ
14C7     D1          POP D
14C8     FEFF        CPI FF       ;IS IT FF
14CA     C2C114      JNZ 14C1     ;LOOP IF NOT
14CD     15          DCR D        ;ELSE COUNT
14CE     C2C314      JNZ 14C3     ;32 FF'S
14D1     CD5Ø1Ø      CALL 1Ø5Ø    ;READ BYTE
14D4     FEFF        CPI FF       ;WAIT FOR
14D6     CAD114      JZ 14D1      ;NON FF BYTE
14D9     D1          POP D
14DA     C9          RET

              CHECKSUM

14EØ     C5          PUSH B
14E1     D5          PUSH D
14E2     E5          PUSH H
14E3     1EØØ        MVI E,ØØ     ;INIT E REG
14E5     AF          XRA A        ;CLEAR ACC
14E6     7E          MOV A,M      ;DATA IN ACC
14E7     83          ADD E        ;ADD TO E
14E8     5F          MOV E,A      ;STORE BACK
14E9     23          INX H        ;ADD ALL BYTES
14EA     ØB          DCX B
14EB     78          MOV A,B
14EC     B1          ORA C        ;RESULT IN E
14ED     C2E614      JNZ 14E6
14FØ     7B          MOV A,E      ;BRING IN ACC
14F1     E1          POP H
14F2     D1          POP D
14F3     C1          POP B
14F4     C9          RET

           START RECORDING

15ØØ     CDEØ14      CALL 14EØ    ;CALL CHECKSUM
15Ø3     32B52Ø      STA 2ØB5     ;STORE AT 2ØB5
15Ø6     E5          PUSH H
15Ø7     D5          PUSH D
15Ø8     C5          PUSH B
15Ø9     CDEØ11      CALL 11EØ    ;MOTOR ON
15ØC     CDDØ1Ø      CALL 1ØDØ    ;DELAY 2 SECONDS
15ØF     CDCØ1Ø      CALL 1ØCØ    ;LEADER
1512     C1          POP B
1513     CDAØ14      CALL 14AØ    ;WRITE SYNC A5 AND
1516     D1          POP D        ;BYTE COUNT
1517     E1          POP H
1518     CDBØ1Ø      CALL 1ØBØ    ;BLOCK WRITE
151B     3AB52Ø      LDA 2ØB5     ;CHECKSUM
151E     4F          MOV C,A      ;WRITE IT
151F     CD391Ø      CALL 1Ø39    ;BY BYTE WTITE
1522     CDCØ1Ø      CALL 1ØCØ    ;TRAILER
1525     CDFØ11      CALL 11FØ    ;MOTOR OFF
1528     C9          RET
              INITIALISE 8251

153Ø     AF          XRA A        ;CLEAR ACC
1531     D3Ø1        OUT Ø1       ;OUTPUT ØØ 3 TIMES
1533     D3Ø1        OUT Ø1       ;TO INIT 8251
1535     D3Ø1        OUT Ø1
1537     3E4Ø        MVI A,4Ø     ;RESET
1539     D3Ø1        OUT Ø1
153B     3E4E        MVI A,4E     ;MODE
153D     D3Ø1        OUT Ø1       ;
153F     3E11        MVI A,11     ;
1541     D3Ø1        OUT Ø1
1543     DB23        IN 23        ;CHANGE RECORD MODE
1545     F6Ø4        ORI Ø4
1547     D323        OUT 23
1549     21CE2Ø      LXI H,2ØCE   ;RST 7.5 VECTOR
154C     36C3        MVI M,C3     ;RELOCATE TO
154E     23          INX H
154F     36F5        MVI M,F5
1551     23          INX H
1552     3612        MVI M,12     ;12F5H
1554     C9          RET

              WAIT FOR CHAR TO OUTPUT

156Ø     DBØ1        IN Ø1
1562     E6Ø1        ANI Ø1
1564     CA6Ø15      JZ 156Ø
1567     79          MOV A,C      ;OUTPUT CHAR
1568     D3ØØ        OUT ØØ
156A     C9          RET

              INITIALISATION FOR READ MODE

157Ø     3E1Ø        MVI A,1Ø     ;CNTRØ,MODEØ
1572     D3ØB        OUT ØB
1574     3EFF        MVI A,FF     ;LOAD FF IN LSB ONLY
1576     D3Ø8        OUT Ø8
1578     3E4Ø        MVI A,4Ø     ;8155 CNTR
157A     D325        OUT 25
157C     3EØA        MVI A,ØA     ;DIVIDE BY 1Ø AND PORT
157E     D324        OUT 24       ;INIT
158Ø     3EØ8        MVI A,Ø8     ;
1582     32ØØ19      STA 19ØØ     ;INIT 8279
1585     3ECC        MVI A,CC     ;8155 START
```

```
1587    D32Ø      OUT 2Ø
1589    21CE2Ø    LXI H,2ØCE   ;NEW RST 7.5 VECTOR
158C    36C3      MVI M,C3
158E    23        INX H
158F    36AØ      MVI M,AØ
1591    23        INX H
1592    3615      MVI M,15     ;15AØH,TO COUNT 3Ø7.2 KHZ
1594    DB23      IN 23        ;IN CNTRØ;SET PC BIT2
1596    F6Ø4      ORI Ø4       ; TØ ONE
1598    D323      OUT 23
159A    3EØB      MVI A,ØB     ;SET INTERRUPT MASK
159C    3Ø        SIM
159D    C9        RET

        SERVICE ROUTINE RST 7.5,READ MODE

15AØ    3EØØ      MVI A,ØØ     ;LATCH CNTRØ AND READ
15A2    D3ØB      OUT ØB
15A4    DBØ8      IN Ø8        ;ITS CONTENTS
15A6    2F        CMA          ;COMPLEMENT IT
15A7    4F        MOV C,A      ;SAVE IN C
15A8    3E1Ø      MVI A,1Ø     ;THEN INIT CNTRØ
15AA    D3ØB      OUT ØB       ;AGAIN
15AC    3EFF      MVI A,FF     ;LOAD FF TO DOWN COUNT
15AE    D3Ø8      OUT Ø8
15BØ    C9        RET

        READ BITS AND COMPILE BYTE,BYTE READ
        DATA RETURNED IN A REG

15CØ    C5        PUSH B
15C1    D5        PUSH D
15C2    E5        PUSH H
15C3    FB        EI           ;ENABLE INT AND WAIT
15C4    76        HLT

15C5    3E28      MVI A,28     ;AT EACH INTERRUPT
15C7    B9        CMP C        ;COMPARE COUNT TO 28H
15C8    D2C315    JNC 15C3     ;IF LESS IT WAS 1,WAIT FOR
15CB    1EØØ      MVI E,ØØ     ;START BIT ;INIT E REG
15CD    Ø6Ø8      MVI B,Ø8     ;BIT COUNT
15CF    16ØØ      MVI D,ØØ     ;
15D1    FB        EI           ;WAIT NEXT PULSE
15D2    76        HLT

15D3    3E28      MVI A,28     ;COMPARE COUNT TO 28H
15D5    B9        CMP C
15D6    D2FØ15    JNC 15FØ     ;IF LESS IT IS 1
15D9    3F        CMC          ;CARRY KILLED
15DA    7A        MOV A,D      ;OLD DATA
15DB    1F        RAR          ;SHIFT RIGHT TO ABSORB
15DC    57        MOV D,A      ;ONE BIT AND SAVE BACK
15DD    Ø5        DCR B        ;ALL BITS OVER
15DE    C2D115    JNZ 15D1     ;IF NOT, REPEAT
15E1    7A        MOV A,D      ;DATA IN A REG
```

```
15E2    E1        POP H
15E3    D1        POP D
15E4    C1        POP B
15E5    C9        RET

        WAIT FOR ONE PULSE ,CALLED FOR COUNTS
        LESS THAN THRESHOLD

15FØ    3EØØ      MVI A,ØØ     ;BIT IS ONE FOR
15F2    BB        CMP E        ;E=ØØ
15F3    CAFC15    JZ 15FC      ;THEN INCEMENT
15F6    1D        DCR E        ;ELSE DECREMENT E
15F7    37        STC          ;CARRY=1
15F8    C3DA15    JMP 15DA     ;JUMP BACK

        INCREMENT E REG

15FC    1C        INR E
15FD    C3D115    JMP 15D1
        DATA READ PROGRAM
        IT READS ONE BLOCK AND DISPLAYS
        HEADER ON DISPLAY
L16ØØ 16AF

16ØØ    F3        DI           ;DISABLE INTERRUPTS
16Ø1    31AØ2Ø    LXI SP,2ØAØ  ;STACK INIT
16Ø4    CD7Ø15    CALL 157Ø    ;INIT PORTS TIMERS
16Ø7    CDEØ11    CALL 11EØ    ;MOTOR ON
16ØA    CDF81Ø    CALL 1ØF8    ;DELAY 2 SECONDS
16ØD    3EFF      MVI A,FF     ;COUNT IN CNTR2
16ØF    D3Ø8      OUT Ø8
1611    21ØØ4Ø    LXI H,4ØØØ   ;MEM POINTER
1614    C5        PUSH B
1615    CDCØ14    CALL 14CØ    ;LOOP FF
1618    FEA5      CPI A5       ;SYNC FOUND
161A    CA2316    JZ 1623      ;IF YES THEN JUMP
161D    CDCØ15    CALL 15CØ    ;ELSE READ ONE BYTE
162Ø    C31816    JMP 1618     ;TEST SYNC AGAIN

        JUMP HERE FOR BLOCK READ AFTER SYNC

1623    CDCØ15    CALL 15CØ    ;BYTE READ
1626    4F        MOV C,A      ;SAVE IN C
1627    C5        PUSH B       ;SAVE
1628    CDCØ15    CALL 15CØ    ;READ NEXT
162B    C1        POP B
162C    47        MOV B,A      ;THIS ONE IN B
162D    C5        PUSH B       ;BC HAS BYTE COUNT
162E    CDCØ15    CALL 15CØ    ;READ NEXT
1631    77        MOV M,A      ;STORE IN MEMORY
1632    23        INX H
1633    C1        POP B
1634    ØB        DCX B
1635    78        MOV A,B
```

```
1636    B1        ORA C
1637    C22D16    JNZ 162D     ;TILL ALL BYTES OVER
163A    CDC015    CALL 15C0    ;READ CHECKSUM
163D    32B620    STA 20B6     ;STORE AT 20B6
1640    CDF011    CALL 11F0    ;MOTOR OFF
1643    CD5014    CALL 1450    ;COPY HEADER
1646    CD3011    CALL 1130    ;DISPLAY HEADER
1649    CF        RST 1        ;BACK TO MONITOR
```

A/D CONVERTER

```
1650    C5        PUSH B
1651    018080    LXI B,8080   ;INIT BC
1654    79        MOV A,C      ;OUTPUT TO
1655    D321      OUT 21       ;D/A CONVERTER
1657    DB22      IN 22        ;READ COMPARATOR
1659    1F        RAR          ;BIT
165A    D26116    JNC 1661     ;IF LESS THAN INPUT
165D    78        MOV A,B      ;IF MORE THEN
165E    2F        CMA          ;COMPLEMENT
165F    A1        ANA C        ;CURRENT BIT SET TO 0
1660    4F        MOV C,A      ;SAVE IT
1661    3EFE      MVI A,FE     ;
1663    A0        ANA B        ;
1664    CA6F16    JZ 166F      ;ALL BITS OVER THEN JUMP
1667    78        MOV A,B      ;ELSE
1668    0F        RRC          ;MOVE TO NEXT MSB
1669    47        MOV B,A      ;
166A    B1        ORA C        ;DO SUCCESSIVE APPROX
166B    4F        MOV C,A      ;SAVE IN C
166C    C35516    JMP 1655     ;REPEAT ALL BITS
```

LOAD DATA IN RAM

```
166F    71        MOV M,C
1670    C1        POP B
1671    C9        RET
```

TEST D/A CONVERTER,GENERATE RAMP

```
1680    31A020    LXI SP,20A0 ;INIT STACK
1683    CDA013    CALL 13A0    ;PORT &TIMER INIT
1686    3E00      MVI A,00     ;OUT 00
1688    D321      OUT 21       ;TO D/A
168A    3C        INR A        ;INCREMENT
168B    C38816    JMP 1688     ;REPEAT
```

TEST DATA ,COUNT 307.2 KHZ IN CNTRØ AND STORE COUNTS FOR SUCCESSIVE PULSES IN RAM STARTING 4000H

```
1690    F3        DI           ;DISABLE INTERRUPTS
1691    31A020    LXI SP,20A0 ;INIT STACK
1694    CD7015    CALL 1570    ;AND PORTS, TIMERS
1697    210040    LXI H,4000   ;MEMORY POINTER
169A    11FF1F    LXI D,1FFF   ;BYTE COUNT
169D    3E0B      MVI A,0B     ;ENABLE INTERRUPT
169F    30        SIM
16A0    FB        EI
16A1    76        HLT

16A2    71        MOV M,C      ;STORE COUNT IN RAM
16A3    23        INX H        ;NEXT LOCATION
16A4    1B        DCX D        ;COUNT BYTES
16A5    7A        MOV A,D
16A6    B3        ORA E
16A7    C2A016    JNZ 16A0     ;STORE 1FFFH LOCATIONS
16AA    CF        RST 1        ;JUMP TO MONITOR
```

HEXDUMP

```
16C0 31 A0 20 21 00 40 01 00 20 CD D0 16 CF FF FF FF
16D0 C5 1E 10 CD EB 05 0E 20 CD FA 07 7E CD C7 06 C1
16E0 0B 78 B1 C8 C5 23 1D 7B FE 08 CA D6 16 7B A7 CA
16F0 D1 16 C3 DB 16 FF FF FF FF FF FF FF FF FF FF FF
```

DATA TRANSFER,CALLS BLOCK TRANSFER

```
16C0    31A020    LXI SP,20A0 ;INIT STACK
16C3    210040    LXI H,4000  ;MEMORY POINTER
16C6    010020    LXI B,2000  ;BYTE COUNT 8096
16C9    CDD016    CALL 16D0   ;CALL BLOCK XFER
16CC    CF        RST 1       ;BACK TO MONITOR
```

BLOCK TRANSFER,HL MEM POINTER BC BYTE COUNTER

```
16D0    C5        PUSH B
16D1    1E10      MVI E,10     ;16 BYTES IN A ROW
16D3    CDEB05    CALL 05EB    ;GIVE CR-LF
16D6    0E20      MVI C,20     ;SPACE
16D8    CDFA07    CALL 07FA    ;OUT
16DB    7E        MOV A,M      ;GET DATA
16DC    CDC706    CALL 06C7    ;SEND AS TWO ASCII
16DF    C1        POP B
16E0    0B        DCX B        ;COUNT
16E1    78        MOV A,B
16E2    B1        ORA C
16E3    C8        RZ           ;ALL BYTES OVER THEN RET
16E4    C5        PUSH B
16E5    23        INX H        ;ELSE GET NEXT DATA
16E6    1D        DCR E        ;COLUMN COUNT
16E7    7B        MOV A,E
16E8    FE08      CPI 08       ;8 BYTES OVER?
16EA    CAD616    JZ 16D6      ;IF YES GIVE A SPACE
16ED    7B        MOV A,E      ;ELSE
16EE    A7        ANA A
16EF    CAD116    JZ 16D1      ;IF 0 THEN
16F2    C3DB16    JMP 16DB     ;GIVE CR-LF AND THEN 16 BYTES
```

HEXDUMP OF RECORDER PROGRAM

```
1ØØØ DB 23 F6 Ø8 D3 23 D5 16 13 16 C2 Ø9 1Ø D1 DB 23
1Ø1Ø E6 37 D3 23 C9 CD ØØ 1Ø D5 16 52 15 C2 1B 1Ø D1
1Ø2Ø C9 DB 22 17 DA 21 1Ø DB 22 17 D2 27 1Ø C9 79 17
1Ø3Ø 4F DC 15 1Ø CD ØØ 1Ø FB C9 C3 6Ø 15 21 1Ø F3 CD
1Ø4Ø 2E 1Ø Ø5 C2 3B 1Ø CD 21 1Ø C9 FF FF FF FF FF FF
1Ø5Ø C3 CØ 15 ØØ 16 ØØ 14 DB 22 E6 4Ø C2 56 1Ø 14 DB
1Ø6Ø 22 E6 4Ø CA 5E 1Ø Ø5 CA 78 1Ø 7A FE 4Ø CD 88 1Ø
1Ø7Ø D2 54 1Ø CD 9Ø 1Ø C3 54 1Ø 7A FE 8Ø CD 88 1Ø D2
1Ø8Ø 85 1Ø CD 9Ø 1Ø 79 C9 FF F5 79 17 4F F1 C9 FF FF
1Ø9Ø DB 22 E6 4Ø C2 9Ø 1Ø DB 22 E6 4Ø CA 97 1Ø C9 FF
1ØAØ 16 8Ø 4B D5 CD 39 1Ø D1 15 C2 A2 1Ø C9 FF FF FF
1ØBØ C5 4E CD 39 1Ø 23 C1 ØB 78 B1 C2 BØ 1Ø C9 FF FF
1ØCØ C5 1E ØØ CD AØ 1Ø 1E FF CD AØ 1Ø C1 C9 FF FF FF
1ØDØ D5 11 1Ø ØØ D5 11 FF FF CD F1 Ø5 D1 1B 7A B3 C2
1ØEØ D4 1Ø D1 C9 3A B3 2Ø F6 8Ø 32 B3 2Ø 11 FØ 1F CD
1ØFØ DB 12 C9 FF FF FF FF FF D5 11 Ø4 ØØ C3 D4 1Ø FF

11ØØ 21 AØ 2Ø CD 25 11 FE 2Ø CØ CD 25 11 FE 6Ø CØ CD
111Ø 25 11 FE 6Ø CØ CD 25 11 FE 24 CØ CD 25 11 E6 FF
112Ø CØ CD 25 11 C9 36 ØØ 23 34 7E 37 3F 27 77 C9 FF
113Ø E5 C5 F5 21 A8 2Ø Ø6 Ø3 3E 9Ø 32 ØØ 19 CD 4Ø 13
114Ø 3E 93 32 ØØ 19 21 A7 2Ø Ø8 Ø6 CD ØØ 13 3E 9F 32
115Ø ØØ 19 3A B3 2Ø 2F 32 ØØ 18 F1 C1 E1 C9 FF FF FF
116Ø E5 D5 DB 23 E6 EF D3 23 2A AC 2Ø EB 2A AE 2Ø CD
117Ø 5Ø 16 23 1B 7A B3 CC 93 11 22 AE 2Ø EB 22 AC 2Ø
118Ø D1 E1 CD BØ 11 CD C3 11 DB 23 F6 1Ø D3 23 C9 FF
119Ø FF FF FF 7C FE 6Ø CA E4 1Ø 26 4Ø 3A B3 2Ø F6 4Ø
11AØ 32 B3 2Ø 11 FØ 1F CD DB 12 C9 FF FF FF FF FF FF
11BØ C5 3A B3 2Ø 4F 3C E6 Ø3 47 79 E6 FC BØ 32 B3 2Ø
11CØ C1 C9 FF C5 DB 23 E6 FC 4F 3A B3 2Ø E6 Ø3 B1 D3
11DØ 23 C1 C9 DB 23 F6 2Ø D3 23 DB 23 E6 1F D3 23 C8
11EØ DB 23 F6 Ø8 D3 23 3A B3 2Ø F6 1Ø 32 B3 2Ø C9 FF
11FØ DB 23 E6 F7 D3 23 3A B3 2Ø E6 EF 32 B3 2Ø C9 F7

12ØØ 3A B3 2Ø E6 1Ø CA 3Ø 11 C9 FF FF FF FF FF FF FF
121Ø 3A B3 2Ø F6 2Ø 32 B3 2Ø E6 Ø4 11 ØØ 2Ø CA 22 12
122Ø 16 ØØ 21 ØØ 4Ø 19 Ø1 ØØ 2Ø CD ØØ 15 3A B3 2Ø E6
123Ø DF 32 B3 2Ø 7C FE 8Ø C2 44 12 3A B3 2Ø E6 73 F6
124Ø Ø8 C3 5Ø 12 FE 8Ø C2 11 Ø6 3A B3 2Ø E6 B3 F6 Ø4
125Ø 32 B3 2Ø C3 C3 12 FF FF FF FF FF FF FF FF FF FF
1263 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
127Ø CD 6Ø 11 3E 6Ø 32 B4 2Ø F1 E1 D1 C1 FB C9 FF FF
128Ø CD ØØ 11 3E 8Ø 32 B4 2Ø F1 E1 D1 C1 FB C9 FF FF
129Ø CD 3Ø 11 3E 7Ø 32 B4 2Ø F1 E1 D1 C1 FB C9 FF FF
12AØ 31 AØ 2Ø 21 Ø4 7Ø 22 B3 2Ø 21 FØ 1F 22 AC 2Ø CD
12BØ AØ 13 3E ØB 3Ø 3E 1Ø D3 23 21 ØØ 4Ø CD DB 12 22
12CØ AE 2Ø FB 3A B3 2Ø E6 CØ CA C3 12 C3 1Ø 12 FF 4E
12DØ CD 39 1Ø 23 1B 7A B3 C2 CF 12 C8 D5 C5 11 AB 2Ø
12EØ Ø1 Ø6 ØA 1A 77 23 1B Ø5 C2 E3 12 36 ØØ 23 ØD C2
12FØ EB 12 C1 D1 C9 C5 D5 E5 F5 3A B4 2Ø 6F 26 12 E9

13ØØ ØE Ø2 CD 1C 13 E5 21 F9 2Ø 7E CD 2D 13 32 ØØ 18
131Ø 23 ØD C2 Ø9 13 E1 2B Ø5 C2 ØØ 13 C9 7E ØF ØF ØF
132Ø ØF E6 ØF 32 F9 2Ø 7E E6 ØF 32 FA 2Ø C9 E5 21 84
133Ø Ø3 85 6F 7E E1 2F C9 FF FF FF FF FF FF FF FF FF
134Ø 3E 9Ø 32 ØØ 19 ØØ ØØ ØØ D5 5E 16 ØØ E5 21 AØ 13
135Ø 19 7E 2F E1 ØØ 32 ØØ 18 23 Ø5 C2 49 13 D1 C9 FF
136Ø 3E 3E D3 ØB 3E Ø4 D3 Ø8 3E ØØ D3 Ø8 3E 7C D3 ØB
137Ø 3E ØØ D3 Ø9 3E 14 D3 Ø9 3E BE D3 ØB 3E ØØ D3 ØA
138Ø 3E Ø6 D3 ØA 3E 4Ø D3 25 3E ØA D3 24 3E CD D3 2Ø
139Ø C9 DB 23 E6 3B D3 23 C9 DB 23 F6 Ø4 D3 23 C9 FF
13AØ CD 6Ø 13 CD 91 13 3E Ø8 32 ØØ 19 CD 3Ø 15 C9 FF
13BØ FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
13CØ ØØ 28 22 ØØ ØØ ØØ ØØ 2Ø 93 FØ ØØ 64 Ø8 Ø4 Ø8 21
13DØ F3 8Ø B5 F4 66 D6 D7 7Ø F7 76 77 C7 93 E5 B7 17
13EØ 35 77 C7 93 E5 97 17 D3 47 4Ø E1 67 83 45 45 C5
13FØ 37 CD Ø5 D6 87 E3 C1 C1 25 E6 B5 93 46 FØ 32 8Ø

14ØØ C3 ØØ 16 CD AØ 13 CD EØ 11 CD F8 1Ø 21 ØØ 3Ø C5
141Ø CD CØ 14 FE A5 CA 2Ø 14 CD 5Ø 1Ø C3 13 14 ØØ ØØ
142Ø CD 5Ø 1Ø 4F C5 CD 5Ø 1Ø C1 47 C5 CD 5Ø 1Ø 77 23
143Ø C1 ØB 78 B1 C2 2A 14 CD 5Ø 1Ø 32 B6 2Ø CD FØ 11
144Ø CD 5Ø 14 CD 3Ø 11 CF FF FF FF FF FF FF FF FF FF
145Ø C5 21 ØØ 4Ø 11 AB 2Ø Ø6 ØA 7E 12 23 1B Ø5 C2 59
146Ø 14 C1 C9 FF FF FF FF FF FF FF FF FF FF FF FF FF
147Ø C5 CD 5Ø 1Ø 77 23 ØØ C1 ØB 78 B1 C2 7Ø 2Ø C9 FF
148Ø 31 AØ 2Ø 21 ØØ 38 11 ØØ 1Ø Ø1 FF Ø7 7E 12 23 13
149Ø ØB 78 B1 C2 8C 3C CF FF FF FF FF FF FF FF FF FF
14AØ C5 ØE A5 CD 39 1Ø C1 C5 CD 39 1Ø C1 78 C5 4F CD
14BØ 39 1Ø C1 C9 FF FF FF FF FF FF FF FF FF FF FF FF
14CØ D5 16 2Ø D5 CD 5Ø 1Ø D1 FE FF C2 C1 14 15 C2 C3
14DØ 14 CD 5Ø 1Ø FE FF CA D1 14 D1 C9 FF FF FF FF FF
14EØ C5 D5 E5 1E ØØ AF 7E 83 5F 23 ØB 78 B1 C2 E6 14
14FØ 7B E1 D1 C1 C9 FF FF FF FF FF FF FF FF FF FF FF

15ØØ CD EØ 14 32 B5 2Ø E5 D5 C5 CD EØ 11 CD DØ 1Ø CD
151Ø CØ 1Ø C1 CD AØ 14 D1 E1 CD BØ 1Ø 3A B5 2Ø 4F CD
152Ø 39 1Ø CD CØ 1Ø CD FØ 11 C9 FF FF FF FF FF FF FF
153Ø AF D3 Ø1 D3 Ø1 D3 Ø1 3E 4Ø D3 Ø1 3E 4E D3 Ø1 3E
154Ø 11 D3 Ø1 DB 23 F6 Ø4 D3 23 21 CE 2Ø 36 C3 23 36
155Ø F5 23 98 12 C9 FF FF FF FF FF FF FF FF FF FF FF
156Ø DB [illegible] E6 Ø1 CA 6Ø 15 79 D3 ØØ C9 FF FF FF FF FF
157Ø 3E 1Ø D3 ØB 3E FF D3 Ø8 3E 4Ø D3 25 3E ØA D3 24
158Ø 3E [illegible] 32 ØØ 19 3E CC D3 2Ø 21 CE 2Ø 36 C3 23 36
159Ø AØ 23 36 16 DB 23 E6 3B D3 23 3E ØB 3Ø C9 FF FF
15AØ 3E ØØ D3 ØB DB Ø8 2F 4F 3E 1Ø D3 ØB 3E FF D3 Ø8
15BØ C9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
15CØ C5 D5 E5 FB 76 3E 25 B9 D2 C3 15 1E ØØ Ø6 Ø8 16
15DØ ØØ FB 76 3E 25 B9 D2 FØ 15 3F 7A 1F 57 Ø5 C2 D1
15EØ 15 7A E1 D1 C1 C9 FF FF FF FF FF FF FF FF FF FF
15FØ 3E ØØ BB CA FC 15 1D 37 C3 DA 15 ØØ 1C C3 D1 15
```

```
1600 F3 31 AØ 2Ø CD 7Ø 15 CD EØ 11 CD F8 1Ø 3E FF D3
1610 Ø8 21 ØØ 4Ø C5 CD CØ 14 FE A5 CA 23 16 CD CØ 15
1620 C3 18 16 CD CØ 15 4F C5 CD CØ 15 C1 47 C5 CD CØ
1630 15 77 23 C1 ØB 78 B1 C2 2D 16 CD CØ 15 32 B6 2Ø
1640 CD FØ 11 CD 5Ø 14 CD 3Ø 11 CF FF FF FF FF FF FF
1650 C5 Ø1 8Ø 8Ø 79 D3 21 DB 22 1F D2 61 16 78 2F A1
1660 4F 3E FE AØ CA 6F 16 78 ØF 47 B1 4F C3 55 16 71
1670 C1 C9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1680 31 AØ 2Ø CD AØ 13 3E ØØ D3 21 3C C3 88 16 FF FF
1690 F3 31 AØ 2Ø CD 7Ø 15 21 ØØ 4Ø 11 FF 1F 3E ØB 3Ø
16AØ FB 76 71 23 1B 7A B3 C2 AØ 16 CF FF FF FF FF FF
```

APPENDIX N

DETAILS OF RS232C AND INPUT OUTPUT CONNECTORS

RS232 C,9 PIN FEMALE CONNECTOR D TYPE

1.NC

2.RECEIVE DATA

3.TRANSMIT DATA

4.DTR-----------JOINED TO DSR

5.GND

6.DSR

7.RTS-----------JOINED TO CTS

8.CTS

9.NC

INPUT OUTPUT CONNECTOR 9 PIN D TYPE FEMALE

1.CHANNEL 1 IN

2.CHANNEL 2 IN

3.CHANNEL 3 IN

4.CHANNEL 4 IN

5.GND

6.CHANNEL 1 OUT

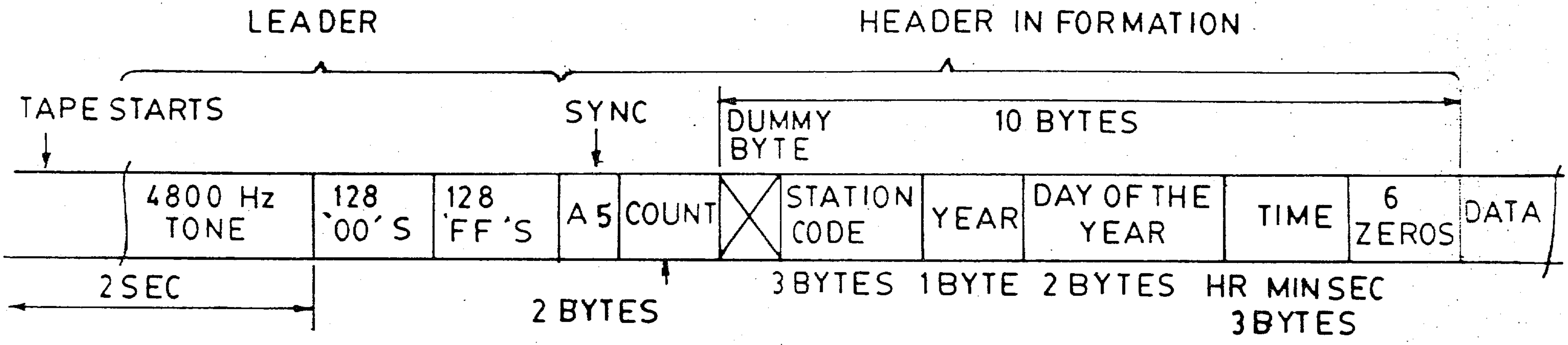
7.CHANNEL 2 OUT

8.CHANNEL 3 OUT

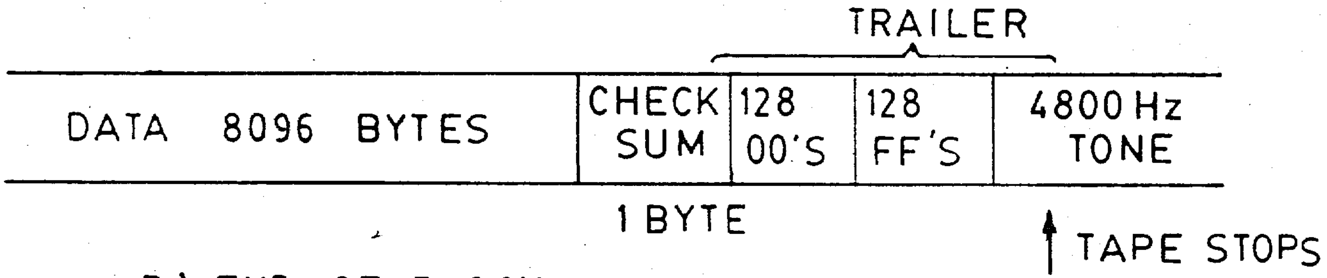
9.CHANNEL 4 OUT

APPENDIX O

CASSETTE TAPE DATA FORMAT



A) BEGINNING OF BLOCK



B) END OF BLOCK

NOTES;1 INTER BLOCK GAP ≈ 95 MM TAPE (2 SECONDS)

2 DATA RECORDED IN 8 BIT SERIAL FORM ONE START BIT (0)
8 BIT DATA LSB FIRST ONE STOP BIT

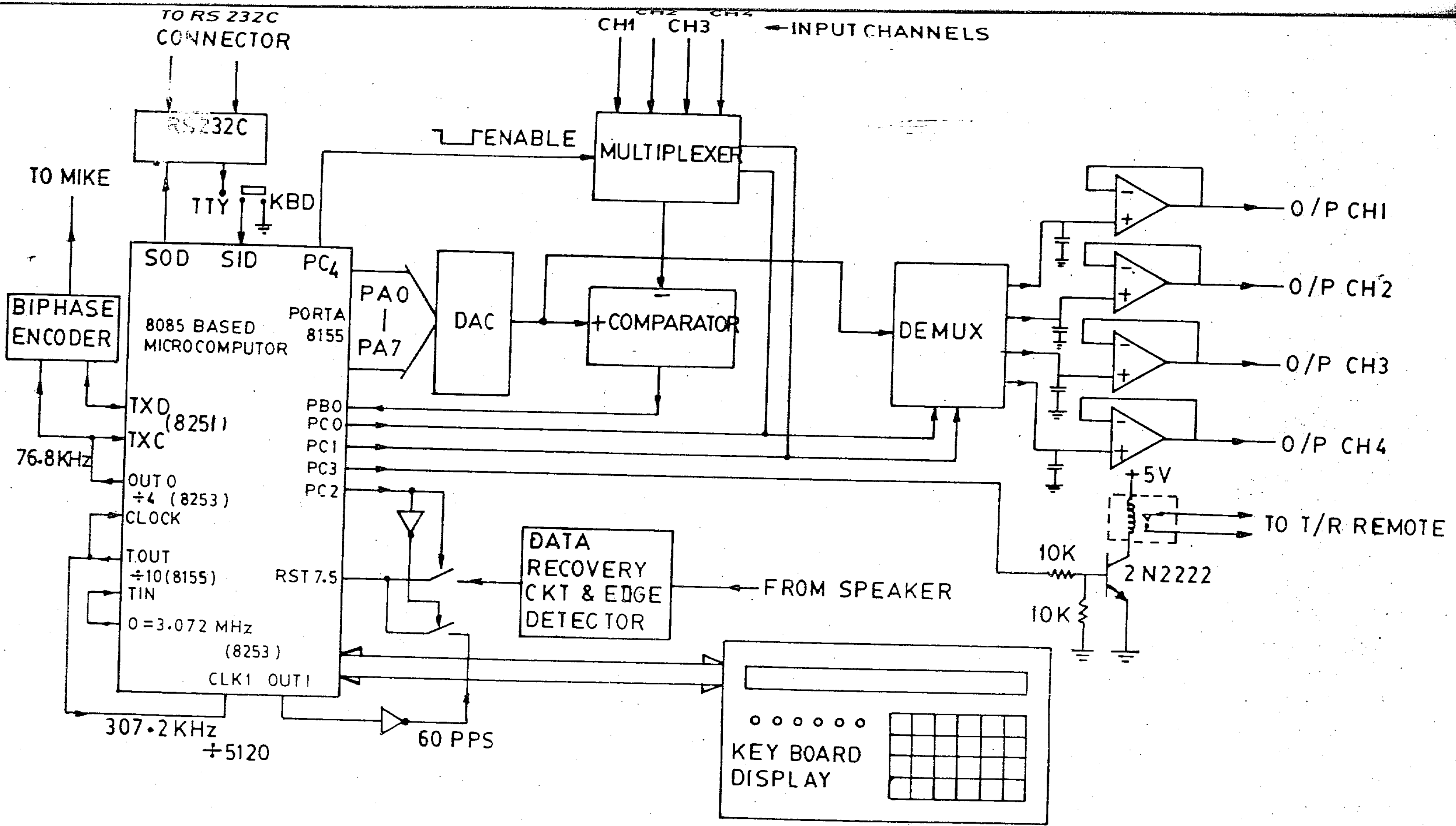


FIG-1 SYSTEM BLOCK DIAGRAM

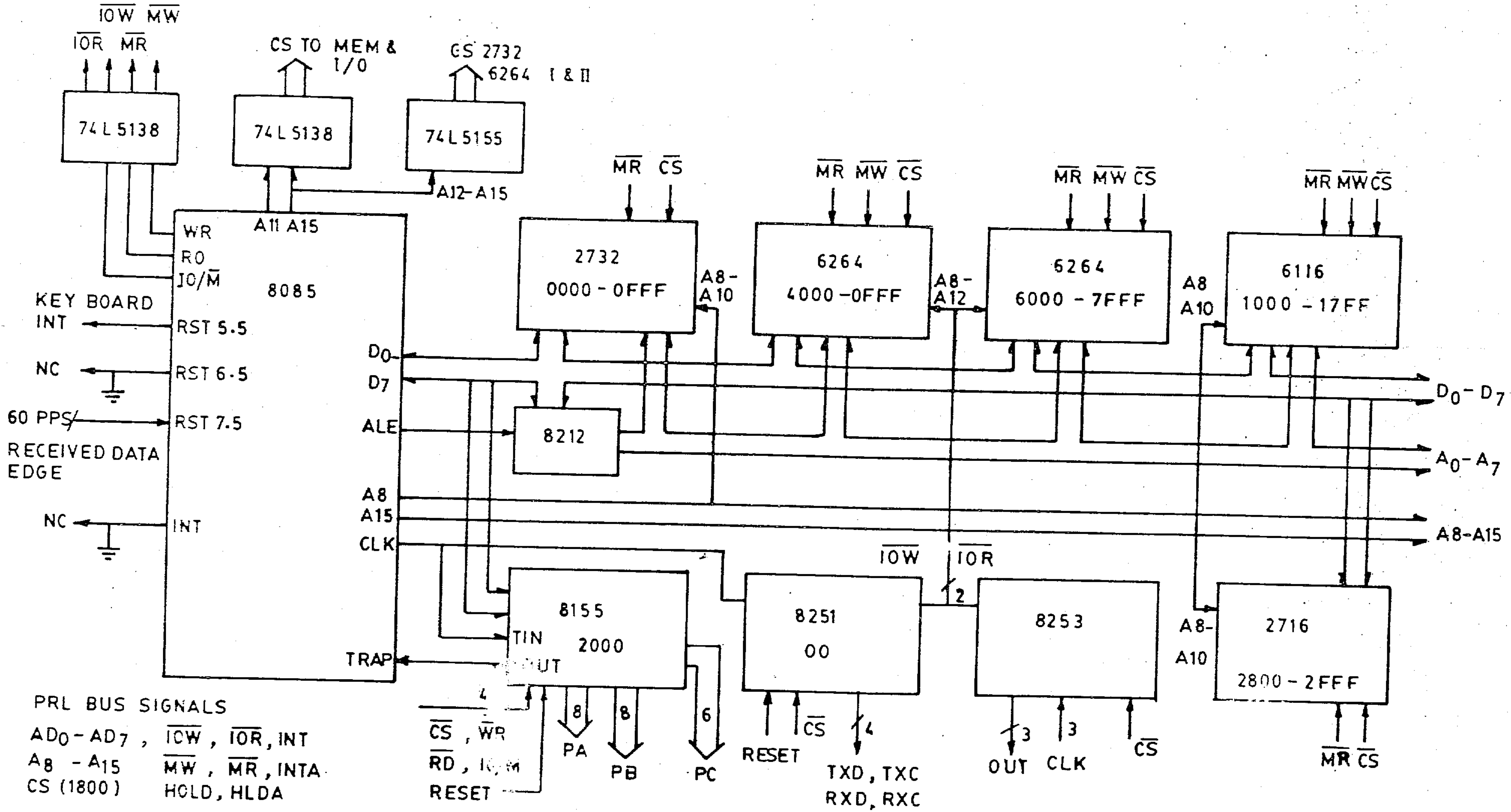


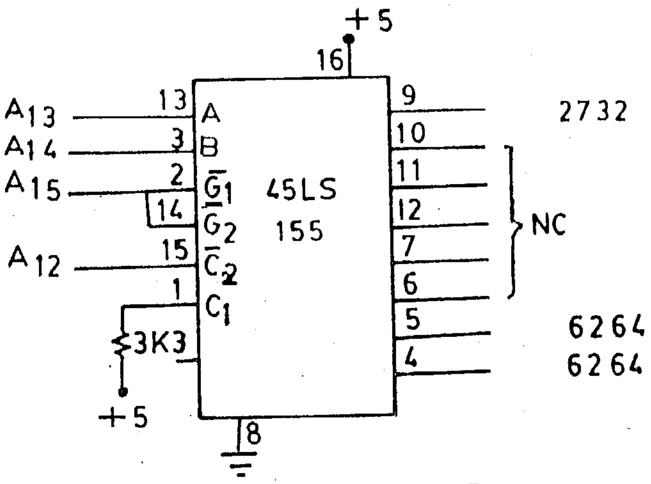
FIG.-2 MICROCOMPUTER BLOCK DIAGRAM

8000 - FFFF	NOT DECODED
6000 - 7FFF	BUFFER B 6264
4000 - 5FFF	BUFFER A 6264
3800 - 3FFF	EPROM 2716
2800 - 37FF	NOT USED
2000 - 27FF	8155
1000 - 1FFF	KEY BOARD
1000 - 17FF	RAM 6116
0000 - 0FFF	MONITOR 2732

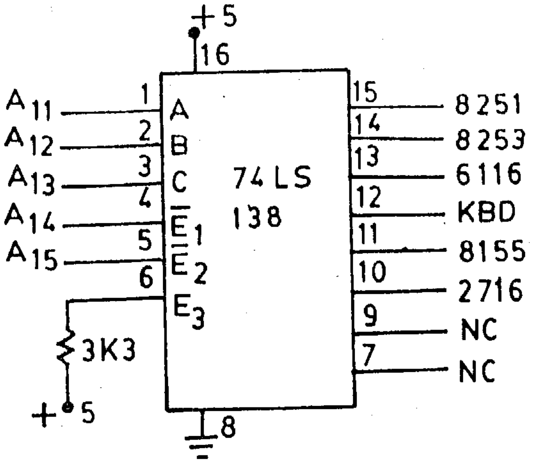
MEMORY MAP

40 - FF	NOT DECODED
38 - 3F	NOT USED
30 - 37	
28 - 2F	
20 - 27	8155
18 - 1F	NOT USED
10 - 17	
08 - 0F	8253
00 - 07	8251

I/O MAP



MEMORY DECODER
(OFF BOARD)



MEM & I/O DECODER

FIG.- 3 MEMORY MAP , I/O MAP AND DECODING LOGIC

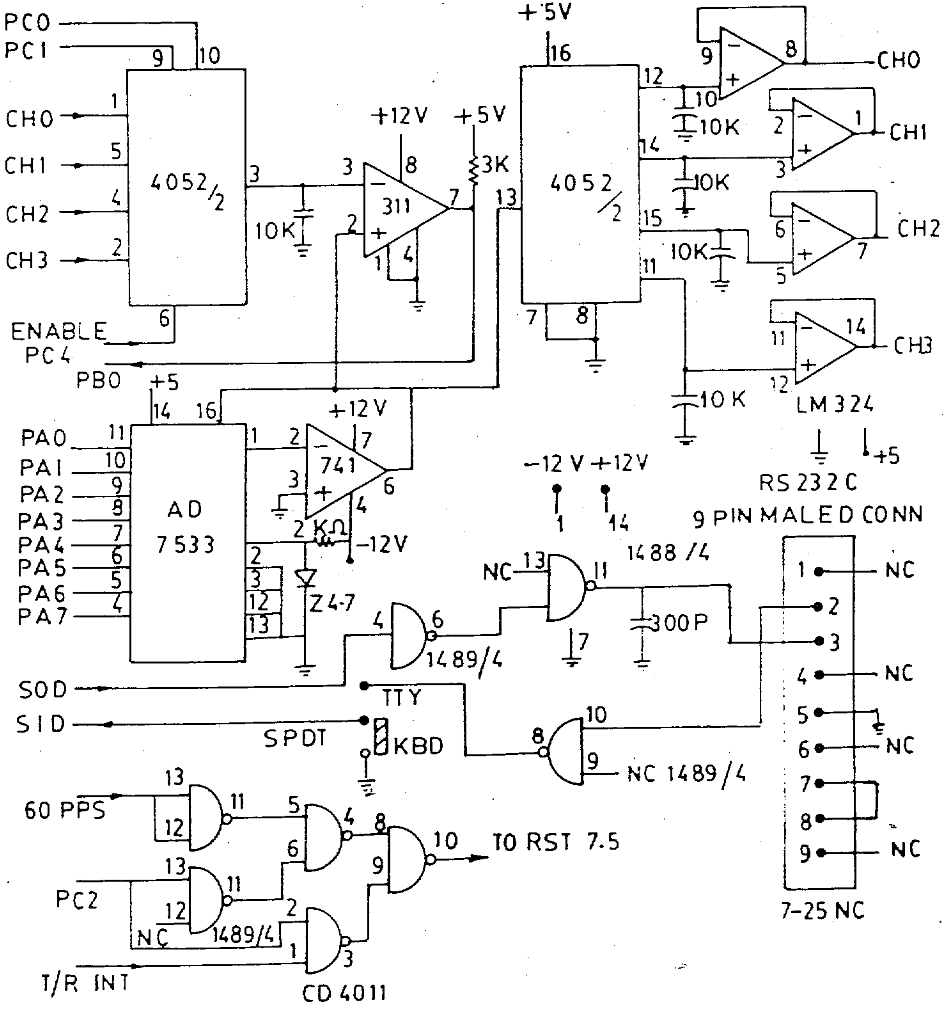


FIG. 4 MUX, DEMUX, D/A, INTERRUPT SELECT AND RS 232 C DRIVER

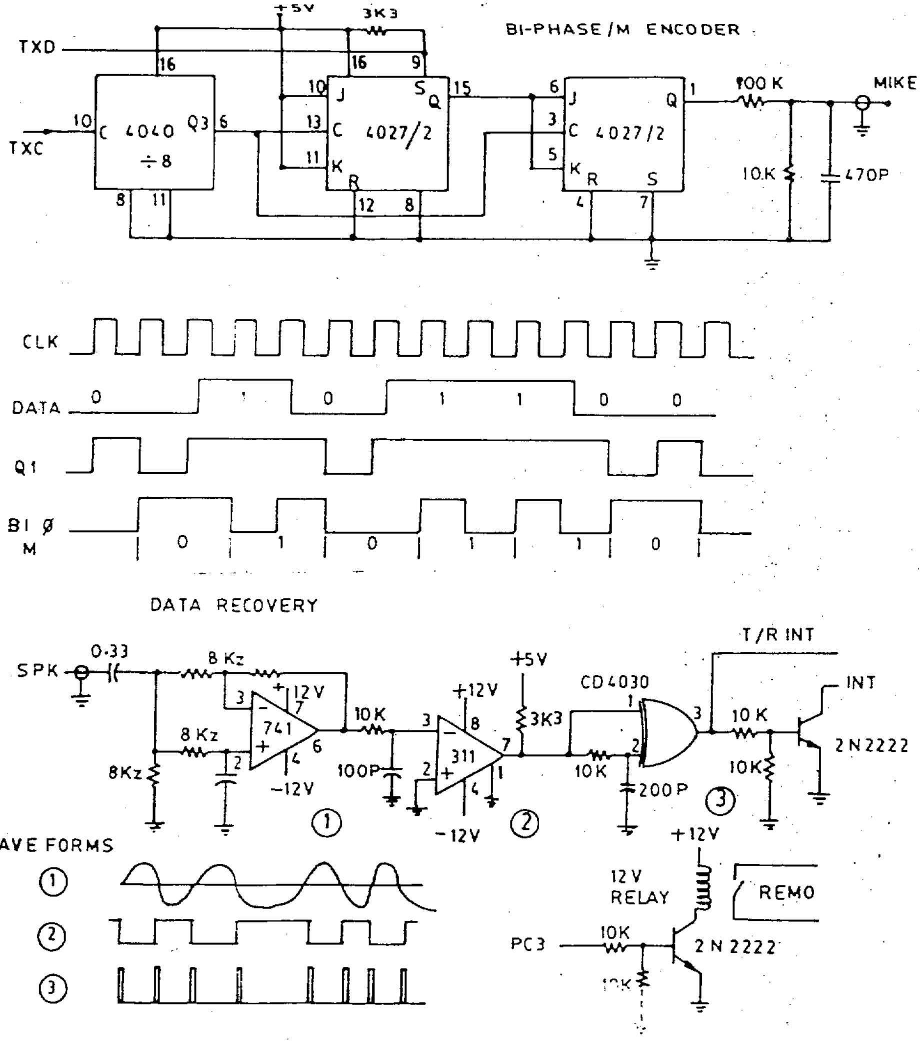


FIG. 5 CASSETTE TAPE READ/WRITE CIRCUIT

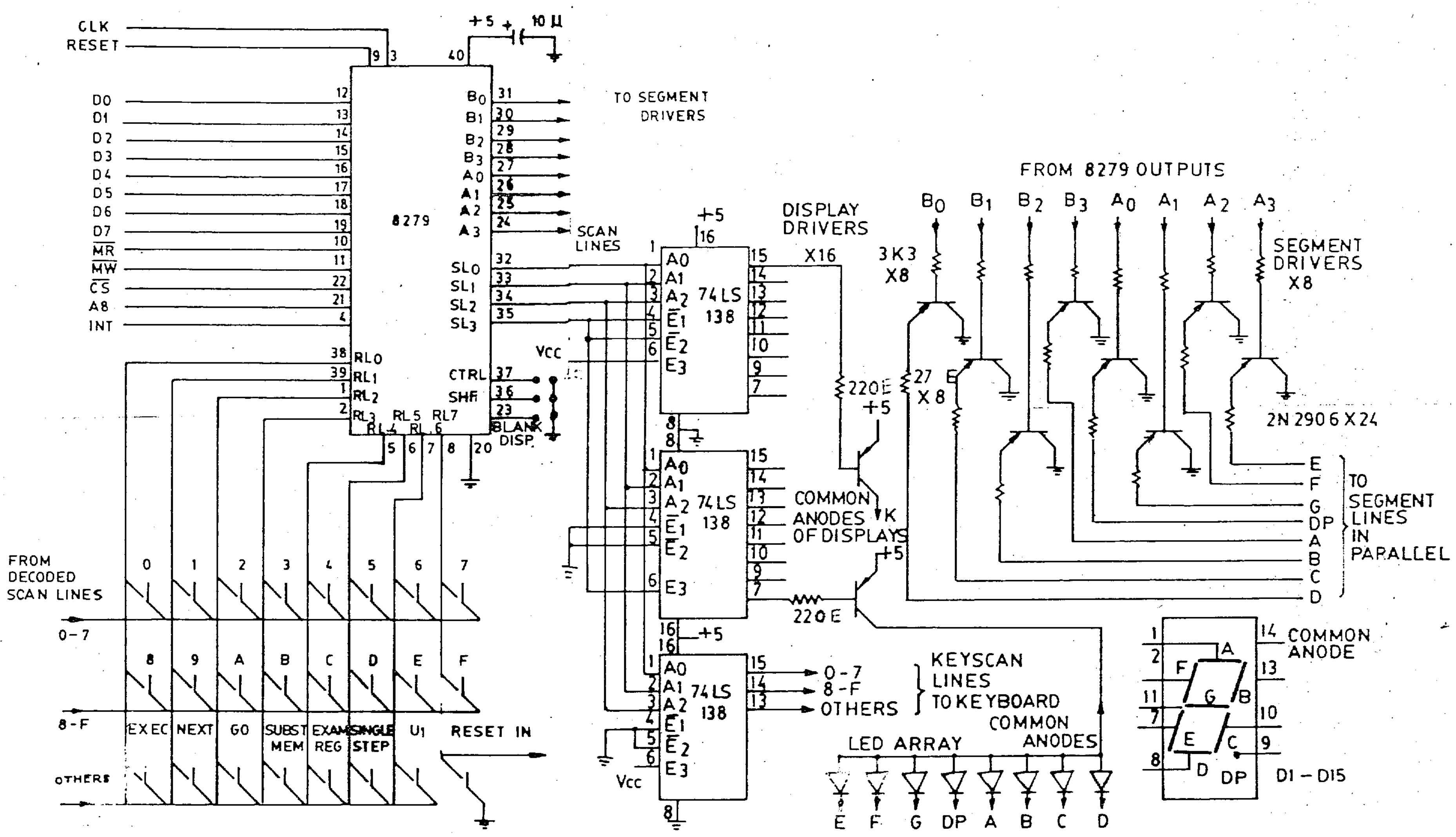


FIG.6 - KEY BOARD & DISPLAY

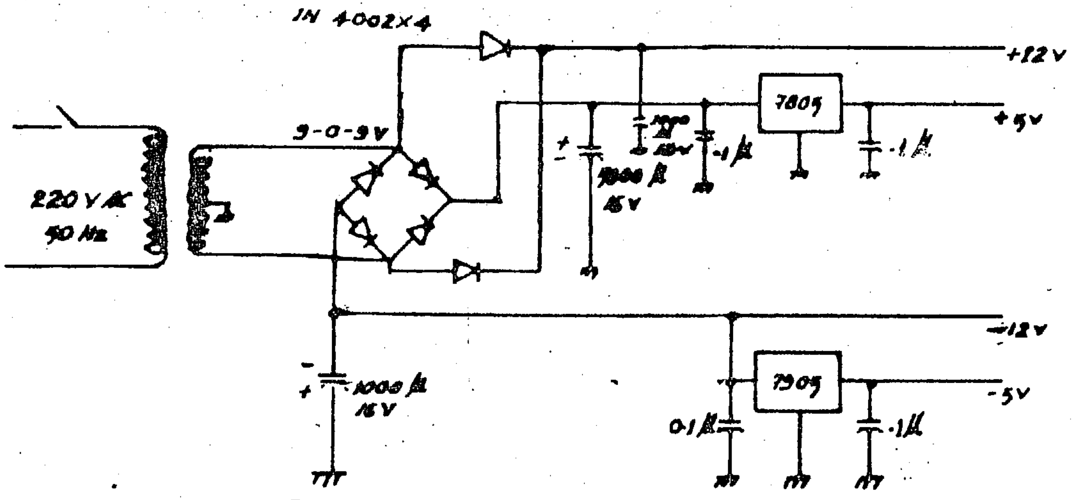


FIG 7 POWER SUPPLY.

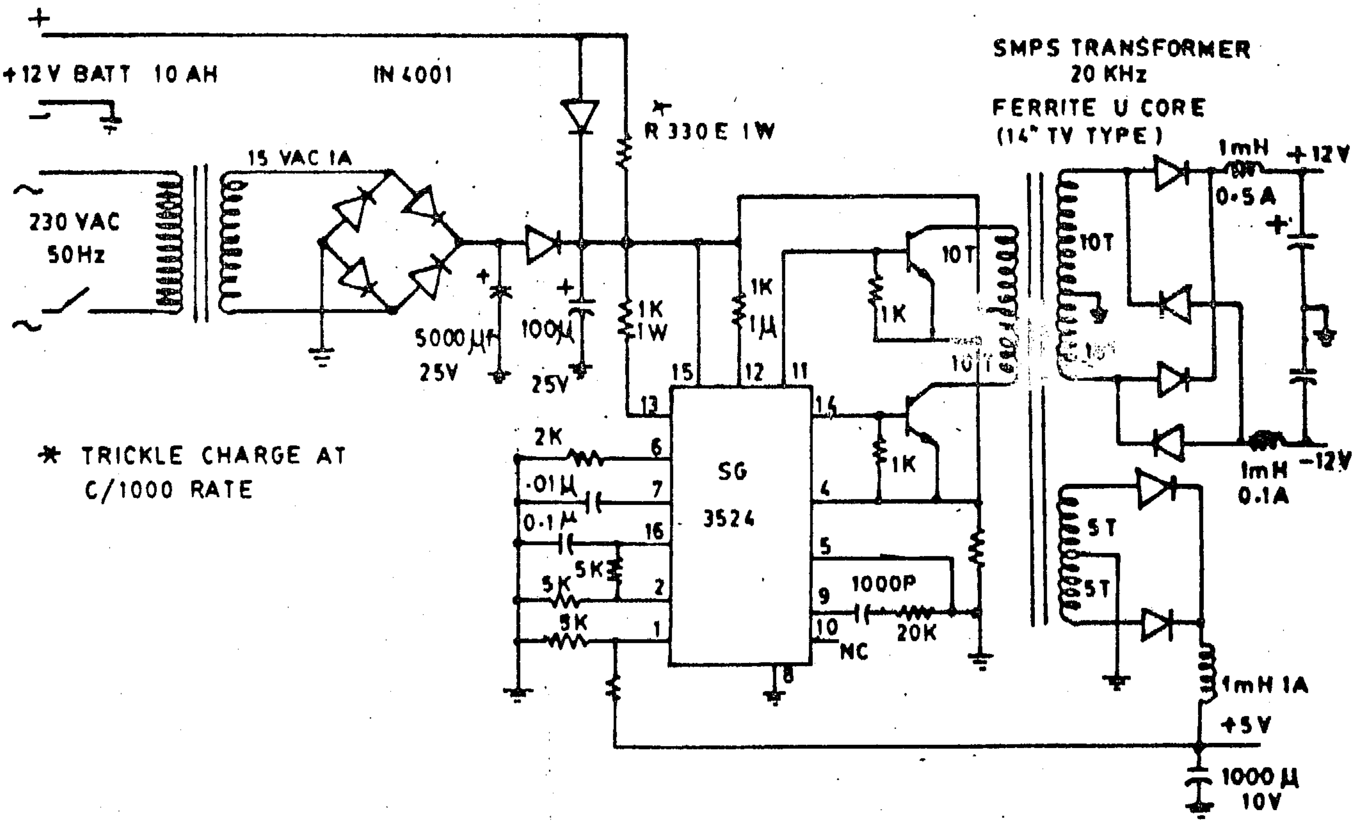


FIG.-8 SWITCH MODE POWER SUPPLY WITH BATTERY BACK UP

CPU CARD(I/O)

COMP. SIDE		SOLD. SIDE
CLK 2	1	PC5
OUT 2	2	PC4
GATE 2	3	PC3
SOD	4	PC2
Timer in	5	PC1
TTY D KBD SID	6	PC0
CLK1	7	PB7
TRAP	8	PB 6
8155 Timerout	9	PB 5
GATE1	10	PB4
RST7.5	11	PB3
RST6.5	12	PB2
OUT 0	13	PB1
RST5.5	14	PB0
OUT 1	15	PA7
GATE 0	16	PA6
CLK 1	17	PA5
GND	18	PA4
TXD	19	PA 3
TO OUT TO TXC	20	PA2
RXD	21	PA1
RXC	22	PA 0

KBD DISPLAY

1	I 5V
2	NC
3	NC
4	NC
5	NC
6	μ,CLKOUT
7	RST 5.5
8	Reset in
9	$\overline{MR}$/$\overline{IOR}$
10	$\overline{MW}$/$\overline{IOW}$
11	D0
12	D1
13	D2
14	D3
15	D4
16	D5
17	D6
18	D7
19	A8
20	CS($\overline{KBDCS}$)
21	Reset key
22	GND

CPU CARD BUS

COMP. SIDE		SOLD SIDE
+5V	1	+5V
Reset out	2	Reset key
INTR	3	HOLD
NC	4	HLDA
$\overline{INTA}$	5	μp clkout
NC	6	A15 to 8155 timer in
NC	7	A14
$\overline{KBDCS}$	8	A13
NC	9	A12
$\overline{IOW}$	10	A11
$\overline{IOR}$	11	A10
$\overline{MW}$	12	A9
$\overline{MR}$	13	A8
D7	14	A7
D6	15	A6
D5	16	A5
D4	17	A4
D3	18	A3
D2	19	A2
D1	20	A1
D0	21	A0
GND	22	GND

CASSETTE INTERFACE

NEAR LM 324

	1	-12V
	2	GND
	3	SPEAKER
	4	+12V
	5	PC3
	6	REM
	7	REM
	8	NC
	9	O/P CH3
	10	O/P CH0
	11	+5V
	12	O/P CH1
	13	O/P CH2
	14	I/P CH0
	15	I/P CH2, RXD
	16	I/P CH3
	17	I/P CH1
	18	PC4
	19	PC1
	20	PC0
	21	-12V
TXD	22	3 NO of RS 2329 PIN

NEAR 4011

1	GND
2	PA0
3	PA1
4	PA2
5	PA3
6	PA 4
7	PA 5
8	PA 6
9	PA 7
10	PC 2
11	OUT1
12	NC
13	RST 7.5
14	GND
15	RS 232C 2 no. 9 pin
16	TTY
17	SOD
18	PB0
19	TXC
20	TXD
21	GND
22	MIKE

R/P AMP

1	GND
2	VC1
3	VC2
4	SPK
5	GND
6	+5V
7	RPH
8	
9	RPH
10	MIKE
11	GND
12	GND

ANALOG I/O GAIN

1	CH0	I/P
2	CH1	I/P
3	CH2	I/P
4	CH3	I/P
5	GND	
6	CH0	O/P
7	CH1	O/P
8	CH2	O/P
9	CH3	O/P

NOTE:-SPEAKER/MIKE SHOULD BE CONNECTED THROUGH SHIELDED CABLE

FIG. 9 INTERCONNECTION DETAILS

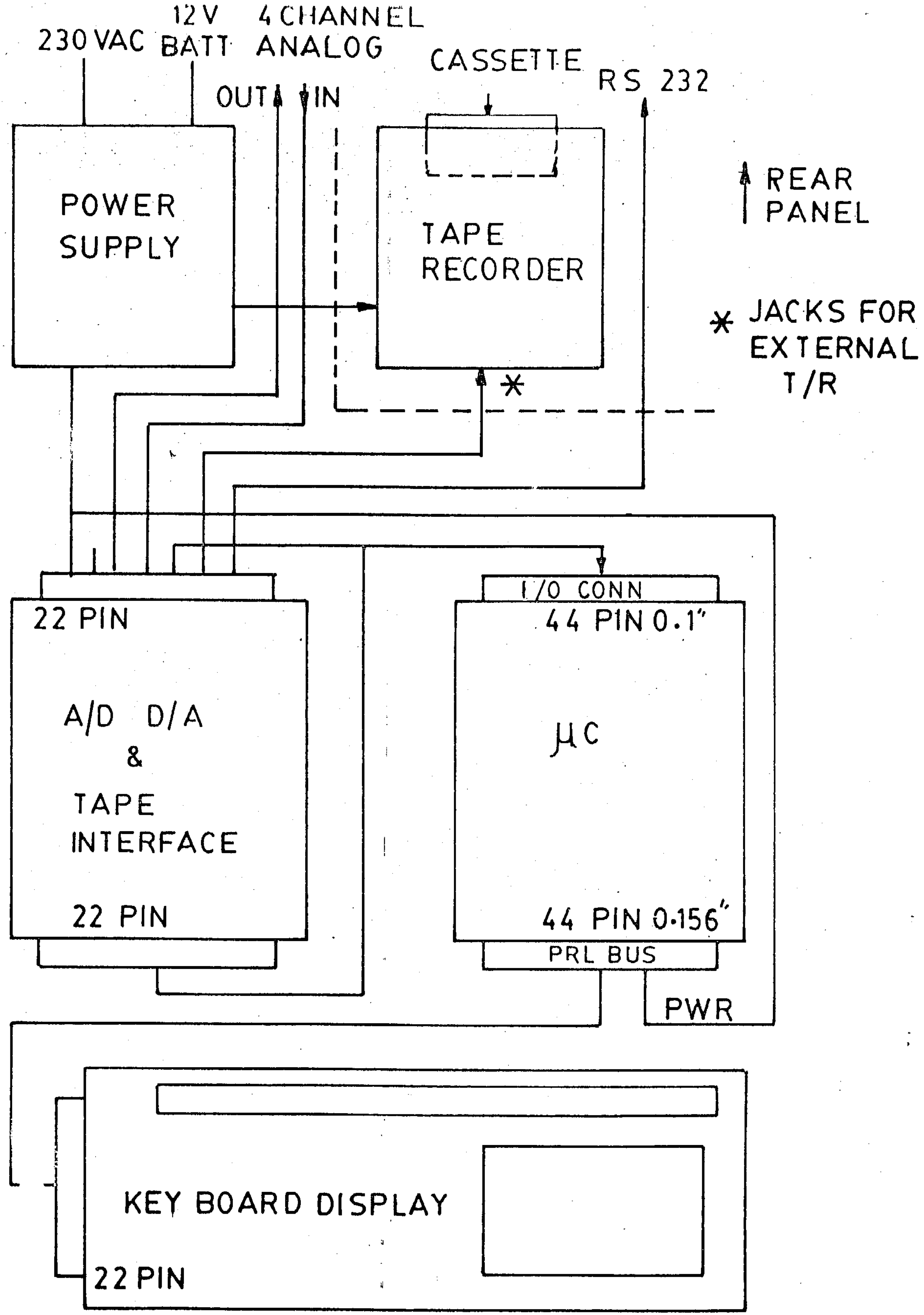
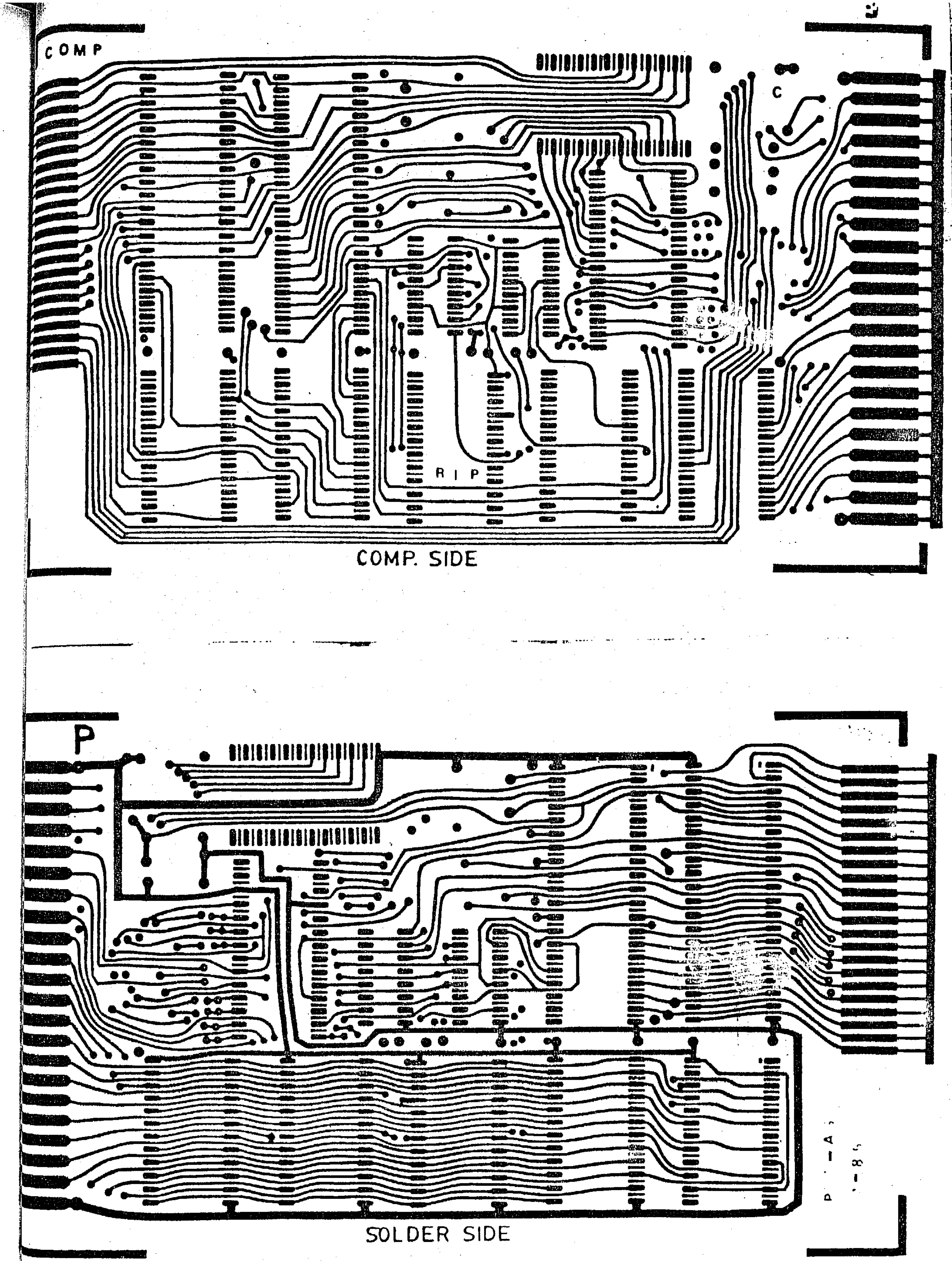


FIG.-10 APPROX LOCATION OF UNITS IN ENCLOSURE & WIRING PLAN



COMP. SIDE

SOLDER SIDE

FIG. 11. PCB LAY-OUT. OF MICRO-COMP.

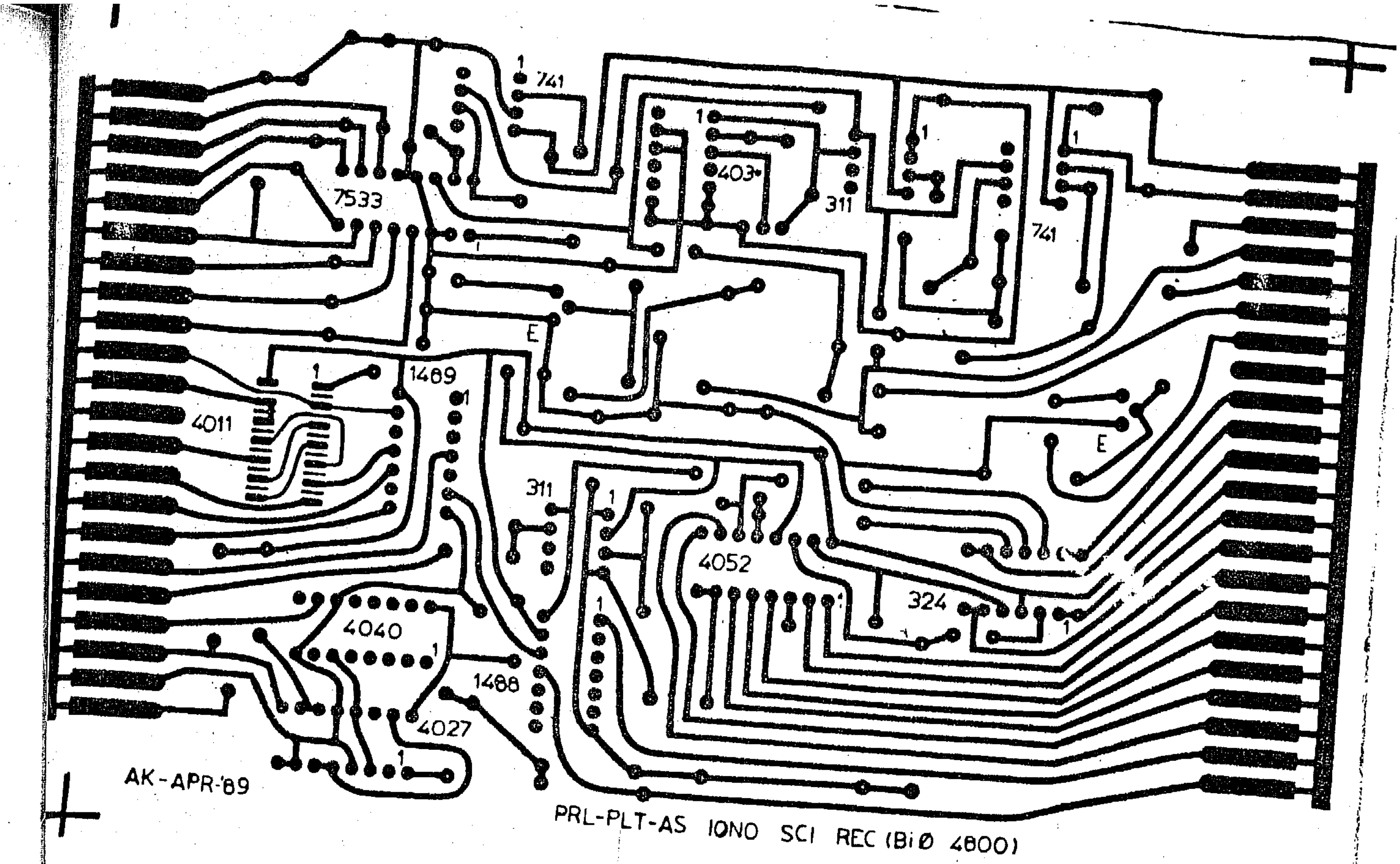
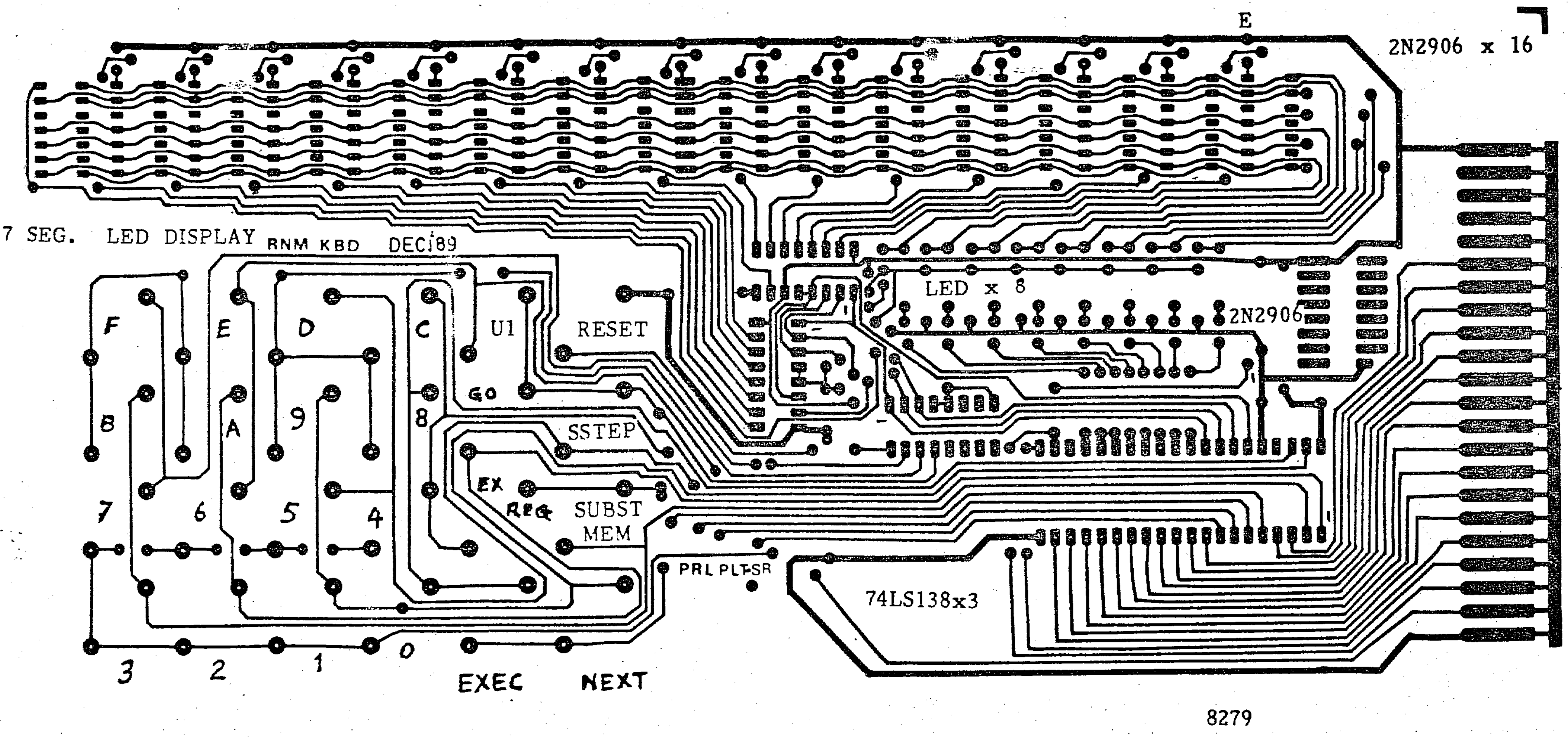


FIG. 12. PCB LAY OUT OF D/A CARD

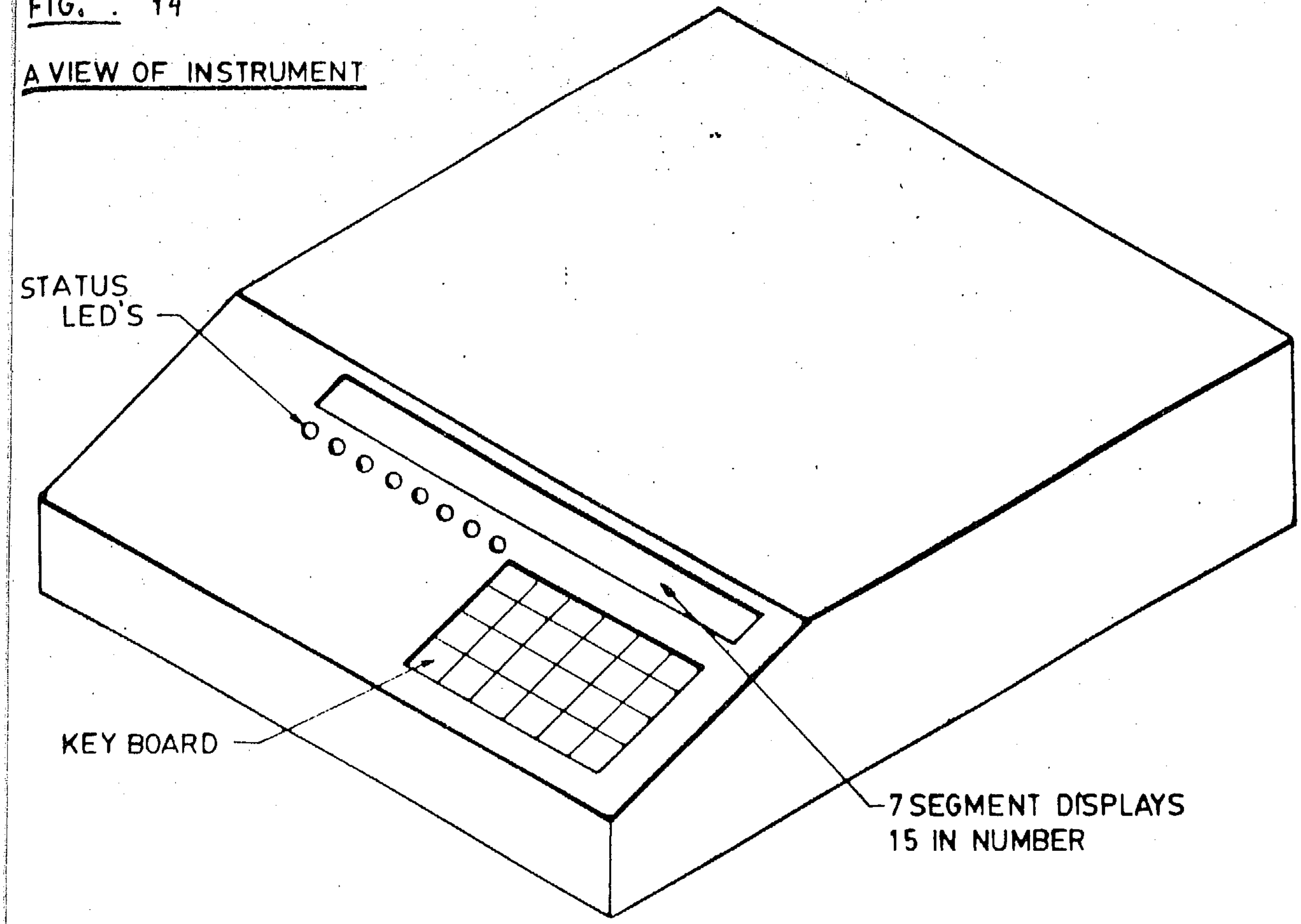


NOTE : JUMPERS NOT SHOWN

FIG. 13 KEYBOARD/DISPLAY CARD (SOLDER SIDE)

FIG. 14

A VIEW OF INSTRUMENT



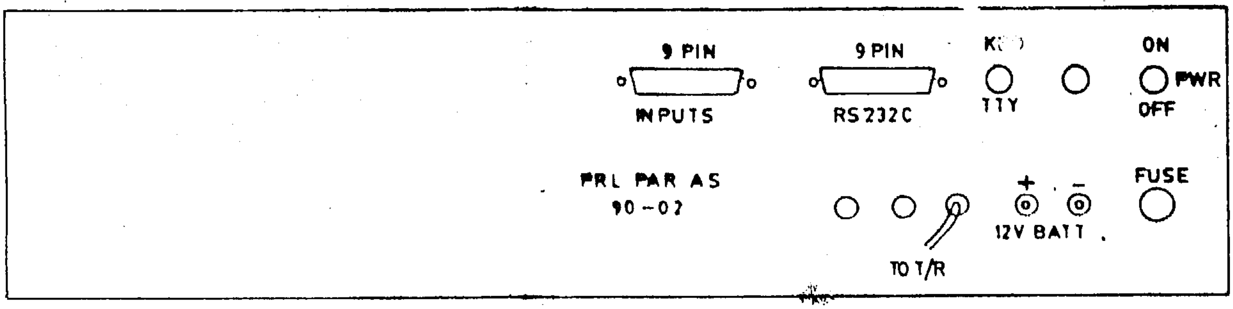


FIG.15 REAR VIEW