P R L

T E C H N I C A L N O T E

TN-95-76

FLOPPY DISK BASED FOUR CHANNEL DATA RECORDER

R. N. MISRA

MARCH 1995

PHYSICAL RESEARCH LABORATORY
NAVRANGPURA
AHMEDABAD 380 009

DOCUMENT CONTROL AND DATA SHEET

1.	Report No.	:	PRL-TN-95-76
2.	Title and sub-title Floppy Disk Based Four Channel Data Recorder		3. Report Date : March 1995 4. Type of Report Technical Report 5. Pages : 89 6. Price : Unpriced 7. No. of References : 5
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 instrument uses simple hardware to record data on floppy disk in Sector serial/Track serial Format. The 8085A microprocessor controls the data acquisition process and issues necessary commands to the Floppy Disk Controller. At a sampling rate of 10 per second, 360K floppy would last for 10 hours. The instrument, in, addition, checks for presence of scintillations in data and selectively records the same.
13.	Key Words	:	Data Recorder, 8085A, Instrumentation, Floppy Disk
14.	Distribution Statement	:	Distribution unlimited
15.	Security Classification	:	Unclassified

CONTENTS

APPENDICES

ILLUSTRATIONS

FLOPPY DISK BASED FOUR CHANNEL DATA RECORDER

Data recorders utilizing audio cassette as storage medium have been extensively used at PRL for recording of Ionospheric Scintillations for the last few years.Although the cassette is a cost effective as well as reliable medium , the processing of data becomes easier if the same is transferred to other commonly used media.It could be **Floppy Disk** , for use in PC's and the half inch computer compatible tape , in case of mainframe computers. Eventhough the latter medium has been used for quite some time , the gradual shift towards the PC has tilted the balance in favour of the floppy disk.The design of the four channel data recorder ,has therefore been altered to accomodate the floppy disk drive instead of the audio casette recorder.The present design caters for the 5.25 inch drive of the double sided double density (DSDD) type storing 36ØK bytes and the same hardware can also be used to store 72ØK bytes of data on 3.5 inch floppy on the corresponding drive with only one byte software change.Details of the floppy disk drive interface and data format have been described in the text.

Keywords:Data Recorder,Magnetic media,Floppy Disk Drive,Floppy Disk Drive Interface.

1.Ø INTRODUCTION

Data acquisition process is inherently sequential in nature,and accordingly , a serially organized memory or storage medium is the most suitable one for data storage in such applications .Thus the audio cassette and computer compatible half inch magnetic tape(CCT) would be the ideal medium. However ,the half inch tape drive is an expensive peripheral costing beyond Rs one lakh, as well as difficult to maintain. Even the quarter inch data cartridges as well as their drives are not very economical as far as cost of drive and medium is concerned . Though the cost per bit is low,the numerous formats and sizes in use for the data cartridges and drives lead to compatibility problems . The widespread use of floppy disks in the PC 's has resulted in easy availability of the disks as well as floppy disk drives(FDD).Although the floppy disk drive was introduced by IBM for random access operation ,the data can be organized serially and FDD used for data aquisition in a manner somewhat akin to a CCT in free format.Data organized in byte serial /sector serial /track serial format gives a feel of serial memory and retains the compatibility with the PC with the same number of tracks, sectors and sector size.Although the compatibility is only at the BIOS level ,it is not a serious handicap and the use of binary format in place of ASCII doubles the capacity of the floppy disk.The data is recorded track wise with header provided to each track and file structure can be provided by the PC during readout.

2.Ø DESIGN

The block diagram of the four channel data recorder has been given in fig.1.It comprises of the following subsystems :

1. Logic Board containing the A/D converter , multiplexer and microcomputer.
2. Keyboard and Display.
3. Floppy Disk Drive (FDD) .
4. Floppy Disk Drive Interface .
5. Power Supply.

2.1 Logic Board:

This board had been designed for use with audio cassette recorder.In the present system , however ,the circuitry relevant to the cassette recorder is redundant and may be left unwired . The block diagram of the microcomputer has been given in fig.2. It is based on the 8Ø85A microprocessor and contains 24 Kbytes of RAM ,8 Kbytes of EPROM,glue logic, USART type 8251A,I/O port type 8255, D/A converter type AD 7533 and analog circuitry along with the input output multiplexer.The monitor and the data acquisition programs are contained in two separate EPROM's.The 8253 timer has been used to generate timing pulses.The D/A converter has been interfaced to the B port of 8255.The output of the D/A converter is compared with the input signal and the comparator output sensed at PAØ. The successive approximation algorithm has been implemented in software .The USART alongwith the CMOS logic is intended to be used for recording on audio casstte and may therefore be left unwired . The circuit diagram of the microcomputer has been given in fig.3.

The logic board uses PRL BUS based on 44 pin double sided connectors of Ø.156 inch spacing, the details of which have been given in Appendix N. The FDD interface also conforms to the same bus and a single sided passive backplane has been used for interconnecting it to the logic board. The memory and I/O map of the microcomputer has been given in fig. 4 , and the circuit diagrams of memory, I/O, D/A converter and RS 232 C interface in figs. 5 to 7.

2.2 KEYBOARD AND DISPLAY

The keyboard cum display card is mounted just below the sloping front panel. It contains 16 displays of 7 segment type and 24 SPST keys along with a set of 8 LEDs . While the first 15 displays of 7 segment type are used for displaying the station code, time etc. during the program run and ,address as well as data during the data loading or initialization. The 8 LEDs ,on the other hand are used to display number of tracks of the floppy used up in the binary format on the 6 LEDs on the right hand side , and the multiplexer operation is indicated on the leftmost two LEDs . The blip of LEDs indicates healthy operation of the system. The card uses keyboard display controller type 8279 and is interfaced to the main logic board through a flat ribbon cable to the backplane bus. The 8279 is located at 8ØH on the I/O map and the circuit diagram of the keyboard display card has been given in figs. 8 and 9 . The PCB lay out of the Keyboard Display card has been given in fig.2Ø and that of the backplane in fig.16.

2.3 The Floppy Disk Drive:

The FDD is a standard peripheral and 5.25 in. DSDD drive storing 360K bytes has been used in the design. However, the newer 3.5 inch FDD can also be interfaced to the recorder with same ribbon cable except for the power connector and one byte change in the software.

The floppy disk consists of a thin , flexible , circular shaped mylar sheet of 40 micron thickness over which powdered iron oxide is coated on both the sides, somewhat similar to the audio magnetic tape. A large hole punched in the center of the disk, which is strengthened by hard collar , serves to hold the disk in the hub on the drive spindle .The data is recorded in concentric cicular tracks , and the number of tracks is dependent upon the mechanical precision of the positioning mechanism which moves the read/write head along the radius of the disk .The head positioning arm is spring loaded so that the disk is kept in slight pressure by the head while the disk itself is held in the hub .The two heads are on either side of the disk and mounted on a common head assembly.The head gaps are offset a bit to reduce interaction between them.The precision attainable today provides for a track densities of 48 per inch (48 TPI) for DSDD disks, 96 TPI for double sided high density (DSHD) and 135 TPI for microfloppy disk of 3.5 inch size.The head positioning mechanism is driven by stepper motor and the required pulses are to be provided by the FDC electronics.A brushless DC motor is used to drive the central hub to provide the disk with spinning motion.

Interchangeability between the disks as well as drives of the same type is the basic requirement ,which , in turn decides the manufacturing tolerance of the disks as well as drives.The DSDD disk has 40 tracks while the DSHD disk of 5.25 inch type as well as the 3.5 microfloppy have 80 tracks.The reference point for the tracks is established by an index hole punched in the floppy near the hub before the start of the recording area.The 5.25 inch disks ,both the DSDD as well as DSHD types, are enclosed in flexible plastic cover, lined inside with soft fabric , to provide smooth surface to the flexible disk and cleaning during use .Three openings are provided in the jacket , first being the central window for holding the disk in hub , a smaller hole corresponding to the index hole in the disk being the second one and a third oblong window to provide access to the read /write head on the disk surface , as shown in fig.10. The microfloppy is enclosed in the hard plastic case which has corresponding holes for hub as well as index hole .The read /write window has been provided with a spring loaded metallic cover to protect the disk from dust. The window opens only when the disk is loaded in the drive.

The disk purchased from the market is virgin , without any magnetization whatsoever, and while the number of tracks as well as their positions on the disk are decided by the drive mechanism specifications.the tracks themselves are subdivided into several sectors. The data is partitioned into data blocks, equal to the sector size , prior to recording .The handling of data is simpler if it is divided in to blocks of manageable size.The sectors were identified, once upon a time, by one hole for each sector , and such floppies called the hard sectored disk.Now these types of disks are obsolete and the sectors are defined by software and sector beginnings identified by sector and track numbers written in the first few bytes of the sector data as a header information.These are therefore called soft sectored disks and their use is universal.

The DSDD disk of 5.25 inch size can store a total of 360 kbytes of data in 9 sectors of 512 byte each on each track.One track on one side of the disk would store 4608 bytes and the same numbered track on the other side an equal amount.The two tracks on the two sides of the disk having the same track number are referred to as a cylinder. Thus a cylinder would store 9216 bytes and all the 40 cylinders taken together would total to 9216X40 or 368640 bytes . As the 1024 bytes of data are referred to as 1 K in computer parlance , the disk capacity is referred to as 360 K bytes.It is possible to choose different sector size and number of sectors from a set available in the floppy disk controller parameters but compatibility with IBM PC requires the use of 9 sectors of 512 bytes each in DSDD and 15 sectors of the same size on DSHD disk of 5.25 inch .The available format for 3.5 inch microfloppy is of 9 sectors or 18 sectors of 512 bytes.The 5.25 inch diskette of DSHD type would store 1.2 Mbyte and the 3.5 inch microfloppy, 720 Kbytes in 9 sector format and 1.44 Mbytes in 18 sector format.

The data is recorded at the rate of 250 kbps in DSDD format and 500 kbps in DSHD format and the same corresponding rates are chosen for the microfloppies in 9 sector and 18 sector formats respectively .The 25Ø kbps data rate means a time period of 27 microseconds per byte of data. As the rotational speed of 5.25 inch DSDD disk as well as the 3.5 inch microfloppy is 3ØØ rpm , one index pulse is generated every 2ØØ milliseconds in both the types. The DSHD disk of 5.25 inch spins at 36Ø rpm and generates index pulse every 166 milliseconds. As said earlier ,the disk available in the market is without any recordings and has to be formatted before use. The process of formatting involves writing of header information in the beginning of every sector and filling the data zone by hex character F6. The header information is read during the read or write operation and the data transfer is initiated only if the desired sector /track can be identified. The same data rate holds good for the microfloppy and it can store 72Ø Kbytes in 8Ø tracks on both sides, and each track subdivided into 9 sectors of 512 bytes .The microfloppy can be used for high density recording at 5ØØ kbps data rate to store 1.44 Mbytes of data on 8Ø tracks on both the sides, each subdivided into 18 sectors of 512 bytes. The data rate in this case becomes 13.5 microseconds per byte.

It is difficult to handle such a high data rate in the interrupt mode and therefore ,the direct memory access (DMA) mode ,is almost invariably used for transferring data to and fro the floppy disk. Close scrutiny of the data transfer operation in floppy disk reveals that it is just possible to handle one data byte every 27 microseconds by software loop ,and therefore, the 36Ø kbyte floppy of 5.25 in. size as well as 72Ø kbyte floppy of 3.5 in. size can be interfaced to a microprocessor by simple hardware and the same has been successfully implemented.

2.4 FLOPPY DISK DRIVE (FDD) INTERFACE

The FDD interface is centered around the popular floppy disk controller type 765 of NEC to which 8272 of intel is also equivalent. The 765 has been interfaced to the PRLBUS as I/O port and operated in non DMA mode. This mode, as explained earlier, is hardware wise simpler and can be used with DSDD floppy disks. The block and circuit diagrams of the system have been given in figs.11 and 12. The FDC generates control signals for the FDD and receives the outputs from it. All the signals towards the FDD are buffered through the inverting open collector buffer type 7406/7416, while the signals from the FDD are inverted through the Schmitt trigger inverter type 7414 to reduce the effect of slow rise times. The FDC is interconnected to the FDD by means of 34 pin flat ribbon cable the pinouts of which have been given in Appendix O.

An 8 MHz crystal oscillator, designed around 7404 inverter, forms the basic clock. It is subdivided in a 4 bit binary counter type 7493 to derive 4 MHz clock for 765 and clock pulses of 2 MHz, 1 MHz and 500 KHz. The latter have been used to derive the write pulses of 250 nano seconds width at 500 KHz rate, corresponding to the data rate of 250 Kbps. Clock of 8 MHz for 765 and 250 nsec pulses at 1 MHz can be selected with help of DIP switches for use with 1.2 M or 1.44 M floppies. The circuit details and the waveforms of the clock circuit has been given in fig 13. The binary counter 7493, driven by the 8 MHz clock, generates 4 MHz, 2 MHz, 1 MHz and 500KHz square waves, indicated by A, B, C, and D. The output C is connected to the J input of one of the J K flip flops in 7473 while the output C is used as clock for it. K input as well as preset are connected to logical 1.The clear input of the same is pulled up through as resistor and may be left high or driven by output D at 500 KHz.The 7473 is rising edge triggered flip flop (FF) and therfore its output changes at the falling edge of the clock.When J is high and CLR is kept high , two falling edges are available at clock input and hence one 250 nsec pulse is generated every 1 MHz i. e. at the rate of C output. In the other case when the CLR input is connected to D output ,the Q output remains low along with D and the output at Q is generated only when D output is logical one .This results in 250 nsec pulse being generated at the rate of 500 KHz.

The FDC is located at F0 in the I/O map and a 6 bit register implemented by latch type 74174 has been located at F2.Bit assignment of the register has been given in Appendix G. While bit 0 has been used to select drive 0.bit 1 controls the drive motor for the same.Bit 4 and bit 5 have been used for selection of drive 1 and its motor on function respectively.The DS0 and DS1 signals of the FDC have not been used to control the disk drives for the sake of hardware simplicity . It is worth noting that the same scheme has also been used in the PC.The bit 2 and bit 3 have been used for generation of reset and TC (terminal count) signals to the FDC.

The read data available from the FDD requires data separation and the same has been implemented by the D flip flop type 7474 used in conjunction with 4 bit synchronous counter type 74161 to provide data synchronization .The write compensation circuit is based upon the 4 bit latch 74175 and dual 4 bit multiplexer type 74153 . Multiplexed control signals LCT/DIR (low current /direction),FLT/TRKØ (fault / track ØØ) ,WP/TS (write protect / two sided) and FR/STP (fault reset / step) have been demultiplexed by means of logic gates controlled by RW*/SEEK signal.All the incoming signals from the FDC have been provided a pull up resistor network. The circuit is an adaptation of the hardware described in detail by Padmanabhan et. al. (3).

A standard card for the PRLBUS is of 111X 18Ø mm size ,and as the total circuit required for the FDC is not very large ,some space left on the PCB has beeen utilized for a set of 8 bit latches and an input port to be used for EPROM programmer in future.One of the output ports can be tristated and is joined to the input port, back to back ,to serve as bidirectional port or data port in EPROM programmer.The other two ports are permanently wired as output to serve as address ports. One extra latch of 6 bit width and realised by 74174 has also been provided in the same group. While one of the bits of this 6 bit latch is used for enabling /disabling the data port ,the other 5 bits are available for control purposes,.The port lines have also been brought out on the 34 pin edge connector as in case of FDC.The pinouts have beeen given in Appendix P.The port map of the FDC as well as EPROM programmer section has been given in appendix G. The circuit diagram of the EPROM programmer section has been given in fig.14. The PCB layout of tne FDC/EPROM card has been given in fig.21.

2.4 DATA CONVERSION AND RECORDING:

The 10 pps pulse generated in the counter chain interrupts the microprocessor and the interrupt service routine , in turn, updates clock, initiates A/D conversion , loads the data in the RAM buffer and updates display. Length of the data buffer is 9 K bytes (1 K =1024 bytes) and the same amount of data is transferred to the FDC in one stroke to write one cylinder consisting of two tracks on either side of the disk.Since the recording is done one side at a time it would take 0.2 second per side i.e. 0.4 seconds to store total contents of the buffer.The data acquisition process has to be held up during the data transfer to the FDC as the data transfer is performed under software loop and microprocessor is fully busy.The loss of time due to this can be compensated to some extent by advancing the clock by the same amount i.e. 5 pulses.Break in the data is inevitable but the discontinuity of 1/2 second between the data blocks of 900 seconds has been found to be not of much consequence because data lengths of 200 or 400 seconds are adequate for analyses of scintillation index or spectrum .The floppy disk drive is enabled ,and the drive motor switched on about one minute before data storage operation.

The read/write head is repositioned to track 00 and then brought over the required track , 30 seconds prior to actual record operation ,to minimize error due to misalignment between data blocks during switch off .It is possible to perform some mathematical operations on the data block before starting the data recording. Since the FDD has to be ready before hand , the operations have to be limited to the first 32 blocks of data i.e. 8 K bytes. Operations like calculation of the scintillation index S_4 can be performed before taking decision about the usefulness of data to be recorded.

2.6 POWER SUPPLY

The instrument runs on 230V 50Hz mains supply and draws about 20 watts of power. The voltages required by the system include +5 volt for digital logic, +12 and - 12 volts for A/D converter and operational amplifiers and also +12 and +5 volts for the FDD. Provision for battery back up by a lead acid battery of nominal 12 volts has been made.

Raw DC of approximately 17 volts is generated from the mains supply by means of step down transformer of 12 volt, 2 ampere type with associated rectifier and filter capacitors. The raw dc of 17 volts has been utilized to drive a switch mode power supply (SMPS) to generate the 5 volt logic power. The pulse width modulator (PWM) type TIL 494 has been used as a control element. The -12v has been obtained from the extra winding on the ferrite core choke used in the +5 volt supply. The efficiency of the power supply is about 70% and ripples on +5 V and +12 V about 100 millivolts. Drain on -12 V line is very low.

The +17 volt raw DC is isolated from the SMPS with the help of a pair of diodes in parallel to increase the current rating, and in case of power failure the battery takes over and supplies 12.6 volt to the SMPS through another pair of diodes. Indicators have been provided to disply AC/Battery mode of operation. The +12 volt line has been generated by means of a low dropout regulator so that it can give nearly 12 volts even on battery mode. During

AC mode this regulator is equally comfortable.The battery is kept under trickle charge through the green LED.Frequency of operation of the SMPS has been set to 2Ø Khz by means of R and C in the PWM circuit. The circuit diagram and the PCB layout of the power supply have been given in figs. 15 and 16 respectively.

3.Ø SOFTWARE:

The total software may be subdivided into three parts , the Main Module ,the Interrupt Service Routine and programs related to detection of occurrence of scintillations alongwith S4 index estimation. The programs related to floppy disk operation have been described elsewhere and therefore not covered in detail here.

3.1 MAIN MODULE

The station code ,year ,day of the year , and time of the day is entered in the RAM following the procedure given in the Appendix F .The data aquisition program runs at 21ØØH up and has to be copied to this location from the EPROM located at 6ØØØH. A small program located at 63AØ can be run through the key board for this purpose.The main program starting location is at 22AØ.It initialises the ports ,timers as well as FDC.Memory pointer is initialised to 3ØØØ and the header information from 2ØA2 -2ØAB is copied to 3ØØØ -3ØØ9 .The next 6 locations are loaded with ØØ to serve as data mark.The interrupt RST 7.5 is enabled and the microprocessor starts responding to the 1Ø pps pulse train .The data acquisition program starts and it checks for the Hi byte of the buffer address.As the start address of the buffer is 3ØØØh, 32 blocks of 256 byte each would have been recorded when the high byte reaches 5Øh.The total buffer size is

9 K bytes or 36 blocks.Scintillations do not occur all the time and therefore S_4 index is calculated on the 32 blocks of data and tested if the scintillations are above 3% of the average.(section 3.4).If scintillations are present the FDD is enabled and the LED on the drive would glow and motor starts . This is done ahead of time to allow FDD to attain proper rotational speed of the disk.

The data acquisition process continues and after one more block has been aquired the drive is recalibrated i.e. the head is brought to track ØØ.On completion of next block the head is brought to the desired track by issueing SEEK command.The completion of last block halts data aquisition and S4 index is calculated for the previous 4 blocks, 33 to 36, and stored.Now one full cylinder is recorded ,one side at a time, and track number incremented by one.The track number is checked for, 4Ø i.e. disk full and the program halts if it is so. Otherwise the data acquisition and recording cycle continues.

In case scintillations are not present the buffer is reinitialised after 32 blocks and calculation of S_4 without enabling the FDD and a new data cycle initiated .In the absence of scintillations, the data acquisition would go on but data would not be recorded. The flow chart of the Main and record Modules has been given in Appendix A.

3.2 INTERRUPT SERVICE ROUTINES

The interrupt service routine consists of three subroutines , namely, Clock , A/D Converter and Display , executed in turn on a single interrupt request pulse, 1Ø times a second.

3.2.1 Clock and Display

The clock subroutine updates a realtime clock , while the displdy subroutine displays the header information on a set of 15 displays of 7 segment type.Alphanumeric data on the 7 segment displays is represented in special shapes and is reasonably understandable after some familiarisation.The status bits are displayed on the 8 LEDs.These form a group and are handled as another 7 segment display .This set of 16 displays as well as keyboard consisting of 24 keys, including one for reset , are controlled by the Keyboard Display controller type 8279.The display subroutine updates the display every Ø.1 seconds.

3.2.2 A/D Converter

The A/D converter subroutine implements successive approximation algorithm in software .The analog signal generated at the output of D/A converter is compared with the input signal and the comparator output sensed at one of the pins of the input port.The D/A converter is driven by 8 bit output port.The conversion process takes 8 software loops .The result is stored in the current location of the buffer,and the buffer pointer incremented.The buffer length is 9 K bytes and the data is recorded on the floppy after the whole buffer is filled up. The flow charts of the clock disply and A/D conversion subroutines have been given in Appendix B.

3.3 DETECTION OF SCINTILLATIONS

As ionospheric scintillations donot occur all the time , it is worthwhile to check for their presence by software before storing the data on floppy.Software has also to cater for the delays required for starting the FDD motor and locating the head on the desired track. These are therfore initiated well before the actual recording of data . S_4 index is calculated on the 8 k bytes of data collected in one cycle , and the block size for calculations is kept 256 bytes for ease of calculation on an 8 bit microprocessor.

As the first data block starts at 3010H, end of the first block is taken as 3110H and S_4 calculated. The next block is marked from 3100 up. The result of calculations consist of 5 bytes for each block and are stored in locations 3010h up . Thus 160 bytes are used up in this process and the first block data is obliterated. Presence of scintillations is indicated by bit 7 of memory location 20B4 being reset to zero. After all the 32 blocks of data are screened for , The bit 7 is checked for 1. If it is so ,the scintillation index is lower than 3 percent and the buffer pointer is initialised once again ,to start the data acquisition cycle afresh without even switching on the FDD. If Bit 7 is 0 ,the drive is selected and a delay time introduced for FDD to stabilize. The data acquisition process is concurrent and continues even during this delay period. S4 index for the blocks 33 and up is not calculated at this juncture. After acquiring one more block of data ,the FDD motor is turned on and delay introduced. The drive is recalibrated after the current block is over and the R/W head positioned over the current track. The last block is being filled this time and the end of this block stops acquisition process .

The S_4 index is calculated for the last 4 blocks but the results are not used for taking decisions for recording. The results are stored at locations next to that of the 32nd block in the memory area 3010H to 30FF. Now full track is recorded one side at a time and motor turned off subsequently. The buffer is reinitialised and the whole recording cycle repeats itself. The track number is incremented and stored at 20B7h. If the track number becomes 28H i.e. 40, the program halts. The last track incase of 3.5 inch floppy would be 50H.

3.4 ESTIMATION OF S_4 INDEX

The S_4 index defined by Briggs and Parkin (5) can be determined from the signal time series by the following expression:

$$S_4 = [(\sum x^2 / n) - (\bar{x})^2]^{1/2} / \bar{x}$$

where n is the number of elements of x.

This expression seems to be difficult to compute on an 8 bit microprocessor without arithmatic processor or multiplier chip. However since the number of samples n can be chosen to be a power of 2, division by 2 to the power m would involve only m bit shift to the right, and in case m is chosen to be 8 i.e. if the samples are 256 in number, then the sum of 256 elements would result in a quantity expressed in 16 bits. Division of this 16 bit quantity by 256 means just using the most significant byte of the result and discarding the lower order byte. Thus choice of n=256 simplifies the operations dramatically. What remains is to calculate sum of squares $\sum x^2$ and sum of the series $\sum x$. While the former is of 3 byte length, the latter is defined in two bytes only. The first two bytes of the sum of squares represent the first term of the expression within the parantheses, the high byte of the sum of the series gives value of $\bar{x}$.

The problem of generation of squares of binary numbers has been solved by preparation of look up table for squares for all numbers from ØØ to FFH. Since the square of a number requires 2 bytes storage space, a total of 512 byte memory space would be sufficient for the square table. The table may be prepared by first principles before hand and stored at some location in the memory. The sum of squares requires 3 bytes as stated earlier and sum of the series two bytes. These are stored in the block 1 of the buffer to be used for calculation of S4 index at a later stage if so desired. The square root operation can be avoided for estimation of S_4, just for taking the decision whether to record or not to record the current buffer contents. The

S_4 index in percent value, may be conceived of as a binary ratio. Ra difference of the average of squares $\sum x^2/n$ and square of average $(\bar{x})^2$ to the square of average $(\bar{x})^2$, is in fact the square of S_4. It would be of the form 1/n for S_4^2 equal to 1/n. Now n can be chosen to be a binary number in the form of 2^m. For n=16 the S_4 would be about 6 percent. It only means that the denominator is 256 times the numerator. Thus the denominator i.e. the square of the average is to be compared with the numerator multiplied by 256 or else the lo byte of the numerator. Two bit shift to left for the numerator means multiplication by 4 or a limit of 3 percent for the S_4 index.

Value of S_4 being larger than 6 percent may be verified by 1) non zero contents of the hi byte of the numerator or 2) in case the hi byte is zero, lo byte being larger than the high byte of the denominator. The same logic applied after shift to left for the numerator ,would mean testing for a S_4 index limit of 3 percent.The flowcharts of the subroutines have been given in Appendices C ,D and E .

The table of squares is generated in the initialization phase.Square of a number is generated by adding it to itself the same number of times.Since it is performed only during initialization ,time taken for it does not matter much.The squares of 00 to FFH are stored sequentially at memory locations incremented by 2 bytes, offset from the beginning of the table being the actual number multiplied by 2 .Square of any number between 00 to FFH may be simply read out from the table by giving an offset equal to two times the number under question , to the starting address of the table.

Software to fill a certain area of RAM by a simulation signal consisting of triangular wave superimposed on DC has been incorporated .The minimum amplitude of the simulated signal , is decided by the contents of A register, increment per step by contents of E register and the number of steps by the contents of D register.It can be used to test the various sections of the software.

To summerise ,the average of data is calculated by adding the 256 elements and retaining the hi byte only. Squares of numbers are generated by table search and as the squares themselves occupy two bytes ,the sum of squares would occupy 3 bytes. The difference between sum of squares and the square of average is to be compared with the latter.If the difference is more than 1/1024 of the square of the average,it would indicate S_4 larger than 3 percent and the recording is to be made.Data between 0-300 millivolts also need not be recorded as it is practically below noise level. This condition is checked by average value which is stored in register pair DE. Contents of D being zero means signal very low. The sum of squares as well as the sum of data are stored at locations starting from 3010H and above. The first block starting at this location is overwritten. The header remains intact however. The subsequent blocks are selected at even locations of 3100 with interval of 256 bytes. The total number of blocks tested are 32. The S_4 is tested for being larger than 3 percent and if found so the bit 7 of location 20B4H is reset to zero.

4.0 CONCLUSION

A floppy disk based four channel data recorder has been designed and has been in use for recording of ionospheric scintillations since the last one year. It is an upgrade of the older model based on audio cassette. The present design utilizes a 360K DSDD floppy disk in byte serial/sector serial/track serial mode to store data in binary format. It has facility to format the floppy disk, w/o header and name, in the rudimentary fashion. Data block length has been selected to be 9 K bytes including header, to facilitate full cylinder write operation to save on time. Data discontinuity does occur during write operation but the time loss is compensated for by updating the clock. Data gap has been found to be of negligible significance for the present applications. The floppy may be read in a PC with the help of suitable software.

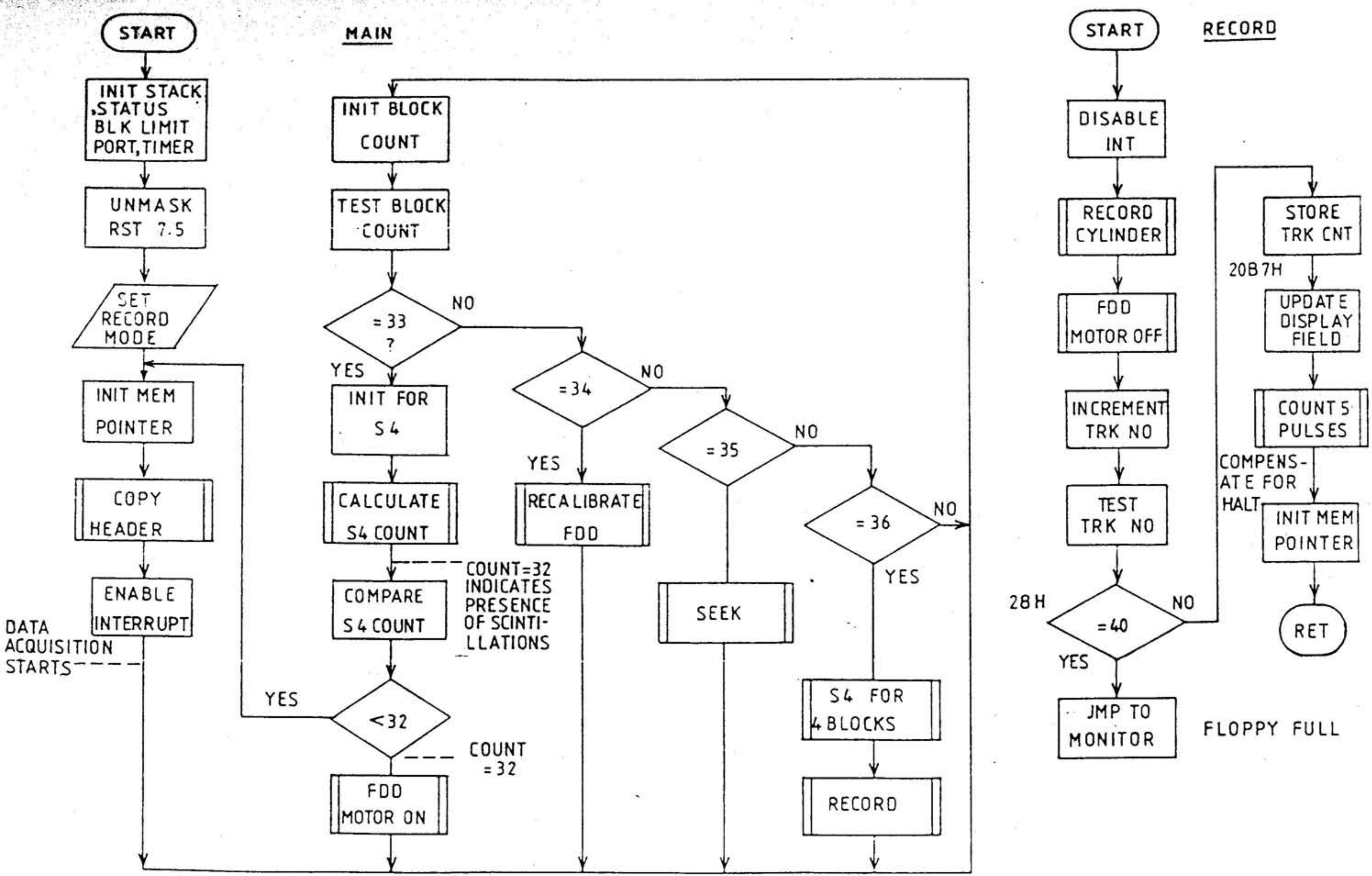
5.Ø ACKNOWLEDGEMENTS

The algorithm for the calculation of S_4 index was suggested by Prof. H.Chandra and the instrument was thoroughly tested by Dr. G.D.Vyas . The author gratefully acknowledges their contribution.The SMPS design was implemented by Messrs M.B.Dadhania and R.I.Patel.Thanks are due to Messrs H.D.Parikh and G.A. Panchal for their help in fabriction of the instrument and preparation of the document.

7.Ø REFERENCES

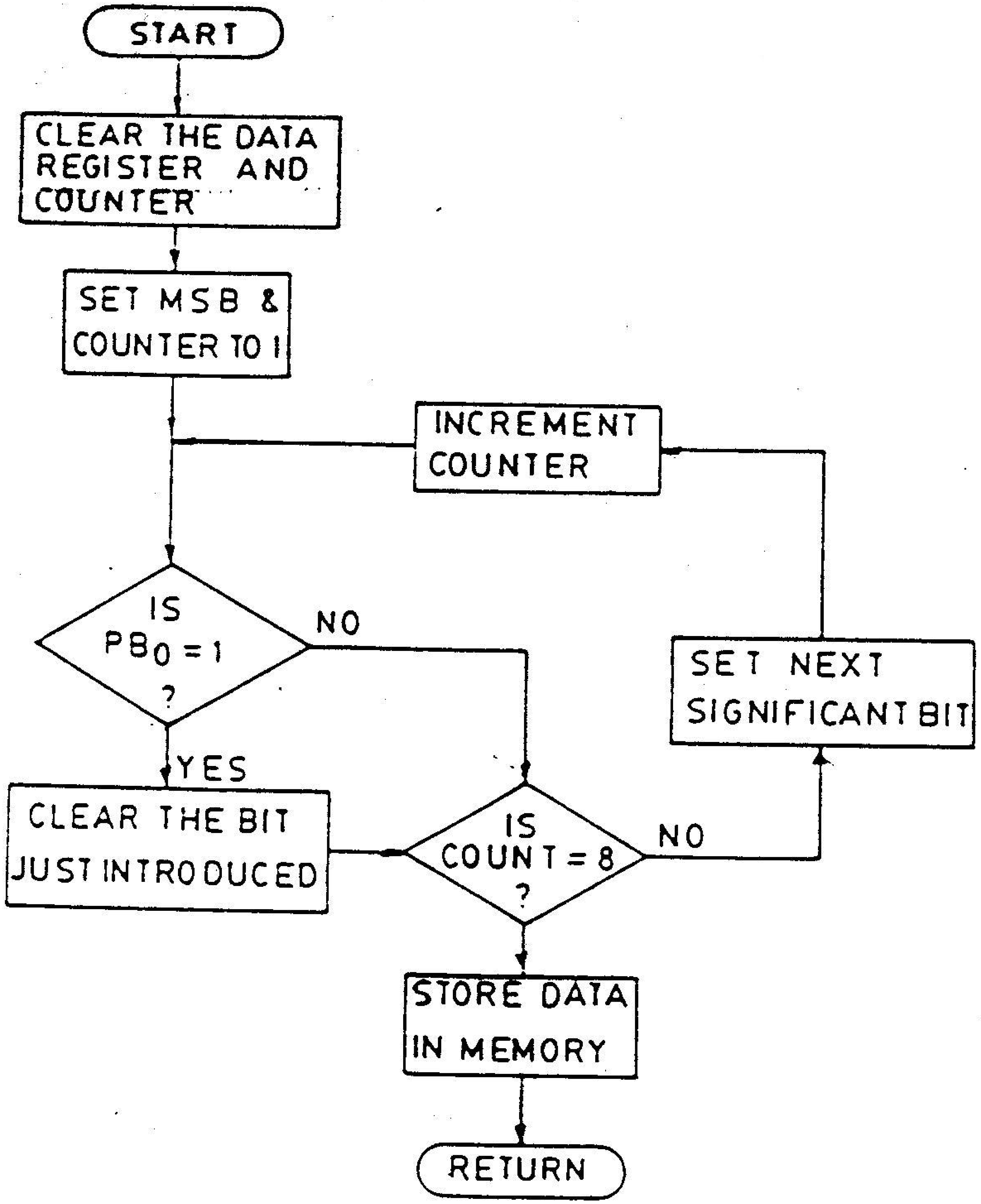
1. "Four Channel Audio Cassette based Data Recorder" R.N.Misra, PRL Technical Note ,TN 91-71, January 1991
2. "Software Design and Implementation of Floppy Disk Subsystem", Microsystem Components Handbook,Vol.2, 1985, INTEL Corporation, 3Ø65 Bowers Avenue,Santa Clara,CA95Ø51, USA
3. "Make Yourself an IBM Compatible PC Part VII, Floppy Controller,K.Padmanabhan et. al., Electronics For You July 1989.
4. "Floppy Disk as Data Storage Medium", R.N.Misra,PRL Technical Note (In Process)
5. Briggs , B.H. and I.A. Parkin , J A T P , vol 25,pp 339,1963

APPENDIX A

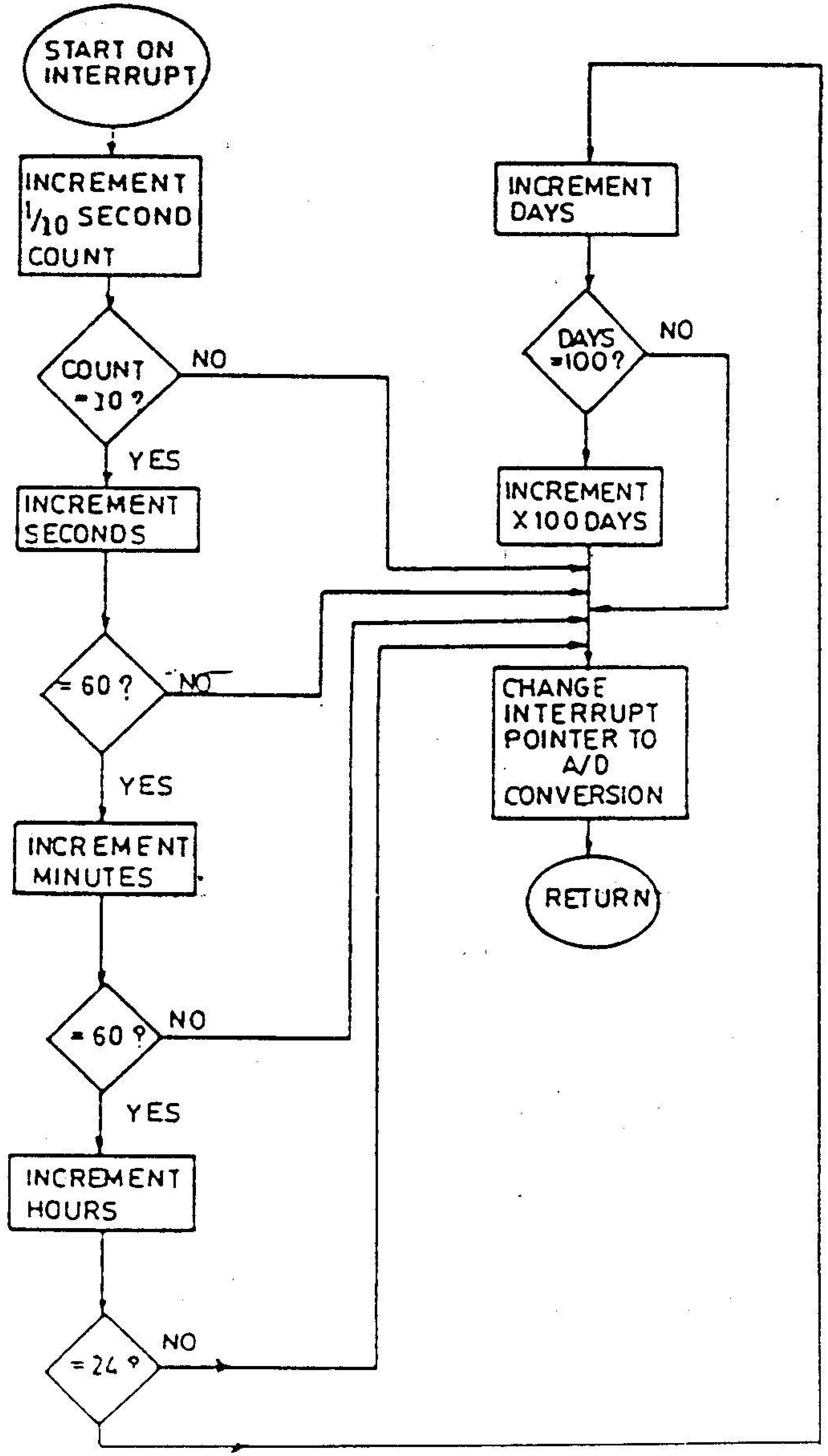

START
MAIN
INIT STACK ,STATUS BLK LIMIT PORT,TIMER
UNMASK RST 7.5
SET RECORD MODE
INIT MEM POINTER
COPY HEADER
ENABLE INTERRUPT
DATA ACQUISITION STARTS
INIT BLOCK COUNT
TEST BLOCK COUNT
=33 ?
YES
NO
INIT FOR S4
CALCULATE S4 COUNT
COUNT=32 INDICATES PRESENCE OF SCINTI-LLATIONS
COMPARE S4 COUNT
<32
YES
COUNT =32
FDD MOTOR ON
=34
YES
NO
RECALIBRATE FDD
=35
NO
SEEK
=36
NO
YES
S4 FOR 4 BLOCKS
RECORD
START
RECORD
DISABLE INT
RECORD CYLINDER
FDD MOTOR OFF
INCREMENT TRK NO
TEST TRK NO
28H
=40
NO
YES
JMP TO MONITOR
FLOPPY FULL
STORE TRK CNT
20B7H
UPDATE DISPLAY FIELD
COUNT 5 PULSES
COMPENS-ATE FOR HALT
INIT MEM POINTER
RET

Appendix-B

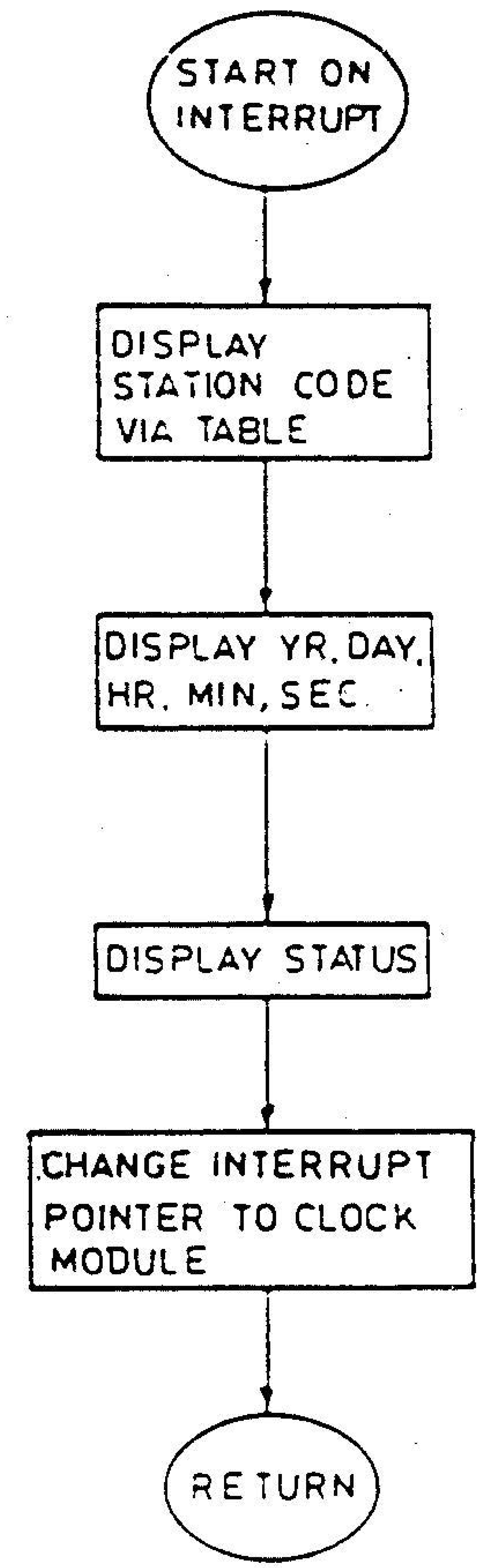
A/D CONVERTER

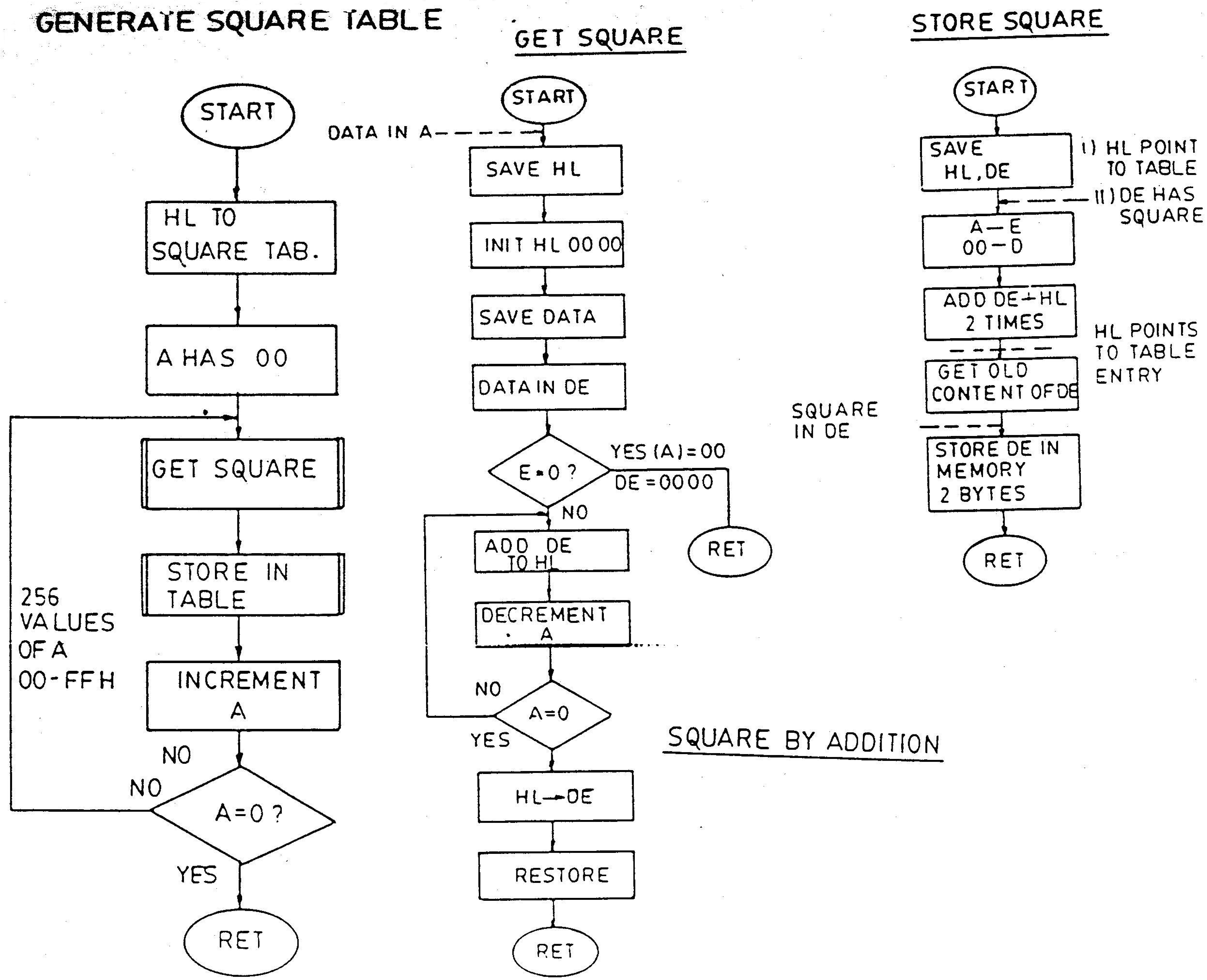


CLOCK MODULE



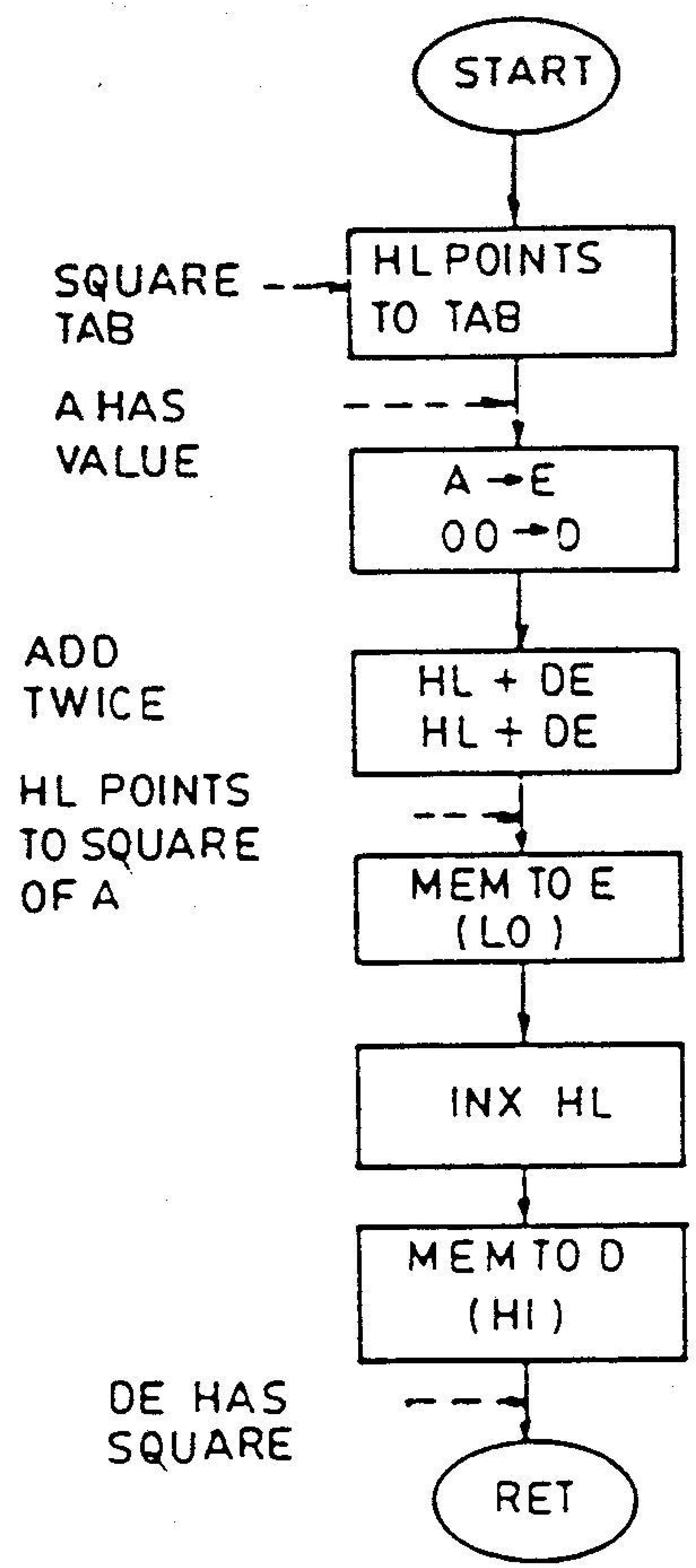
DISPLAY MODULE



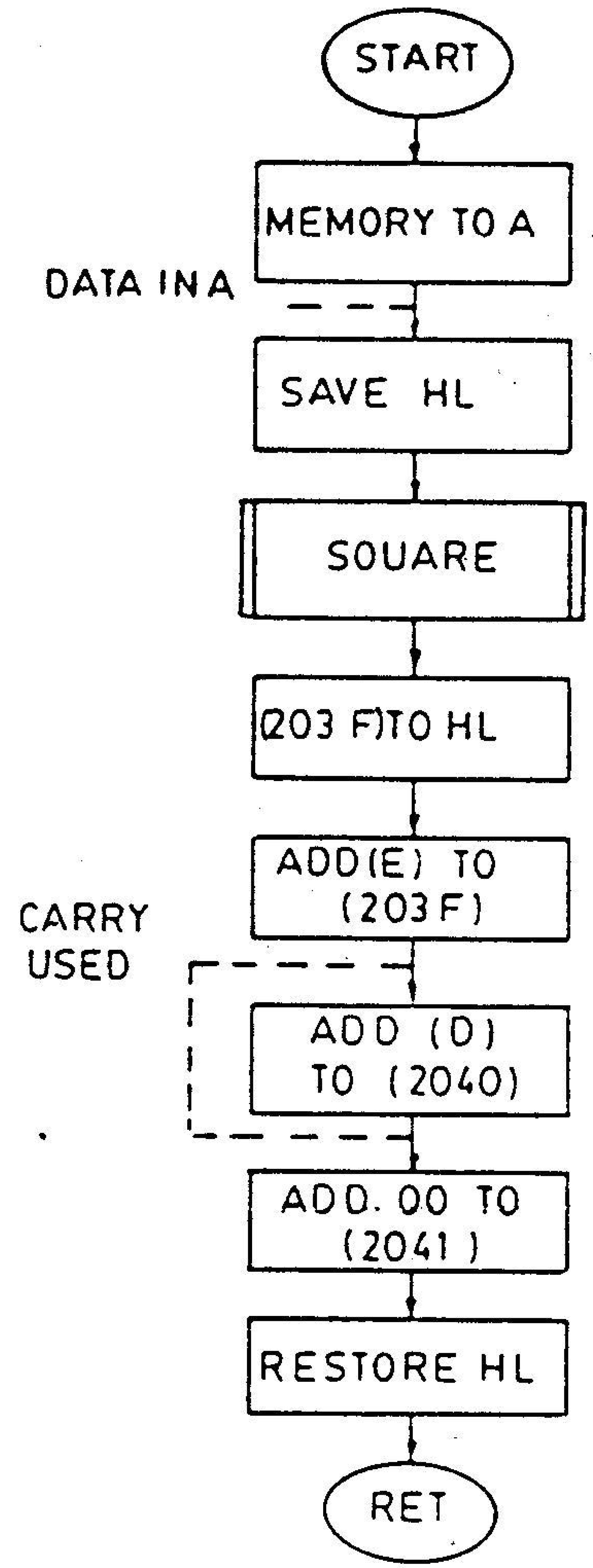

GENERATE SQUARE TABLE
START
HL TO SQUARE TAB.
A HAS 00
GET SQUARE
STORE IN TABLE
INCREMENT A
A=0 ?
NO
NO
YES
RET
256 VALUES OF A 00-FFH
GET SQUARE
DATA IN A
START
SAVE HL
INIT HL 0000
SAVE DATA
DATA IN DE
E=0 ?
YES (A)=00
DE=0000
RET
NO
ADD DE TO HL
DECREMENT A
A=0
NO
YES
HL—DE
RESTORE
RET
SQUARE BY ADDITION
STORE SQUARE
START
SAVE HL,DE
I) HL POINT TO TABLE
II) DE HAS SQUARE
A—E 00—D
ADD DE+HL 2 TIMES
HL POINTS TO TABLE ENTRY
GET OLD CONTENT OF DE
SQUARE IN DE
STORE DE IN MEMORY 2 BYTES
RET

APPENDIX D

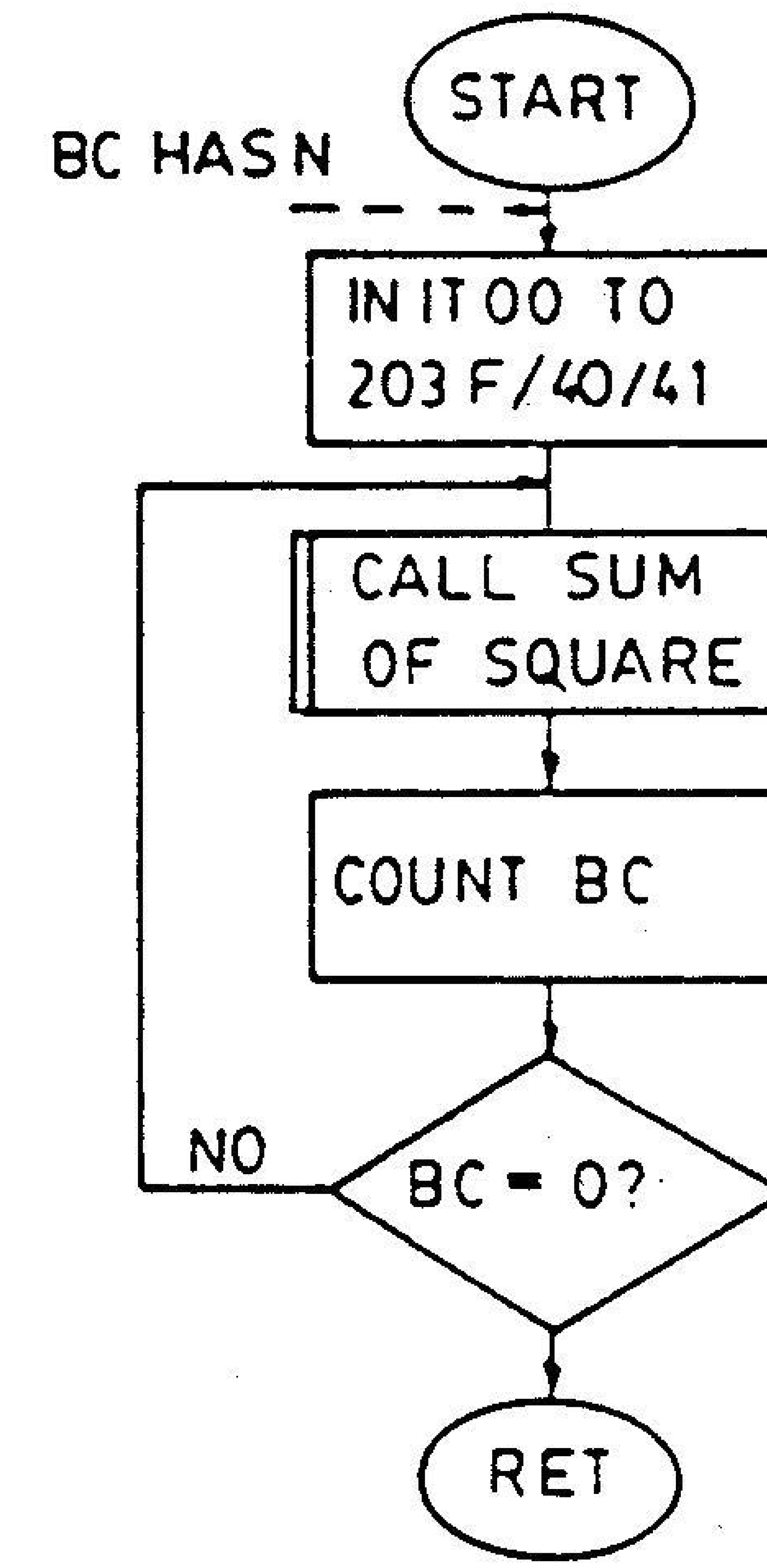
SQUARE



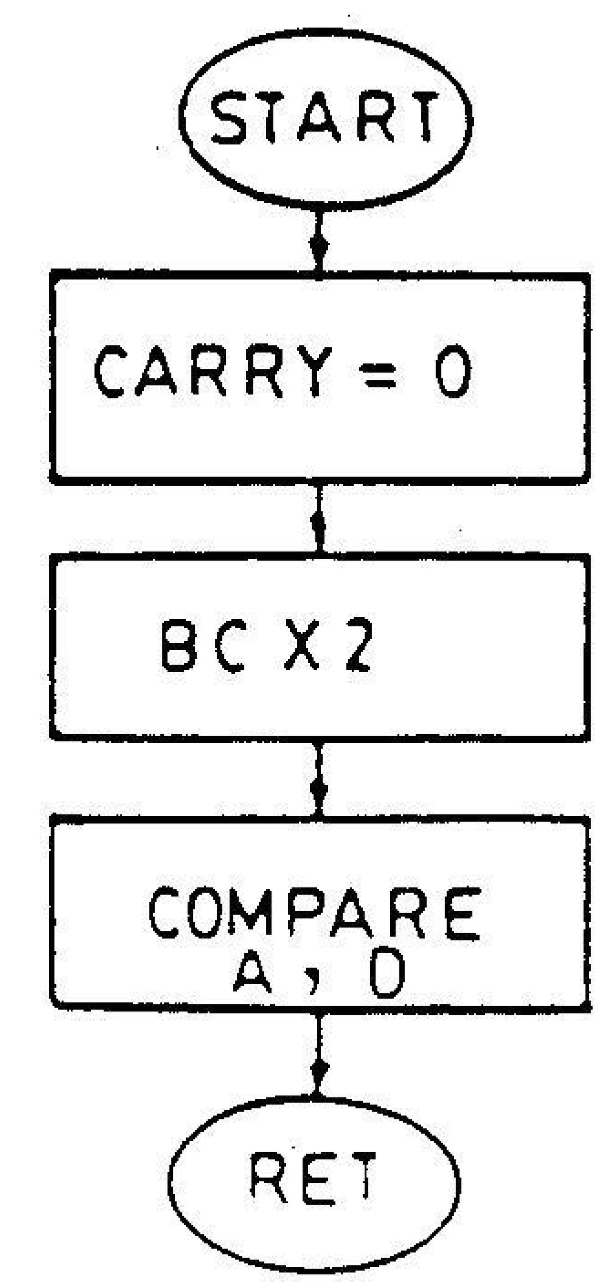
SUM OF SQUARE



SUM OF N SQUARE

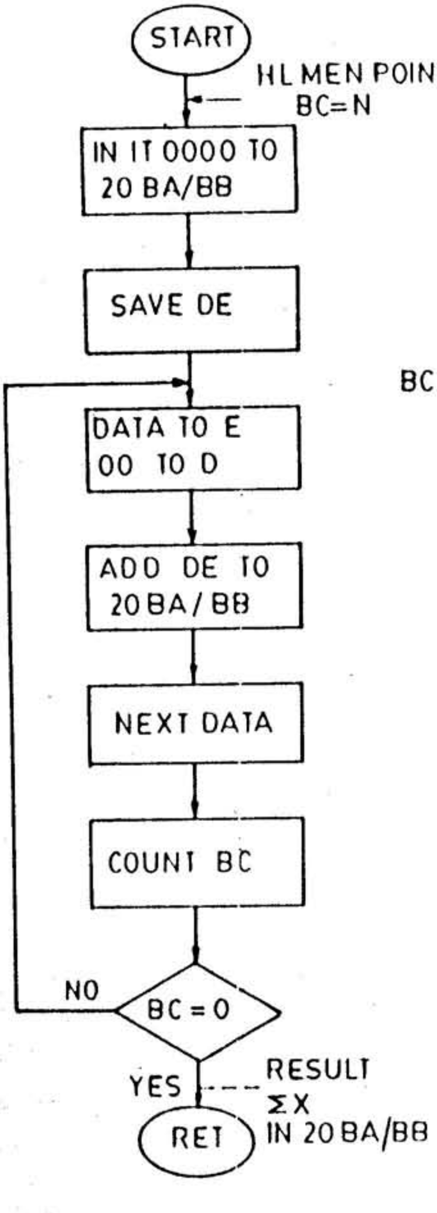


BC X 2

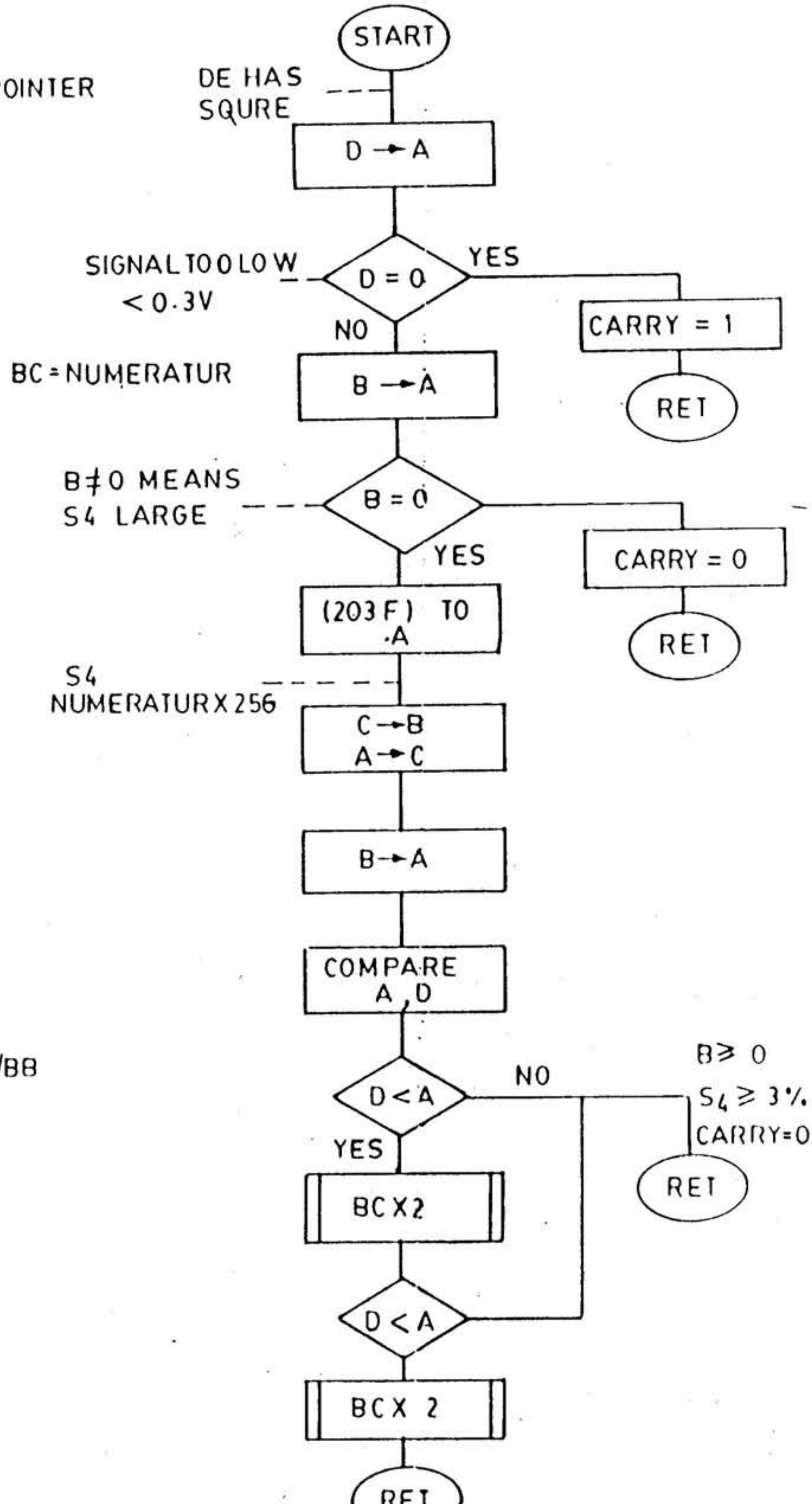


APPENDIX E

SUM OF N POINTS



DECISION



APPENDIX F

STEP BY STEP OPERATION

The instrument behaves like an 8085A microprocessor design kit on power on/reset and diesplays 8085 on the two 8 digit sections of the display .The recorder program is located at 6000h in EPROM and has to be copied at 2100h for operation.The step by step operation is as follows:

1. 63A0 GO : run program at 63A0 to copy

2. Insert blank floppy diskette in the floppy disk drive, close the FDD door andgive format command:

   2440 GO : run format command

   Now the LED in the FDD would light up and motion of head carriage may be heard, tracks are formatted one by one, and when all of the disk is formatted the LED goes off.

3. Enter the time of the day, day of the year and year in the following sequence:

Location	Contents
20A2h	Seconds
20A3h	Minutes
20A4h	Hours
20A5h	Tens and Units of the day
20A6	Leading Zero and Hundreds of the year
20A7	Year(19XX), 1st two digits

Subsequently , Three character station code is entered in locations 20A8, 20A9 and 20AA , for example code AHD for Ahmedabad would be enterd as:

2ØA8h	41 " A "
2ØA9h	48 " H "
2ØAAh	44 " D "

4. The recorder program can now be started by :

   22AØ GO : run program.

   The Displays start showing Station codse , Year , Day and Time the two LED's flicker to show the channel selection and thus proper operation of the instrument .

5. The FDD turns on after about 15 minutes to record a track and then it turns off .Tracks already full are indicated on the 6 LED's as 6 bit binary value.This process would continue till all the 4Ø tacks are over, when the program halts.

6. Data Read:

   Insert recorded floppy in the FDD:

   63AØ GO : to copy the program

   24ØØ GO : to read track Ø

   The display shows header information a new Go command without address would read next track.Reset button may be used to terminate read command.

APPENDIX G

TEMPORARY STORAGE, MEMORY MAP, I/O, TIMERS

1. Display Parameter Storage:

Address	Parameter
20Ø1	1/1Ø Seconds
20Ø2	Seconds
20Ø3	Minutes
20Ø4	Hour of the Day
20Ø5	Tens and Units of Day of the Year
20Ø6	Leading Zero and Hundreds of the Day of the Year
20Ø7	Year (19XX)
20Ø8	Station Code I
20Ø9	Station Code II
20ØA	Station Code III

2. Other Storage:

Stack	20AØ
Sum of Squares	203F/2040/2041 , 3 bytes
S4 count	20B2
Status Byte	20B3
S4 Block Count	20B4
Track Count (FDD)	20B7

3. Storage Related to Floppy Disk Controller:

Command Code	2040 to 2048, 9 bytes
Head Selected(HDS)	20B6
Track Count	20B7
Head No	20B8

Sector Number	20B9

4. Memory Map

0000 to 0FFF	Monitor(EPROM 2732)
1000 to 1FFF	Wrap up
2000 to 5FFF	RAM
6000 to 6FFF	Recorder Program (EPROM 2732)
7000 to 7FFF	Wrap up

5. Status Byte at 20B3, Bit Assignments

Bits 0 and 1;	Multiplexer
Bits 2 to 7;	Tracks Full in Binary 6 bits

The same is displayed on the status LED's

6. I/O Map and Port /Timer Assignments

Address	Device
00	8251 USART
08	8253 TIMER
Counter 0:	/05
Counter 1:	/30720 for 10 PPS, 3072 for 100 PPS during test
Counter 2:	/8 38>4 Khz Clock 8251
10h	8255
Port B;	D/A Converter (output)
Port A ,PA0:	Comparator Output test (input)
Port C;	
PC0	Channel Select bit 0
PC1	Channel Select bit 1
PC2	Cassette T/R Motor ON/OFF
PC3	Mux Enable

PC4	Ø for Write
PC5, PC6, PC7	Not Used

7. Floppy Disk Controller Ports

Address	Device
FØ	FDC (765)
F2	FDC Register
Bit Ø:	FDD Ø Motor On (MOT Ø)
Bit 1:	FDD Ø Enable (SEL Ø)
Bit 2:	Terminal Count (TC)
Bit 3:	Reset 765
Bit 4:	FDD 1 Motor ON (MOT 1)
Bit 5:	FDD 1 Enable (SEL 1)

8. EPROM Programmer Ports:

Address	Device	Function
F4	74LS245 /8282 latch	Data Port (Read/Write)
F5	8282	EPROM Address Low AØ-A7
F6	8282	EPROM Address High A8-A16
F7	74174	Control Port, bit 5 enables output of data latch at F4 others free

APPENDIX H

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	Q.
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	r'
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	45	N	n	5E	32	‾	‾
2F	21	/	r'	3F	17	F	F	4F	C5	0	o	5F	80	'	'

1. ASCII CODE
2. DISPLAY CODE
3. ASCII CHARACTER
4. DISPLAYED CHARACTER

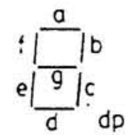
NOTE : INPUT CODES 3A TO 3F ARE USED FOR HEX [NON ASCII]

DISPLAY DATA FORMAT

8279 DISPLAY RAM OUTPUTS	A3	A2	A1	A0	B3	B2	B1	B0
DATA BITS	D7	D6	D5	D4	D3	D2	D1	D0
SEGMENTS	d	c	b	a	dP	g	f	e

'1' – LED is OFF

'0' – LED is ON



APPENDIX I

LIST OF COMPONENTS

A) LOGIC BOARD

INTEGRATED CIRCUITS

1. LM 311 COMPARATOR
2. CD 4Ø52 DUAL 4 IP MUX
3. LM 324 QUAD OP AMP
4. 8255 PROGRAMMABLE PORTS
5. 749Ø DECADE COUNTER
6. 74Ø4 HEX INVERTER
7. 8Ø85A MICROPROCESSOR
8. 74LS138 3 TO 8 DECODER
9. 74ØØ QUAD 2 IP NAND
1Ø. LM324 QUAD OPAMP
11. AD7533 D/A CONVERTER
12. 8253 PROGRAMMABLE CNTR
13. 74LS245
14. 74LS373 OCTAL LATCH
15. 74LS245
16. 74LS139 DUAL 2 TO 4 DECODER
17. 1488 QUAD RS232 DRIVER
18. 1489 QUAD RS232 LINE RCVR
19. LM311 COMPARATOR
2Ø. 8251 USART
21. 2732A 4KX8 EPROM (MONITOR)
22. 6264 8KX8 STATIC RAM
23. 6264 .
24. 2732A 4KX8 EPROM (PROGRAM)
25. 74LS245 OCTAL BUS BUFFER
26. CD 4Ø3Ø QUAD 2 IP XOR
27. CD 4Ø11 QUAD 2 IP NAND
28. CD 4Ø27 DUAL JK FF
29. CD 4Ø4Ø 12 STAGE COUNTER

RESISTORS		*CAPACITORS*	
[illegible]1,R2,R3,R4,R17	2K2	C1,C2,C3,C4	Ø.ØØ1UF
[illegible],R6,R7,R8,R14,R16,R24	3K3	C5,C6,C9,C1Ø,C14	Ø.1 UF
[illegible],R1Ø,R11,R12,R21,R22, [illegible],R25,R26,R27 & R28	1ØK	C7	Ø.ØØ47UF
[illegible]	51K	C8	Ø.Ø2 UF
[illegible]	1ØØE	C11	1ØØ PF
[illegible],R19,R2Ø	8K2	C12	2ØØ PF

R21	1ØØK	C13,C16	1Ø UF 1ØV
		C15	1UF 1Ø V

DIODES D1 TO D8 1N 4148 TRANSISTORS 2N2222

ZENER DIODE 5 V

B) FDC CARD

Resistors :- 22ØE ,5pcs.33ØE,4 pcs.1K 2pcs.1k5 2pcs.

Capacitors:-1ØØØpF 1pc.Ø.1 uF 2pcs.1ØuF 1ØV 1pc.

Integrated Circuits:-

IC1	74Ø6	IC16	742Ø
IC2	7414	IC17	8282
IC3	74Ø4	IC18	8282
IC4	74ØØ	IC19	8282
IC5	74153	IC2Ø	74LS245
IC6	74175	IC21	74LS155
IC7	74161	IC22	74LS32
IC8	7474	IC23	74LS138
IC9	7473	IC24	74LS245
IC1Ø	74Ø6	IC25	74Ø4
IC11	74174		
IC12	74174		
IC13	765 (FDC)		
IC14	7493		
IC15	74Ø4		

C) POWER SUPPLY

RESISTORS		CAPACITORS	
R1,R2,R4,R9,R1Ø	4K7	C1	47ØØ UF 25V
R3	47 K	C2, C6,C8 ,C11	Ø.1 UF
R5	5.6 K	C3	Ø.Ø15 UF

R6,R12	1 K	C4,C5	Ø.Ø1 UF
R7,R8	1ØØ E 1W	C7	47Ø UF 25V
R13	5Ø E 5W	C9	22ØØ UF 25V
R11	1ØØ E	C1Ø	33 UF 25V
R14	33Ø E	C12	1ØØØ UF 25 V
R15	22Ø E		
R16	12Ø E Ø.5 W		
R17	1K5		
R18	1K8		
R19	1 K POT		
DIODES		TRANSISTORS	
D1 TO D12	1N54Ø8	T1,T2,T3	TIP42A
D13 TO D22	BY 159	T4	2N2222
Z1	6.2 V Ø.5 W		
LED1	GREEN		
LED2	RED		

INTEGRATED CIRCUIT

TL494CN

TRANSFORMER 5Ø HZ

23Ø V PRIMARY, 9-Ø-9 SECONDARY 2 AMP

SMPS TRANSFORMER

EE CORE 7X15X3Ø MM AIR GAP 1/32 IN

PRIMARY 2Ø TURNS 18 SWG IN PARALLEL

SECONDARY 42 TURNS (INNER LAYER)

APPENDIX J

PROGRAM SEGMENTS AND SUBROUTINES RELATED TO 4 CHANNEL RECORDER

S.No.	Address	Function
1	21ØØ	Clock
2	213Ø	Display
3	216Ø	A/D Converter
4	218F	Buffer Full
5	21A6	Channel Change over
6	21B8	Multiplexer Control
7	21C8	A/D Conversion,SAR in software loop
8	21FØ	D/A Converter test loop,Ramp output
9	22ØØ	Record Data , one cylinder 9 Kbytes
1Ø	223Ø	Count Five , to compensate for time loss
11	224Ø	Initialize and sense S4 occurrence count
12	225Ø	Coppy Header
13	227Ø	Interrupt Service Routine
14	2283	Sum of squares and sum of series
15	22AØ	-:MAIN PROGRAM ENTRY POINT:-
16	22F7	Go Ahead
16	23ØØ	Display BCD packed
17	231C	Split Nibbles
18	232D	Display Hex via table at Ø384H/23dØ
19	2337	Display Alphanumeric data
2Ø	235Ø	Initialize
21	2379	Record Mode,1Ø PPS to RST 7.5
22	238Ø	-:Main Initialization Call:-
23	23AØ	Write Block or cylinder

24	23B0	Read Block or Cylinder
25	13C0-13FF	Display Code Table
26	2400	Read one cylinder 9K bytes and display
27	2435	Copy ,HL source,DE destination ,BC count
28	2440	Format Disk ,9 Sectors per track ,40
29		tracks,double sided;Direct call
30	2450	Copy data ,HL to DE, count in BC
31	245B	Send File (Call)
32	2480	Initialize FDD for Read/Write
33	2490	Direct call,Write one cylinder
34	24A0	Self Copy , Runs at 63A0
35	24B6	Test 40 Tracks Over
36	24C0	Data Transfer to PC'via Serial Port
37	24CD	Reset Carry
38	24DF	Format Loop
39	24F7	Delay Loop
40	2500	Average , Sum of series in 20BA/20BBH
41	2524	Square, Data in A result in DE
42	2530	Calculate, S4 numerator
43	2541	Sum of squares
44	2550	Initialise sum of squares
45	2560	Sum of squares one value
46	2580	Store sum of squares and Sum in block 1
47	259C	Process one block
48	25B2	Block result
49	25C0	Process 32 Blocks
50	25E2	S4 count , BC numerator, DE denominator
51	2600	Test Square Table and other subroutines

52	268Ø	Square by addition method, data in A result in DE
53	26AØ	Store Square
54	26BØ	Generatre Square Table
55	26CØ	Fill memory by Triangular data
56	26EC	Fill memory

APPENDIX K

System Specifications

1.	Number of channels		1 to 4
2.	Input Range		Ø to +5 v
3.	Sampling Rate	i) One Channel	1Ø /sec
		ii) Two Channels	5 /sec
		iii) Four Channels	2.5/sec
4.	Recording Medium	36Ø K DSDD Mini Floppy Diskette	
5.	Data Block Size	9216 Bytes (one cylinder)	
6.	Power Source	23Ø V 5Ø Hz AC, OR, 12 V Lead Acid Battery	
7.	Power consumption		2Ø W approx.
8.	Dimensions		3Ø5mm X38Ømm X76 mm
9.	Weight		5.Ø Kg

APPENDIX L

HEXDUMP OF FOUR CHANNEL RECORDER (FLOPPY BASED)

```
21ØØ 21 AØ 2Ø CD 25 21 FE 1Ø CØ CD 25 21 FE 6Ø CØ CD
211Ø 25 21 FE 6Ø CØ CD 25 21 FE 24 CØ CD 25 21 E6 FF
212Ø CØ CD 25 21 C9 36 ØØ 23 34 7E 37 3F 27 77 C9 FF
213Ø E5 C5 F5 21 A8 2Ø Ø6 Ø3 3E 9Ø D3 81 CD 37 23 3E
214Ø 93 D3 81 21 A7 2Ø Ø6 Ø6 CD ØØ 23 3E 9F D3 81 3A
215Ø B3 2Ø 2F D3 8Ø F1 C1 E1 C9 DB 12 E6 F7 D3 12 C9
216Ø E5 CD 59 21 2A AC 2Ø CD C8 21 23 22 AC 2Ø CD A6
217Ø 21 CD B8 21 CD 9F 21 7D B7 C2 8Ø 21 7C 32 B4 2Ø
218Ø E1 C9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
219Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF DB
21AØ 12 F6 Ø8 D3 12 C9 C5 3A B3 2Ø 4F 3C E6 Ø3 47 79
21BØ E6 FC BØ 32 B3 2Ø C1 C9 C5 DB 12 E6 FC 4F 3A B3
21CØ 2Ø E6 Ø3 B1 D3 12 C1 C9 C5 Ø1 8Ø 8Ø 79 D3 11 D5
21DØ D1 DB 1Ø 1F D2 DB 21 78 2F A1 4F 3E FE AØ CA E9
21EØ 21 78 ØF 47 B1 4F C3 CD 21 71 C1 C9 FF FF FF FF
21FØ 31 AØ 2Ø CD 8Ø 23 3E ØØ D3 11 3C C3 F8 21 FF FF
22ØØ F3 CD AØ 23 ØØ CD 83 Ø1 3A B7 2Ø 3C FE 28 CA DA
221Ø 22 32 B7 2Ø 17 17 C5 E6 FC 47 21 B3 2Ø 7E E6 Ø2
222Ø BØ 77 C1 CD 3Ø 22 3E ØØ 32 B4 2Ø C9 FF FF FF FF
223Ø E5 C5 ØE Ø5 C5 CD ØØ 21 C1 ØD C2 34 22 C1 E1 C9
224Ø 3E ØØ 32 B2 2Ø CD CØ 25 3A B2 2Ø FE 2Ø C9 FF FF
225Ø D5 C5 11 AB 2Ø Ø1 Ø6 ØA 1A 77 23 1B Ø5 C2 58 22
226Ø 36 ØØ 23 ØD C2 6Ø 22 C1 D1 C9 FF FF FF FF FF FF
227Ø C5 D5 E5 F5 CD ØØ 21 CD 6Ø 21 CD 3Ø 21 F1 E1 D1
228Ø C1 FB C9 C5 Ø1 ØØ Ø1 CD 9C 25 C1 Ø5 C2 83 22 C9
229Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
22AØ 31 AØ 2Ø 21 ØØ ØØ 22 B3 2Ø CD 8Ø 23 3E ØB 3Ø 3E
22BØ Ø8 D3 12 21 ØØ 3Ø CD 5Ø 22 22 AC 2Ø FB 3E ØØ 32
22CØ B4 2Ø 3A B4 2Ø FE 51 DA C2 22 C2 D9 22 CD 4Ø 22
22DØ DA B3 22 CD 7Ø Ø1 C3 BD 22 FE 52 C2 E4 22 CD 48
22EØ Ø1 C3 BD 22 FE 53 C2 EF 22 CD 6Ø Ø1 C3 BD 22 FE
22FØ 54 CC F7 22 C3 B3 22 Ø6 Ø4 CD 83 22 CD ØØ 22 C9
23ØØ ØE Ø2 CD 1C 23 E5 21 F9 2Ø 7E CD 2D 23 D3 8Ø ØØ
231Ø 23 ØD C2 Ø9 23 E1 2B Ø5 C2 ØØ 23 C9 7E ØF ØF ØF
232Ø ØF E6 ØF 32 F9 2Ø 7E E6 ØF 32 FA 2Ø C9 E5 21 84
233Ø Ø3 85 6F 7E E1 2F C9 3E 9Ø D3 81 D5 5E 16 ØØ E5
234Ø 21 AØ 23 19 7E 2F E1 D3 8Ø 23 Ø5 C2 3C 23 D1 C9
235Ø 3E 3E D3 ØB 3E Ø5 D3 Ø8 3E ØØ D3 Ø8 3E 7C D3 ØB
236Ø 3E ØØ D3 Ø9 3E 78 D3 Ø9 3E BE D3 ØB 3E Ø8 D3 ØA
237Ø 3E ØØ D3 ØA 3E 9Ø D3 13 C9 DB 12 E6 EF D3 12 C9
238Ø CD 5Ø 23 CD 79 23 3E Ø8 D3 81 CD 2A ØF CD ØØ ØF
239Ø 21 CE 2Ø 36 C3 23 36 7Ø 23 36 22 CD BØ 26 C9 FF
23AØ 21 ØØ 3Ø Ø1 ØØ 24 3E C5 CD CC ØD C9 FF FF FF FF
23BØ 21 ØØ 3Ø Ø1 ØØ 24 3E C6 CD E1 ØD C9 FF FF FF FF
23CØ ØØ 28 22 ØØ ØØ ØØ ØØ 2Ø 93 FØ ØØ 64 Ø8 Ø4 Ø8 21
23DØ F3 6Ø B5 F4 66 D6 D7 7Ø F7 76 77 C7 93 E5 B7 17
23EØ 35 77 C7 93 E5 97 17 D3 47 4Ø E1 67 83 45 45 C5
23FØ 37 CD Ø5 D6 87 E3 C1 C1 25 E6 B5 93 46 FØ 32 8Ø
```

```
24Ø0 31 AØ 2Ø CD 8Ø 24 CD BØ 23 3A B7 2Ø 3C 32 B7 2Ø
241Ø 21 ØØ 3Ø 11 AB 2Ø Ø1 ØA ØØ CD 35 24 CD 3Ø 21 CD
242Ø 6Ø Ø1 CF C3 Ø6 24 FF FF FF FF FF FF FF FF FF FF
243Ø FF FF FF FF FF 7E 12 23 1B ØB 78 B1 C2 35 24 C9
244Ø 31 AØ 2Ø CD 8Ø 24 CD EØ 24 CD 83 Ø1 CF FF FF FF
245Ø 7E 12 23 13 ØB 78 B1 C9 FF FF FF C5 1E 1Ø CD EB
246Ø Ø5 ØE 2Ø CD FA Ø7 7E CD C7 Ø6 C1 ØB 78 B1 C8 C5
247Ø 23 1D 7B FE Ø8 CA 61 24 7B A7 CA 5C 24 C3 66 24
248Ø CD 2A ØF CD ØØ ØF CD 7Ø Ø1 CD 48 Ø1 C9 FF FF FF
249Ø 31 AØ 2Ø CD 8Ø 24 CD AØ 23 CD 83 Ø1 CF FF FF FF
24AØ 31 AØ 2Ø 21 ØØ 6Ø 11 ØØ 21 Ø1 ØØ Ø8 CD 5Ø 63 C2
24BØ AC 63 CF CA BE 24 FE 28 CA CD 24 CD 6Ø Ø1 37 C9
24CØ 31 AØ 2Ø 21 ØØ 3Ø Ø1 ØØ 24 CD 5B 24 CF 37 3F C9
24DØ FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 34
24EØ CD A1 ØE 21 B8 2Ø 7E E6 Ø1 CA DF 24 35 2B 34 7E
24FØ CD B6 24 DA EØ 24 C9 D5 11 FF FF CD F1 Ø5 D1 C9
25ØØ E5 21 ØØ ØØ 22 BA 2Ø E1 D5 5E 16 ØØ E5 2A BA 2Ø
251Ø 19 22 BA 2Ø E1 23 ØB 78 B1 C2 Ø9 25 D1 C9 FF FF
252Ø FF FF FF FF 21 ØØ 28 5F 16 ØØ 19 19 5E 23 56 C9
253Ø 3A BB 2Ø CD 24 25 2A 4Ø 2Ø E5 D5 CD 51 ØF D1 E1
254Ø C9 CD 5Ø 25 CD 6Ø 25 23 ØB 78 B1 C2 44 25 C9 FF
255Ø E5 21 3F 2Ø 36 ØØ 23 36 ØØ 23 36 ØØ E1 C9 FF FF
256Ø 7E E5 CD 24 25 21 3F 2Ø 7B 8E 77 23 7A 8E 77 23
257Ø 3E ØØ 8E 77 E1 C9 FF FF FF FF FF FF FF FF FF FF
258Ø D5 C5 11 3F 2Ø Ø6 Ø3 1A 77 23 13 Ø5 C2 87 25 3A
259Ø BA 2Ø 77 23 3A BB 2Ø 77 23 C1 D1 C9 E5 C5 CD ØØ
25AØ 25 C1 E1 CD 41 25 E5 2A BC 2Ø CD 8Ø 25 22 BC 2Ø
25BØ E1 C9 CD 9C 25 E5 CD 33 25 CD E2 25 E1 C9 FF FF
25CØ 21 1Ø 3Ø 22 BC 2Ø Ø6 2Ø C5 Ø1 ØØ Ø1 CD B2 25 C1
25DØ Ø5 21 ØØ 31 C5 Ø1 ØØ Ø1 CD B2 25 C1 Ø5 C2 D4 25
25EØ C9 FF 7A B7 C8 78 B7 C2 F4 25 79 BA D2 F4 25 37
25FØ 3F 17 BA D8 3A B2 2Ø 3C 32 B2 2Ø C9 FF FF FF FF
26ØØ 31 AØ 2Ø CD BØ 26 21 ØØ 3Ø Ø1 ØØ Ø1 E5 C5 CD 41
261Ø 25 C1 E1 CD ØØ 25 CD 3Ø 25 CF FF FF FF FF FF FF
262Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
263Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
264Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
265Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
266Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
267Ø FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
268Ø E5 21 ØØ ØØ F5 5F 16 ØØ ØØ B7 CA 93 26 19 3D C2
269Ø 8D 26 EB F1 E1 C9 FF FF FF FF FF FF FF FF FF FF
26AØ E5 D5 5F 16 ØØ 19 19 D1 73 23 72 E1 C9 FF FF FF
26BØ 21 ØØ 28 3E ØØ CD 8Ø 26 CD AØ 26 3C C2 B5 26 C9
26CØ 77 D5 7E 82 23 77 ØB 78 B1 CA E8 26 1D C2 C2 26
26DØ D1 D5 7E 92 23 77 ØB 78 B1 CA E8 26 1D C2 D2 26
26EØ D1 C3 C1 26 FF FF FF FF D1 C9 FF FF 31 AØ 2Ø 21
26FØ ØØ 3Ø Ø1 ØØ 2Ø 11 Ø4 Ø1 3E ØF CD CØ 26 CF FF FF
```

APPENDIX M

DISASSEMBLED SOFTWARE

" CLOCK SUBROUTINE "

```
2100    21A020    LXI H,20A0   ; POINT TO 1/10 SECONDS
2103    CD2521    CALL 2125    ; COUNT AND UPDATE
2106    FE10      CPI 10       ; 10 PULSES OVER ?
2108    C0        RNZ          ; RETURN IF NOT
2109    CD2521    CALL 2125    ; COUNT SECONDS
210C    FE60      CPI 60       ; 60 SECONDS ?
210E    C0        RNZ          ; RETURN IF NOT YET
210F    CD2521    CALL 2125    ; COUNT MINUTES
2112    FE60      CPI 60       ; 60 MINUTES ?
2114    C0        RNZ          ; RETURN IF NOT
2115    CD2521    CALL 2125    ; COUNT HOURS
2118    FE24      CPI 24       ; 24 HOURS ?
211A    C0        RNZ          ; RETURN TILL ONE DAY OVER
211B    CD2521    CALL 2125    ; COUNT DAYS
211E    E6FF      ANI FF       ; 99  DAYS ?
2120    C0        RNZ          ; RETURN  TILL 100 DAYS
2121    CD2521    CALL 2125    ; COUNT DAYS x100
2124    C9        RET          ; ALL OVER
```

LOAD ZERO AND ADVANCE MEM POINTER .USED BY CLOCK

```
2125    3600      MVI M,00     ; RESET MEM
2127    23        INX H        ; NEXT LOCATION
2128    34        INR M        ; INCREMENTD
2129    7E        MOV A,M      ; VALUE ADJUSTED FOR
212A    37        STC          ;
212B    3F        CMC          ; DECIMAL COUNTING
212C    27        DAA          ;
212D    77        MOV M,A      ; STORE BACK
212E    C9        RET          ;
```

" DISPLAY"

```
2130    E5        PUSH H       ; SAVE
2131    C5        PUSH B       ;
2132    F5        PUSH PSW     ;
2133    21A820    LXI H,20A8   ; POINT TO STATION CODE ASCII
2136    0603      MVI B,03     ; 3  CHAR TO BE DISPLAYED
2138    3E90      MVI A,90     ; CODE TO 8279
213A    D381      OUT 81       ; CONTROL REG.
213C    CD3723    CALL 2337    ; DISPLAY 2 , ALPHA
213F    3E93      MVI A,93     ; 4 TH DISPLAY UP FOR NUMERIC
2141    D381      OUT 81       ; CODE TO 8279
2143    21A720    LXI H,20A7   ; POINT TO YEAR DAYS TIME
2146    0606      MVI B,06     ; 6 BYTES
2148    CD0023    CALL 2300    ; CALL DISPLAY 1, NUMERIC
214B    3E9F      MVI A,9F     ; 16 TH DISPLAY 8 LED "S
214D    D381      OUT 81       ;
```

```
214F    3AB32Ø    LDA 2ØB3      ; TO DISPLAY STATUS BYTE
2152    2F        CMA           ; PREPARE FOR DISPLAY
2153    D38Ø      OUT 8Ø        ; BEFORE SENDING TO 8279
2155    F1        POP PSW       ; RESTORE
2156    C1        POP B         ; REGS.
2157    E1        POP H         ;
2158    C9        RET           ;
```

" ENABLE MUX"

```
2159    DB12      IN 12         ; PORT C
215B    E6F7      ANI F7        ; BIT 3 =Ø
215D    D312      OUT 12        ; TO ENABLE MUX
215F    C9        RET           ;
```

"A/D CONVERTER "

```
216Ø    E5        PUSH H        ; SAVE MEM POINTER
2161    CD5921    CALL 2159     ; CALL ENABLE MUX
2164    2AAC2Ø    LHLD 2ØAC     ; LOAD DATA POINTER
2167    CDC821    CALL 21C8     ; CALL CONVERT AND STORE
216A    23        INX H         ; ADVANCE  MEM POINTER
216B    22AC2Ø    SHLD 2ØAC     ; AND SAVE BACK
216E    CDA621    CALL 21A6     ; CALL STATUS UPDATE AND
2171    CDB821    CALL 21B8     ; MUX UPDATE
2174    CD9F21    CALL 219F     ; CALL DISABLE MUX
2177    7D        MOV A,L       ; MEM OINTER LO BYTE
2178    B7        ORA A         ; TESTED FOR ZERO
2179    C28Ø21    JNZ 218Ø      ; IF NO THEN RETURN
217C    7C        MOV A,H       ; ELSE HI BYTE
217D    32B42Ø    STA 2ØB4      ; STORED AT 2ØB4H
218Ø    E1        POP H         ; RESTORE
2181    C9        RET           ; AND RETURN
```

"DISABLE MULTIPLEXER"

```
219F    DB12      IN 12         ; PORT C BIT 3
21A1    F6Ø8      ORI Ø8        ; MADE 1
21A3    D312      OUT 12        ; TO DISABLE MUX
21A5    C9        RET           ;
```

"STATUS UPDATE"

```
21A6    C5        PUSH B        ; SAVE COUNT
21A7    3AB32Ø    LDA 2ØB3      ; GET STATUS BYTE
21AA    4F        MOV C,A       ; SAVE IN C REG.
21AB    3C        INR A         ; INCREMENT MUX COUNT
21AC    E6Ø3      ANI Ø3        ; TO SELECT NEXT CHANNEL
21AE    47        MOV B,A       ; SAVE 2 LO BITS IN B REG
21AF    79        MOV A,C       ; GET OLD STATUS  BACK
21BØ    E6FC      ANI FC        ; REJECT MUX BITS
21B2    BØ        ORA B         ; 'OR' WITH NEW BITS
21B3    32B32Ø    STA 2ØB3      ; STORE NEW STATUS
21B6    C1        POP B         ; RESTORE
21B7    C9        RET           ;
```

"MULTIPLEXER UPDATE"

```
21B8    C5       PUSH B      ; SAVE COUNT
21B9    DB12     IN 12       ; READ PORT C TO GET MUX STATUS
21BB    E6FC     ANI FC      ; KILL MUX BITS
21BD    4F       MOV C,A     ; SAVE THE REST
21BE    3AB32Ø   LDA 20B3    ; NEW STATUS
21C1    E6Ø3     ANI Ø3      ; LO 2 BITS ARE MUX BITS
21C3    B1       ORA C       ; 'OR' MUX BITS TO REST OLD
21C4    D312     OUT 12      ; AND OUTPUT
21C6    C1       POP B       ;
21C7    C9       RET         ;
```

"CONVERT AND STORE (SUCCESSIVE APPROX. METHOD)"

```
21C8    C5       PUSH B      ; SAVE COUNT
21C9    Ø18Ø8Ø   LXI B,8Ø8Ø  ; INIT BC
21CC    79       MOV A,C     ; GET DATA FROM C REG.
21CD    D311     OUT 11      ; OUT TO B PORT FOR D/A
21CF    D5       PUSH D      ; SAVE DE AND RESTORE
21DØ    D1       POP D       ; ,TIME WAS NEEDED  FOR D/A
21D1    D5       PUSH D      ; ,TIME WAS NEEDED  FOR D/A
21D2    D1       POP D       ; ,TIME WAS NEEDED  FOR D/A
21D3    DB1Ø     IN 1Ø       ; STABILIZATION,READ COMPARATOR
21D5    1F       RAR         ; OUTPUT ,BIT Ø OF PORT A
21D6    D2DD21   JNC 21DB    ; OUTPUT ZERO MEANS LESS THAN INPUT
21D9    78       MOV A,B     ; IF MORE THAN IT THEN,
21DA    2F       CMA         ; COMPLEMENT
21DB    A1       ANA C       ; CURRENT BIT SET TO ZERO
21DC    4F       MOV C,A     ;  SAVE IT
21DD    3EFE     MVI A,FE    ;
21DF    AØ       ANA B       ;
21EØ    CAEB21   JZ 21EB     ; IF ALL 8 BITS OVER THEN STORE
21E3    78       MOV A,B     ; AND RET,ELSE MOVE TO NEXT
21E4    ØF       RRC         ; MSB
21E5    47       MOV B,A     ;
21E6    B1       ORA C       ; DO SUCCESSIVE APPROX.
21E7    4F       MOV C,A     ; SAVE IN C REG.
21E8    C3CD21   JMP 21CD    ; REPEAT FOR ALL BITS
```

STORE AND RETURN

```
21EB    71       MOV M,C
21EC    C1       POP B
21ED    C9       RET
```

"D/A CONVERTER TEST"

```
21FØ    31AØ2Ø   LXI SP,2ØAØ ; INIT STACK
21F3    CD8Ø23   CALL 238Ø   ; AND PORTS AND TIMERS
21F6    3EØØ     MVI A,ØØ    ; START FROM ØØ
21F8    D311     OUT 11      ; OUTPUT TO D/A CONVERTER
21FA    3C       INR A       ; NEXT STEP
21FB    C3F821   JMP 21F8    ; AND REPEAT TO GET RAMP
```

"RECORD DATA"

9X1Ø24 BYTES ON A CYLINDER, BOTH SIDES OF A TRACK

```
2200    F3        DI           ; DISABLE INTERRUPTS, STOP DATA ACQ
2201    CDA023    CALL 23A0    ; CALL 'RECORD CYLINDER'
2204    00        NOP          ;
2205    CD8301    CALL 0183    ; STOP FDD MOTOR
2208    3AB720    LDA 20B7     ; TRACK COUNT
220B    3C        INR A        ; ADVANCED
220C    FE28      CPI 28       ; 40 TRACKS OVER ?
220E    CADA22    JZ 22DA      ; IF YES THEN HALT!!
2211    32B720    STA 20B7     ; ELSE STORE BACK TRACK COUNT
2214    17        RAL          ; SHIFT TRACK COUNT
2215    17        RAL          ; BY 2 PLACES TO GET 6 BITS IN
2216    C5        PUSH B       ; STATUS HI BITS AS TRACK COUNT
2217    E6FC      ANI FC       ; KILL LO 2 BITS , SAVE BC
2219    47        MOV B,A      ; SAVE STATUS 6 HI BITS IN B REG
221A    21B320    LXI H,20B3   ; POINT TO STATUS
221D    7E        MOV A,M      ; GET CURRENT STATUS IN A REG.
221E    E602      ANI 02       ; KILL HI 6 BITS TO UPDATE
2220    B0        ORA B        ; TRACK COUNT
2221    77        MOV M,A      ; STORE BACK STATUS
2222    C1        POP B        ;
2223    CD3022    CALL 2230    ; CALL 'COUNT FIVE'
2226    3E00      MVI A,00     ; INIT
2228    32B420    STA 20B4     ; HI  OF MEM POINTER
222B    C9        RET          ;
```

COUNT FIVE

```
2230    E5        PUSH H       ;
2231    C5        PUSH B       ; SAVE REGS.
2232    0E05      MVI C,05     ; COUNT IN C REG
2234    C5        PUSH B       ; SAVE
2235    CD0021    CALL 2100    ; COUNT 1/10 SEC
2238    C1        POP B        ;
2239    0D        DCR C        ;
223A    C23422    JNZ 2234     ; COUNT TILL FIVE TIMES TO ACCOUNT
223D    C1        POP B        ; FOR LOSS OF TIME IN RECORDING
223E    E1        POP H        ;
223F    C9        RET          ;
```

"INIT AND SENSE S4 COUNT"

```
2240    3E00      MVI A,00     ; INITIALIZE
2242    32B220    STA 20B2     ; S4 COUNT
2245    CDC025    CALL 25C0    ; CALL 'CALCULATE S4 AND ITS COUNT'
2248    3AB220    LDA 20B2     ; GET CURRENT S4 COUNT
224B    FE20      CPI 20       ; COUNT COMPARED TO 32
224D    C9        RET          ;
```

"COPY HEADER"

```
2250    D5        PUSH D      ;
2251    C5        PUSH B      ; SAVE REGS.
2252    11AB20    LXI D,20AB  ; POINT DE TO HEADER
2255    01060A    LXI B,0A06  ; 10 BYTES HEADER COUNT, 6 FOR 0
2258    1A        LDAX D      ; GET HEADER BYTE
2259    77        MOV M,A     ; STORE IN MEM POINTED BY HL
225A    23        INX H       ; NEXT UP LOC BY HL
225B    1B        DCX D       ; NEXT DOWN LOC BY DE
225C    05        DCR B       ; COUNT HEADER BYTES
225D    C25822    JNZ 2258    ; IF 10 BYTES OVER THEN
2260    3600      MVI M,00    ; LOAD  SIX  00  IN  COSECUTIVE
2262    23        INX H       ; LOCATIONS
2263    0D        DCR C       ;
2264    C26022    JNZ 2260    ;
2267    C1        POP B       ;
2268    D1        POP D       ; RESTORE
2269    C9        RET         ;
```

"INTERRUPT SERVICE ROUTINE"

```
2270    C5        PUSH B      ; SAVE ALL
2271    D5        PUSH D      ; REGISTERS
2272    E5        PUSH H      ;
2273    F5        PUSH PSW    ;
2274    CD0021    CALL 2100   ; CALL 'CLOCK'
2277    CD6021    CALL 2160   ; CALL 'A/D CONVERTER'
227A    CD3021    CALL 2130   ; CALL 'DISPLAY'
227D    F1        POP PSW     ; RESTORE
227E    E1        POP H       ; ALL
227F    D1        POP D       ; REGISTERS
2280    C1        POP B       ;
2281    FB        EI          ; ENABLE INTERRUPTS
2282    C9        RET         ;
```

"SUM OF SQUARES AND SUM OF SERIES FOR M BLOCKS"

```
2283    C5        PUSH B      ;
2284    010001    LXI B,0100  ; BLOCK SIZE 256 BYTES
2287    CD9C25    CALL 259C   ; CALL ' CALCULATE S4 AND STORE'
228A    C1        POP B       ;
228B    05        DCR B       ; COUNT 'M' AS IN B REG
228C    C28322    JNZ 2283    ; REPEAT M TIMES
228F    C9        RET         ;
```

"MAIN PROGRAM ENTRY POINT"

```
22A0    31A020    LXI SP,20A0 ; INIT STACK
22A3    210000    LXI H,0000  ; AND
22A6    22B320    SHLD 20B3   ; STATUS BYTE , BLOCK COUNT
22A9    CD8023    CALL 2380   ; INIT PORTS AND TIMERS
22AC    3E0B      MVI A,0B    ; ENABLE RST 7.5
22AE    30        SIM         ;
22AF    3E08      MVI A,08    ; SET UP RECORD MODE TO CONNECT
22B1    D312      OUT 12      ; 10 PPS PULSES TO RST 7.5
22B3    210030    LXI H,3000  ; INIT MEMORY POINTER
22B6    CD5022    CALL 2250   ; CALL 'COPY HEADER'
22B9    22AC20    SHLD 20AC   ; STORE BACK MEM POINTER
22BC    FB        EI          ; ENABLE INTERRUPT
22BD    3E00      MVI A,00    ; INIT
22BF    32B420    STA 20B4    ; BLOCK COUNT
22C2    3AB420    LDA 20B4    ; GET BLOCK COUNT
22C5    FE51      CPI 51      ; 32 BLOCKS STORED IN RAM ?
22C7    DAC222    JC 22C2     ; IF LESS THEN WAIT TILL THEN
22CA    C2D922    JNZ 22D9    ; IF COUNT MORE THAN 51,JUMPOVER
22CD    CD4022    CALL 2240   ; CALL 'INIT AND SENSE S4 COUNT'
22D0    DAB322    JC 22B3     ; S4 COUNT LESS THAN 32 THEN REPEAT
22D3    CD7001    CALL 0170   ; ACQUISITION  FROM START,ELSE CALL
22D6    C3BD22    JMP 22BD    ; 'FDD MOTOR ON',AND CONTINUE
```

BLOCK COUNT MORE THAN 51

```
22D9    FE52      CPI 52      ; TEST FOR BLOCK COUNT 52
22DB    C2E422    JNZ 22E4    ; IF MORE THEN JUMPOVER ELSE
22DE    CD4801    CALL 0148   ; CALL 'RECALIBRATE', GO TO TR 00
22E1    C3BD22    JMP 22BD    ; CONTINUE ACQUISITION
```

BLOCK COUNT MORE THAN 52

```
22E4    FE53      CPI 53      ; COUNT =53 ?
22E6    C2EF22    JNZ 22EF    ; IF LARGER THEN JUMPOVER
22E9    CD6001    CALL 0160   ; ELSE CALL 'SEEK' POSITION HEAD
22EC    C3BD22    JMP 22BD    ; THEN CONTINUE
```

BLOCK COUNT MORE THAN 53

```
22EF    FE54      CPI 54      ; COUNT =54 ?
22F1    CCF722    CZ 22F7     ; IF YES THEN CALL 'GO AHEAD'
22F4    C3B322    JMP 22B3    ; THEN START  AFRESH
```

9 K DATA OVER, GO AHEAD

```
22F7    0604      MVI B,04    ; 4 BLOCK DATA ACQUIRED AFTER
22F9    CD8322    CALL 2283   ; CALL 'SUM OF SQUARES..BLOCKS'
22FC    CD0022    CALL 2200   ; CALL 'RECORD DATA'
22FF    C9        RET         ;
```

" DISPLAY PACKED BCD (2 DECIMAL IN ONE BYTE)"

```
2300    ØEØ2     MVI C,Ø2     ; NIBBLE COUNT
2302    CD1C23   CALL 231C    ; CALL SPLIT
2305    E5       PUSH H       ;
2306    21F92Ø   LXI H,2ØF9   ; GET DISPLAY CODE
2309    7E       MOV A,M      ;
230A    CD2D23   CALL 232D    ; DISPLAY 2 DIGITS
230D    D38Ø     OUT 8Ø       ; VIA 8279
230F    ØØ       NOP          ;
2310    23       INX H        ; NEXT DATA
2311    ØD       DCR C        ;
2312    C2Ø923   JNZ 2309     ; DISPLAY ALL
2315    E1       POP H        ;
2316    2B       DCX H        ;
2317    Ø5       DCR B        ;
2318    C2ØØ23   JNZ 2300     ;
231B    C9       RET          ;
```

" SPLIT NIBBLES"

```
231C    7E       MOV A,M      ;
231D    ØF       RRC          ;
231E    ØF       RRC          ; HI NIBBLE
231F    ØF       RRC          ;
2320    ØF       RRC          ;
2321    E6ØF     ANI ØF       ; SELECTED
2323    32F92Ø   STA 2ØF9     ; SAVE IT
2326    7E       MOV A,M      ; GET
2327    E6ØF     ANI ØF       ; LO NIBBLE
2329    32FA2Ø   STA 2ØFA     ; STORE NEXT
232C    C9       RET          ;
```

" DISPLAY HEX DATA VIA TABLE AT Ø384 (ALSO AT 23DØ)"

```
232D    E5       PUSH H       ;
232E    2184Ø3   LXI H,Ø384   ; SYMBOL TABLE
2331    85       ADD L        ; POINT TO ENTRY CORRESPONDING
2332    6F       MOV L,A      ; TO DATA IN A
2333    7E       MOV A,M      ; GET CODE IN A
2334    E1       POP H        ;
2335    2F       CMA          ; COMPLEMENT TO DISPLAY
2336    C9       RET          ;
```

" DISPLAY ALPHANUMERIC DATA"

```
2337    3E9Ø     MVI A,9Ø     ; CODE TO 8279
2339    D381     OUT 81       ;
233B    D5       PUSH D       ;
233C    5E       MOV E,M      ; DATA IN E REG.
233D    16ØØ     MVI D,ØØ     ; INIT D
233F    E5       PUSH H       ; SAVE HL
234Ø    21AØ23   LXI H,23AØ   ; HL POINTS TO TABLE
2343    19       DAD D        ; POINT TO CODE
```

```
2344     7E          MOV A,M       ; GET TABLE ENTRY
2345     2F          CMA           ; COMPLEMENT TO DISPLAY
2346     E1          POP H         ;
2347     D380        OUT 80        ; VIA 8279
2349     23          INX H         ; NEXT LOCATION
234A     05          DCR B         ; COUNT CHAR
234B     C23C23      JNZ 233C      ; DISPLAY ALL CHAR
234E     D1          POP D         ;
234F     C9          RET           ;
```

"INITIALIZATION OF PORTS AND TIMERS"

```
2350     3E3E        MVI A,3E      ; CNTR 0 IN MODE 3 , LSB FIRST
2352     D30B        OUT 0B        ;
2354     3E05        MVI A,05      ; DIVIDE BY 5, OUTPUT 307.2 KHZ
2356     D308        OUT 08        ;
2358     3E00        MVI A,00      ; HI BYTE 00
235A     D308        OUT 08        ;
235C     3E7C        MVI A,7C      ; CNTR 1 MODE 2 , LSB FIRST
235E     D30B        OUT 0B        ;
2360     3E00        MVI A,00      ; LSB 00
2362     D309        OUT 09        ;
2364     3E78        MVI A,78      ; MSB 120 , DIVIDE BY 7800H
2366     D309        OUT 09        ; OR 30720 DECIMAL OUT 10 PPS
2368     3EBE        MVI A,BE      ; CNTR 2 MODE 3 LSB FIRST
236A     D30B        OUT 0B        ;
236C     3E08        MVI A,08      ; DIVIDE BY 8
236E     D30A        OUT 0A        ;
2370     3E00        MVI A,00      ; HI BYTE 00
2372     D30A        OUT 0A        ; OUTPUT 38.4 KHZ TO 8251 BAUD CLK
2374     3E90        MVI A,90      ; 8255 CODE . A INPUT ,B & C OUTPUT
2376     D313        OUT 13        ;
2378     C9          RET           ;
```

" RECORD MODE 10 PPS TO RST 7.5 ENABLED"

```
2379     DB12        IN 12         ; BIT 4 OF
237B     E6EF        ANI EF        ; PORT C MADE ZERO
237D     D312        OUT 12        ;
237F     C9          RET           ;
```

" INITIALIZATION SUBROUTINES CALLED"

```
2380     CD5023      CALL 2350     ; CALL 'INIT PORTS AND TIMERS'
2383     CD7923      CALL 2379     ; CALL RECORD MODE
2386     3E08        MVI A,08      ; CODE FOR 8279
2388     D381        OUT 81        ; 16 DISPLAYS , SCAN KEYBOARD
238A     CD2A0F      CALL 0F2A     ; RESET TO FDC
238D     CD000F      CALL 0F00     ; SPECIFY TO FDC
2390     21CE20      LXI H,20CE    ; INIT RST 7.5 TO POINT
2393     36C3        MVI M,C3      ; TO 2270H
2395     23          INX H         ;
2396     3670        MVI M,70      ;
2398     23          INX H         ;
2399     3622        MVI M,22      ;
```

```
239B     CDB026    CALL 26B0   ; CALL ' GENERATE TABLE OF SQUARES'
239E     C9        RET         ;
```

"WRITE BLOCK OR CYLINDER"

```
23A0     210030    LXI H,3000  ; DATA POINTER
23A3     010024    LXI B,2400  ; BYTE COUNTER DECIDES SIZE
23A6     3EC5      MVI A,C5    ; MULTITRACK WRITE, MFM
23A8     CDCC0D    CALL 0DCC   ; EXECUTE CODE IN MONITOR
23AB     C9        RET         ;
```

"READ BLOCK OR CYLINDER"

```
23B0     210030    LXI H,3000  ; MEMORY POINTER
23B3     010024    LXI B,2400  ; SIZE
23B6     3EC6      MVI A,C6    ; MULTI TRACK, MFM  READ
23B8     CDE10D    CALL 0DE1   ; EXECUTE IN MONITOR
23BB     C9        RET         ;
```

"DISPLAY TABLE"

```
23C0 00 28 22 00 00 00 00 20 93 F0 00 64 08 04 08 21
23D0 F3 60 B5 F4 66 D6 D7 70 F7 76 77 C7 93 E5 B7 17
23E0 35 77 C7 93 E5 97 17 D3 47 40 E1 67 83 45 45 C5
23F0 37 CD 05 D6 87 E3 C1 C1 25 E6 B5 93 46 F0 32 80
```

" READ ONE TRACK AND WAIT FOR NEXT"

```
2400     31A020    LXI SP,20A0 ; STACK INITIALIZED
2403     CD8024    CALL 2480   ; CALL 'INIT FDD FOR READ/WRITE'
2406     CDB023    CALL 23B0   ; CALL 'READ BLOCK'
2409     3AB720    LDA 20B7    ; TRACK NUMBER
240C     3C        INR A       ; INCREMENTED
240D     32B720    STA 20B7    ; AND STORED
2410     210030    LXI H,3000  ; DATA POINTER
2413     11AB20    LXI D,20AB  ; DISPLAY AREA
2416     010A00    LXI B,000A  ; DISPLAY DATA COUNT
2419     CD3524    CALL 2435   ; COPY HEADER
241C     CD3021    CALL 2130   ; DISPLAY
241F     CD6001    CALL 0160   ; CALL 'NEXT TRACK'
2422     CF        RST 1       ; WAIT FOR

2423     C30624    JMP 2406    ; NEXT TRACK  BY GO COMMAND
```

"COPY HEADER"

```
2435     7E        MOV A,M     ; HEADER FROM DATA
2436     12        STAX D      ; COPIED TO DISPLAY AREA AS IN DE
2437     23        INX H       ; NEXT HIGHER DATA
2438     1B        DCX D       ; LOWER DISPLAY LOC
2439     0B        DCX B       ; COUNT
243A     78        MOV A,B     ;
243B     B1        ORA C       ;
243C     C23524    JNZ 2435    ; COPY ALL
243F     C9        RET         ;
```

"FORMAT DISK, DSDD 36Ø KBYTE, 9 SECTORS 512 BYTE,4Ø TR

```
2440     31A020     LXI SP,20A0 ; STACK INIT
2443     CD8024     CALL 2480   ; CALL 'INIT FDD FOR READ/WRITE'
2446     CDE024     CALL 24E0   ; CALL ' FORMAT'
2449     CD8301     CALL 0183   ; CALL 'FDD MOTOR OFF'
244C     CF         RST 1       ; WAIT NEXT COMMAND FROM MONITOR
```

"COPY DATA HL TO DE COUNT IN BC"

```
2450     7E         MOV A,M     ; DATA FROM MEM BY HL
2451     12         STAX D      ; COPIED TO THAT POINTED BY DE
2452     23         INX H       ;
2453     13         INX D       ; ADVANCE POINTERS
2454     0B         DCX B       ; COUNT
2455     78         MOV A,B     ;
2456     B1         ORA C       ;
2457     C9         RET         ;
```

"SEND FILE FULL BLOCK"

```
245B     C5         PUSH B      ; SAVE
245C     1E10       MVI E,10    ; 16 CHAR PER LINE
245E     CDEB05     CALL 05EB   ; SEND CR-LF
2461     0E20       MVI C,20    ; SEND A SPACE
2463     CDFA07     CALL 07FA   ;
2466     7E         MOV A,M     ; GET BYTE
2467     CDC706     CALL 06C7   ; SEND AS TWO CHAR CALL 'NMOUT'
246A     C1         POP B       ;
246B     0B         DCX B       ; COUNT BYTE
246C     78         MOV A,B     ;
246D     B1         ORA C       ;
246E     C8         RZ          ; RETURN IF ALL OVER
246F     C5         PUSH B      ; ELSE
2470     23         INX H       ; DO MORE
2471     1D         DCR E       ;
2472     7B         MOV A,E     ;
2473     FE08       CPI 08      ; PUT A GAP AFTER 8 BYTES
2475     CA6124     JZ 2461     ;
2478     7B         MOV A,E     ;
2479     A7         ANA A       ;
247A     CA5C24     JZ 245C     ;
247D     C36624     JMP 2466    ;
```

"INITIALIZE FDD FORE READ/WRITE"

```
2480     CD2A0F     CALL 0F2A   ; CALL 'RESET FDC'
2483     CD000F     CALL 0F00   ; CALL 'SPECIFY'
2486     CD7001     CALL 0170   ; CALL 'FDD MOTOR ON'
2489     CD4801     CALL 0148   ; CALL 'RECLIBRATE .TR OO'
248C     C9         RET         ;
```

"DIRECT WRITE ONE CYLINDER, USED FOR TESTING"

```
2490    31A020    LXI SP,20A0 ; INIT STACK
2493    CD8024    CALL 2480   ; CALL 'INIT FDC FOR READ/WRITE'
2496    CDA023    CALL 23A0   ; CALL 'WRITE BLOCK'
2499    CD8301    CALL 0183   ; CALL 'FDD MOTOR OFF'
249C    CF        RST 1       ; WAIT MONITOR
```

"SELF COPY , RUNS IF EPROM IN 6000H"

```
24A0    31A020    LXI SP,20A0 ;
24A3    210060    LXI H,6000  ; EPROM POINTER
24A6    110021    LXI D,2100  ; RUN AREA
24A9    010008    LXI B,0800  ; BYTE COUNT
24AC    CD5063    CALL 6350   ; CALL COPY
24AF    C2AC63    JNZ 63AC    ; COPY FULL
24B2    CF        RST 1       ; WAIT MONITOR
```

"TEST 40 TRACKS"

```
24B6    FE28      CPI 28      ; 40 TRACKS OVER ?
24B8    CACD24    JZ 24CD     ; IF YES IT'S END,SHOW BY CARRY 0
24BB    CD6001    CALL 0160   ; ELSE GO NEXT TRACK
24BE    37        STC         ; CARRY ONE .
24BF    C9        RET         ; RETURN
```

"FILE TRASFER ASCII"

```
24C0    31A020    LXI SP,20A0 ; STACK
24C3    210030    LXI H,3000  ; DATA POINTER
24C6    010024    LXI B,2400  ; 9 K DATA
24C9    CD5B24    CALL 245B   ; CALL 'SEND FILE FULL BLOCK'
24CC    CF        RST 1       ; WAIT MONITOR
```

"RESET CARRY"

```
24CD    37        STC         ;
24CE    3F        CMC         ;
24CF    C9        RET         ;
```

"FORMAT "

```
24DF    34        INR M       ; MEMORY INCREMENTED,HEAD 1
24E0    CDA10E    CALL 0EA1   ; CALL 'FORMAT TRACK'
24E3    21B820    LXI H,20B8  ; HEAD NUMBER
24E6    7E        MOV A,M     ; IN ACC.
24E7    E601      ANI 01      ; HEAD 1 ?
24E9    CADF24    JZ 24DF     ; IF NOT THEN CHANGE TO HEAD 1
24EC    35        DCR M       ; IF IT WAS THEN  MAKE IT HEAD 0
24ED    2B        DCX H       ; POINT TO TRACK NO
24EE    34        INR M       ; NEXT TRACK
24EF    7E        MOV A,M     ; NUMBER IN ACC
24F0    CDB624    CALL 24B6   ; TEST LAST TRACK
24F3    DAE024    JC 24E0     ; IF NOT THEN FORMAT MORE
```

```
24F6    C9        RET          : ELSE  ALL OVER
           "DELAY LOOP"  Ø.5 SEC ?

24F7    D5        PUSH D       :
24F8    11FFFF    LXI D,FFFF   : DE HOLDS 64 K
24FB    CDF1Ø5    CALL Ø5F1    : CALL 'DELAY'
24FE    D1        POP D        :
24FF    C9        RET

         "AVERAGE, ADDS  N  VALUES ,N AS IN BC  .

                 RESULT AT 2ØBA/BBH"

25ØØ    E5        PUSH H       :
25Ø1    21ØØØØ    LXI H,ØØØØ   : INIT HL
25Ø4    22BA2Ø    SHLD 2ØBA    : AND SUM AS WELL
25Ø7    E1        POP H        :
25Ø8    D5        PUSH D       :
25Ø9    5E        MOV E,M      : DATA IN E
25ØA    16ØØ      MVI D,ØØ     :
25ØC    E5        PUSH H       :
25ØD    2ABA2Ø    LHLD 2ØBA    : GET PREVIOUS SUM
251Ø    19        DAD D        : ADD CURRENT VALUE
2511    22BA2Ø    SHLD 2ØBA    : STORE BACK SUM
2514    E1        POP H        :
2515    23        INX H        : ADVANCE TWICE TO TEST ONE
2516    23        INX H        : OF TWO CHANNELS
2517    ØB        DCX B        : COUNT
2518    78        MOV A,B      :
2519    B1        ORA C        :
251A    C2Ø925    JNZ 25Ø9     : ADD MORE
251D    D1        POP D        :
251E    C9        RET          :

         "SQUARE,DATA IN A RESULT IN DE"

2524    21ØØ28    LXI H,28ØØ   : HL POINTS TO SQUARE TABLE
2527    5F        MOV E,A      : DATA IN DE
2528    16ØØ      MVI D,ØØ     :
252A    19        DAD D        : ADD 2 TIMES TO POINT TO
252B    19        DAD D        : CORRESPONDING SQUARE
252C    5E        MOV E,M      : GET LO OF SQUARE IN E
252D    23        INX H        : AND
252E    56        MOV D,M      : HI IN D
252F    C9        RET          :

         "CALCULATE"

253Ø    3ABB2Ø    LDA 2ØBB     : HI OF SUM IS AVERAGE FOR 256
2533    CD2425    CALL 2524    : CALL 'SQUARE'
2536    2A4Ø2Ø    LHLD 2Ø4Ø    : HL GETS SUM OF SQUARES
2539    E5        PUSH H       :
253A    D5        PUSH D       : DE HAS SQUARE OF AVERAGE
253B    CD51ØF    CALL ØF51    : CALL SUBTRACT, NUMERATOR OF S4
253E    D1        POP D        : IN BC,DE HL RESTORED
```

```
253F    E1        POP H        :
2540    C9        RET          :
```

"SUM OF SQUARES"

```
2541    CD5025    CALL 2550    : CALL 'INIT SUM OF SQUARES'
2544    CD6025    CALL 2560    : CALL 'SINGLE SUM OF SQUARES'
2547    23        INX H        : MORE
2548    ØB        DCX B        : COUNT BYTE
2549    78        MOV A,B      :
254A    B1        ORA C        :
254B    C24425    JNZ 2544     : DO MORE
254E    C9        RET          :
```

" INIT SUM OF SQUARES"

```
2550    E5        PUSH H       :
2551    213F2Ø    LXI  H,2Ø3F  : SUM OF SQUARES  AT 2Ø3F/4Ø/41
2554    36ØØ      MVI M,ØØ     : SET ALL 3 BYTES
2556    23        INX H        : TO ZERO
2557    36ØØ      MVI M,ØØ     :
2559    23        INX H        :
255A    36ØØ      MVI M,ØØ     :
255C    E1        POP H        :
255D    C9        RET          :
```

"SUM OF SQUARE ONE BYTE"

```
256Ø    7E        MOV A,M      : GET BYTE
2561    E5        PUSH H       :
2562    CD2425    CALL 2524    : CALL SQUARE
2565    213F2Ø    LXI H,2Ø3F   : POINT TO LO BYTE
2568    7B        MOV A,E      : GET LO OF SQUARE
2569    8E        ADC M        : ADD
256A    77        MOV M,A      : STORE BACK
256B    23        INX H        : MID BYTE
256C    7A        MOV A,D      : ADD HI OF SQUARE
256D    8E        ADC M        : TO IT
256E    77        MOV M,A      : AND STORE BACK
256F    23        INX H        : NEXT IS MS BYTE
257Ø    3EØØ      MVI A,ØØ     : ONLY CARRY ADDED
2572    8E        ADC M        :
2573    77        MOV M,A      : STORE BACK
2574    E1        POP H        :
2575    C9        RET          :
```

" STORE SUM OF SQUARES AND SUM IN 5 CONSECUTIVE LOCATIONS"

```
258Ø    D5        PUSH D       :
2581    C5        PUSH B       :
2582    113F2Ø    LXI D,2Ø3F   : DE POINTS TO SUM OF SQUARES
2585    Ø6Ø3      MVI B,Ø3     : 3 BYTES
2587    1A        LDAX D       : GET LO BYTE
2588    77        MOV M,A      : STORE IN MEM
2589    23        INX H        : NEXT LOC
```

```
258A    13        INX D        ; BY MID BYTE
258B    Ø5        DCR B        ; THEN NEXT
258C    C28725    JNZ 2587     ; BY HI BYTE
258F    3ABA2Ø    LDA 2ØBA     ; GET LO OF SUM
2592    77        MOV M,A      ; STORE IT NEXT
2593    23        INX H        ; NEXT LOCATION
2594    3ABB2Ø    LDA 2ØBB     ; WILL HOLD
2597    77        MOV M,A      ; HI BYTE OF SUM
2598    23        INX H        ; GET READY FOR MORE
2599    C1        POP B        ;
259A    D1        POP D        ;
259B    C9        RET          ;

          "PROCESS ONE BLOCK"

259C    E5        PUSH H       ; SAVE POINTER
259D    C5        PUSH B       ; SAVE COUNT
259E    CDØØ25    CALL 25ØØ    ; CALL 'SUM'
25A1    C1        POP B        ; RESTORE COUNT
25A2    E1        POP H        ; RESTORE POINTER
25A3    CD4125    CALL 2541    ; CALL "SUM OF SQUARES'
25A6    E5        PUSH H       ; SAVE
25A7    2ABC2Ø    LHLD 2ØBC    ; GET RESULT POINTER
25AA    CD8Ø25    CALL 258Ø    ; STORE ONE SET
25AD    22BC2Ø    SHLD 2ØBC    ; SAVE RESULT POINTER
25BØ    E1        POP H        ;
25B1    C9        RET          ;

          " BLOCK RESULT"

25B2    CD9C25    CALL 259C    ; CALL  ' PROCESS ONE BLOCK'
25B5    E5        PUSH H       ;
25B6    CD3325    CALL 2533    ; CALL ' CALCULATE'
25B9    CDE225    CALL 25E2    ; CALL ' S4 COUNT'
25BC    E1        POP H        ;
25BD    C9        RET          ;

          "PROCESS 32 BLOCKS"

25CØ    21Ø3Ø     LXI H,3Ø1Ø   ; HL POINTS TO DATA BEGINNING
25C3    22BC2Ø    SHLD 2ØBC    ; IT BECOMES RESULT POINTER
25C6    Ø61Ø      MVI B,1Ø     ; 16 BLOCKS
25C8    C5        PUSH B       ; SAVE BLOCK NUMBER
25C9    Ø1ØØØ1    LXI B,Ø1ØØ   ; BC HOLDS BLOCK LENGTH
25CC    CDB225    CALL 25B2    ; CALL 'BLOCK RESULT'
25CF    C1        POP B        ;
25DØ    Ø5        DCR B        ; GO NEXT BLOCK
25D1    21ØØ32    LXI H,32ØØ   ; IT SHOULD START AT 32ØØH
25D4    C5        PUSH B       ; SAVE BLOCK COUNT
25D5    Ø1ØØØ1    LXI B,Ø1ØØ   ; BLOCK LENGTH
25D8    CDB225    CALL 25B2    ; CALL 'BLOCK RESULT'
25DB    C1        POP B        ;
25DC    Ø5        DCR B        ; COUNT BLOCK NUMBER
25DD    C2D425    JNZ 25D4     ; DO ALL 32 BLOCKS
25EØ    C9        RET          ;
```

" S4 COUNT.BC NUMERATOR , DE DENOMINATOR "

```
25E2   7A        MOV A.D        ; HI OF AVG SQUARE
25E3   B7        ORA A          ; ZERO MEANS LOW SIGNAL
25E4   C8        RZ             ; IGNORE IT
25E5   78        MOV A.B        ; ELSE TEST HI OF NUMERATOR
25E6   B7        ORA A          ; NON ZERO RESULT SHOWS
25E7   C2F425    JNZ 25F4       ; S4 HIGH
25EA   79        MOV A.C        ; FOR B=Ø TEST C
25EB   BA        CMP D          ; IF C LARGER
25EC   D2F425    JNC 25F4       ; THEN S4 6% UP
25EF   37        STC            ;
25FØ   3F        CMC            ; CARRY ZERO
25F1   17        RAL            ; ELSE MULTIPLY C BY 2
25F2   BA        CMP D          ; AGAIN TEST
25F3   D8        RC             ; S4 LESS THAN 4%. NO GOOD
25F4   3AB22Ø    LDA 2ØB2       ; GET S4 COUNT
25F7   3C        INR A          ; INCREMENT IT
25F8   32B22Ø    STA 2ØB2       ; AND STORE
25FB   C9        RET            ;
```

"TEST SQUARE TABLE AND OTHER SUBROUTINES"

```
26ØØ   31AØ2Ø    LXI SP.2ØAØ    ; INIT STACK .
26Ø3   CDBØ26    CALL 26BØ      ; CALL GENERATE SQUARE TABLE'
26Ø6   21ØØ3Ø    LXI H.3ØØØ     ; MEM POINTER
26Ø9   Ø1ØØØ1    LXI B.Ø1ØØ     ; BLOCK COUNT
26ØC   E5        PUSH H         ;
26ØD   C5        PUSH B         ; SAVE POINTER & COUNT
26ØE   CD4125    CALL 2541      ; CALL 'SUM OF SQUARES'
2611   C1        POP B          ; RESTORE COUNT
2612   E1        POP H          ; AND POINTER
2613   CDØØ25    CALL 25ØØ      ; CALL 'SUM'
2616   CD3Ø25    CALL 253Ø      ; CALL 'SUBTRACT'
2619   CF        RST 1          ; WAIT 'MONITOR'
```

"SQUARE BY ADDITION . DATA IN A REG RESULT IN DE "

```
268Ø   E5        PUSH H         ;
2681   21ØØØØ    LXI H,ØØØØ     ; INIT HL
2684   F5        PUSH PSW       ; SAVE DATA
2685   5F        MOV E.A        ; DATA IN DE
2686   16ØØ      MVI D.ØØ       ;
2688   ØØ        NOP            ;
2689   B7        ORA A          ; TEST A OR DATA FOR ZERO
268A   CA9326    JZ 2693        ; ZERO MEANS DO NOTHING
268D   19        DAD D          ; ADD DATA TO ITSELF
268E   3D        DCR A          ; COUNT DATA
268F   C28D26    JNZ 268D       ; ADD AS MANY TIMES
2692   EB        XCHG           ; DE HAS RESULT
2693   F1        POP PSW        ; RESTORE INPUT
2694   E1        POP H          ; AND MEM POINTER
2695   C9        RET            ;
```

" STORE SQUARE "

```
26A0    E5        PUSH H      ; SAVE TABLE POINTER
26A1    D5        PUSH D      ; SAVE SQUARE
26A2    5F        MOV E,A     ; DATA IN E REG
26A3    1600      MVI D,00    ; D HAS 00
26A5    19        DAD D       ; HL POINTS TO TABLE ADDRESS
26A6    19        DAD D       ; CORRESPONDING TO DATA
26A7    D1        POP D       ; GET BACK SQUARE
26A8    73        MOV M,E     ; STORE LO
26A9    23        INX H       ; NEXT LOCATION
26AA    72        MOV M,D     ; STORE HI
26AB    E1        POP H       ;
26AC    C9        RET         ;
```

"GENERATE SQUARE TABLE, AT 2800H UP"

```
26B0    210028    LXI H,2800  ; HL POINTS TO SQUARE TABLE ADDRESS
26B3    3E00      MVI A,00    ; FIRST DATA 00
26B5    CD8026    CALL 2680   ; CALL 'SQUARE BY ADD'
26B8    CDA026    CALL 26A0   ; CALL  'STORE SQUARE'
26BB    3C        INR A       ; INCREMENT DATA
26BC    C2B526    JNZ 26B5    ; REPEAT FOR UP TO FFH
26BF    C9        RET         ;
```

"FILL MEMORY WITH TRIANGLULAR DATA"

A HAS LOWEST VALUE , D STEP SIZE , E NUMBER OF STEPS PER CYCLE

```
26C0    77        MOV M,A     ; STORE VALUE
26C1    D5        PUSH D      ; SAVE
26C2    7E        MOV A,M     ; GET CURRENT VALUE
26C3    82        ADD D       ; ADD ONE STEP INTERVAL
26C4    23        INX H       ; POINT TO NEXT
26C5    77        MOV M,A     ; STORE NEW VALUE
26C6    0B        DCX B       ; COUNT BYTE
26C7    78        MOV A,B     ;
26C8    B1        ORA C       ;
26C9    CAE826    JZ 26E8     ; ALL OVER ?
26CC    1D        DCR E       ; ELSE COUNT STEP NUMBER
26CD    C2C226    JNZ 26C2    ; REPEAT FOR ALL STEPS
26D0    D1        POP D       ;
26D1    D5        PUSH D      ;
26D2    7E        MOV A,M     ;
26D3    92        SUB D       ; AFTER RISIG EDGE ,IT IS FALLING
26D4    23        INX H       ; TO GENERATE TRIANGLE
26D5    77        MOV M,A     ;
26D6    0B        DCX B       ; BYTE COUNT
26D7    78        MOV A,B     ;
26D8    B1        ORA C       ;
26D9    CAE826    JZ 26E8     ; DATA FULL THEN END
26DC    1D        DCR E       ; ELSE CONTINUE FOR EQUAL STEPS
26DD    C2D226    JNZ 26D2    ; IN FALLING EDGE TOO
26E0    D1        POP D       ;
```

```
26E1     C3C126    JMP 26C1       : AFTER LOWEST VALUE AGAIN RISE
            "JUMP HERE FOR END"

26E8     D1        POP D
26E9     C9        RET

            "FILL 3000 UP BY TRUANGULAR DATA"

26EC     31A020    LXI SP,20A0  ; STACK INIT
26EF     210030    LXI H,3000   ; MEM POINTER
26F2     010020    LXI B,2000   ; 8092 BYTES
26F5     110401    LXI D,0104   ; D= INCREMENT ,E NO OF STEPS
26F8     3E0F      MVI A,0F     ; A= LOWEST VALUE
26FA     CDC026    CALL 26C0    ; CALL 'FILL'
26FD     CF        RST 1        : WAIT MONITOR
```

APPENDIX N

PRLBUS SIGNALS

Connector type 86Ø7 Ø44 21 14 O/E/N make ,44 pin Dual readout,Ø.156 inch spacing

Pin No.(Component Side)		Pin No. (Solder Side)	
1	GROUND (Bottom most)	1	GROUND
2	DØ	2	AØ
3	D1	3	A1
4	D2	4	A2
5	D3	5	A3
6	D4	6	A4
7	D5	7	A5
8	D6	8	A6
9	D7	9	A7
1Ø	MR*	1Ø	A8
11	MW*	11	A9
12	IOR*	12	A1Ø
13	IOW*	13	A11
14	NC	14	A12
15	NC	15	A13
16	NC	16	A14
17	NC	17	A15
18	INTA*	18	CLOCK OUT
19	NC	19	HLDA
2Ø	INTR	2Ø	HOLD
21	RESET OUT	21	RESET (Button)
22	+5V (TOPMOST)	22	+5V

*Indicates Inverted Signals

APPENDIX O

FLOPPY DISK DRIVE CONNECTOR PINOUTS

Pin No.	Signal Designation	Data Direction
1-33 (odd)	Ground	--
2,4,6	Unused	--
8	Index	FDD To FDC
1Ø	Motor Enable A	FDC To FDD
12	Drive Select B	FDC To FDD
14	Drive Select A	FDC To FDD
16	Motor Enable B	FDC To FDD
18	Direction (Step)	FDC To FDD
2Ø	Step Pulse	FDC To FDD
22	Write Data	FDC To FDD
24	Write Enable	FDC To FDD
26	Track ØØ	FDD To FDC
28	Write Protect	FDD To FDC
3Ø	Read Data	FDD To FDC
32	Select Head 1	FDC To FDD
34	Unused	--

APPENDIX P

PIN OUTS OF THE EPROM PROGRAMMER PORT

34 pin double row Ø.1" connector (same as FDD interface)

Pin No.	Signal	Pin No.	Signal
1	+5 V	2	+5 V
3	AØ (addr)	4	A8
5	A1	6	A9
7	A2	8	A1Ø
9	A3	1Ø	A11
11	A4	12	A12
13	A5	14	A13
15	A6	16	A14
17	A7	18	A15
19	DØ (data)	2Ø	D1
21	D2	22	D3
23	D4	24	D5
25	D6	26	D7
27	CØ(control)	28	C1
29	C2	3Ø	C3
31	C4	32	NC
33	GND	34	GND

Notes: a)C5 disables data port out put drivers.data port is bidirectional.

b)CØ-C5 at F7H

c)DØ-D7 at F4H

d)AØ-A7 at F5H

e)A8-A15 at F6H

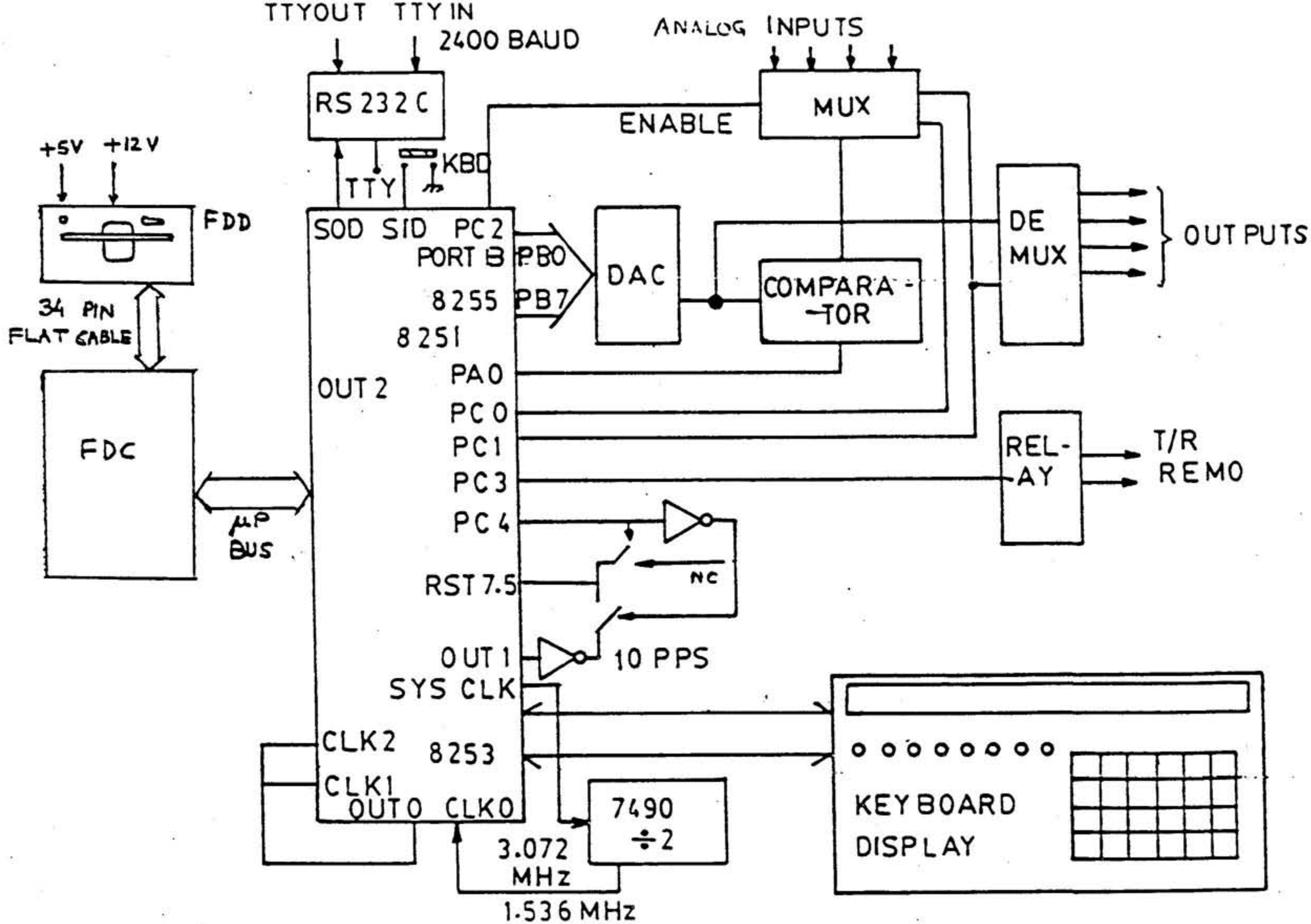


FIG.1 SYSTEM BLOCK DIAGRAM

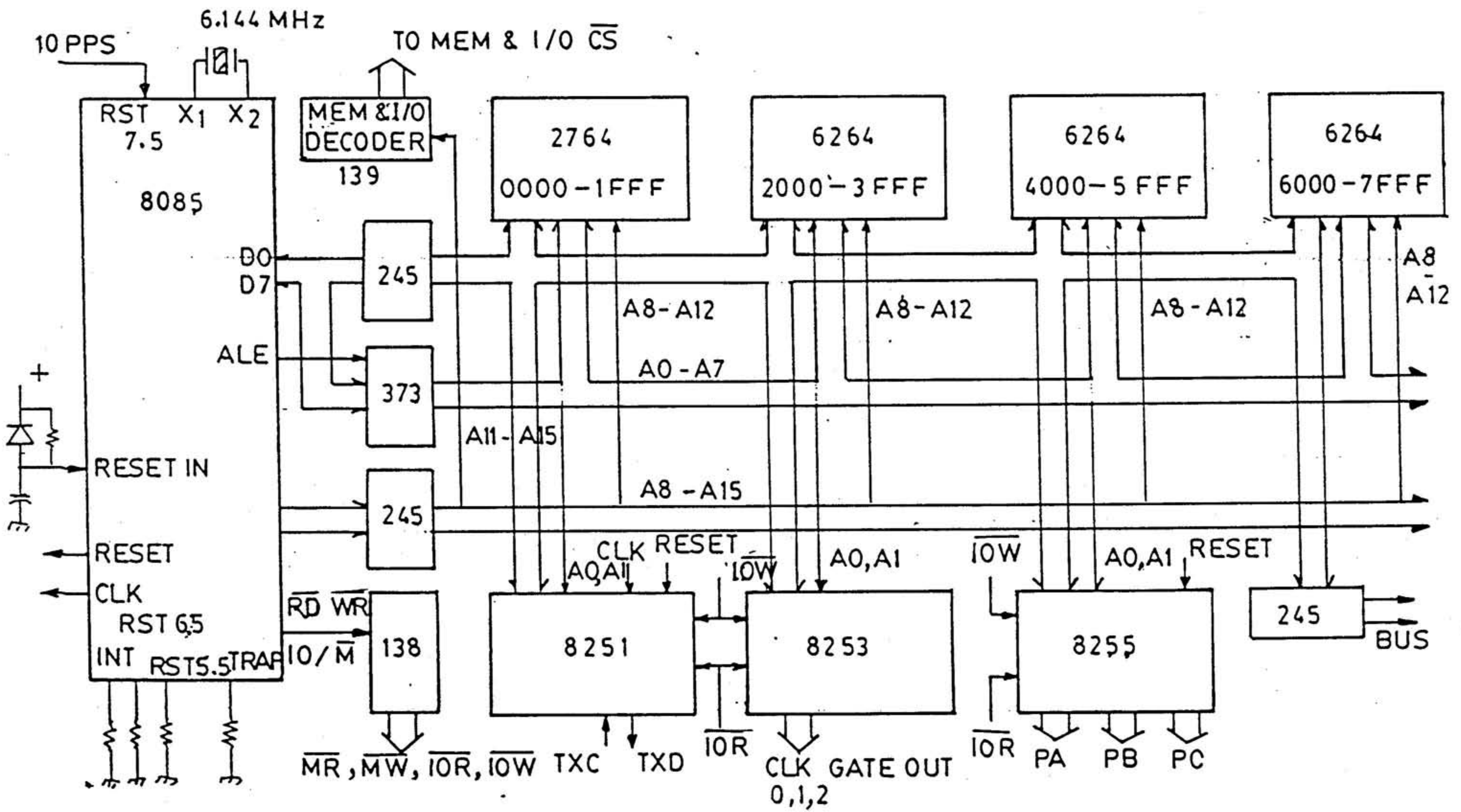


FIG. 2 MICROCOMPUTER BLOCK DIAGRAM

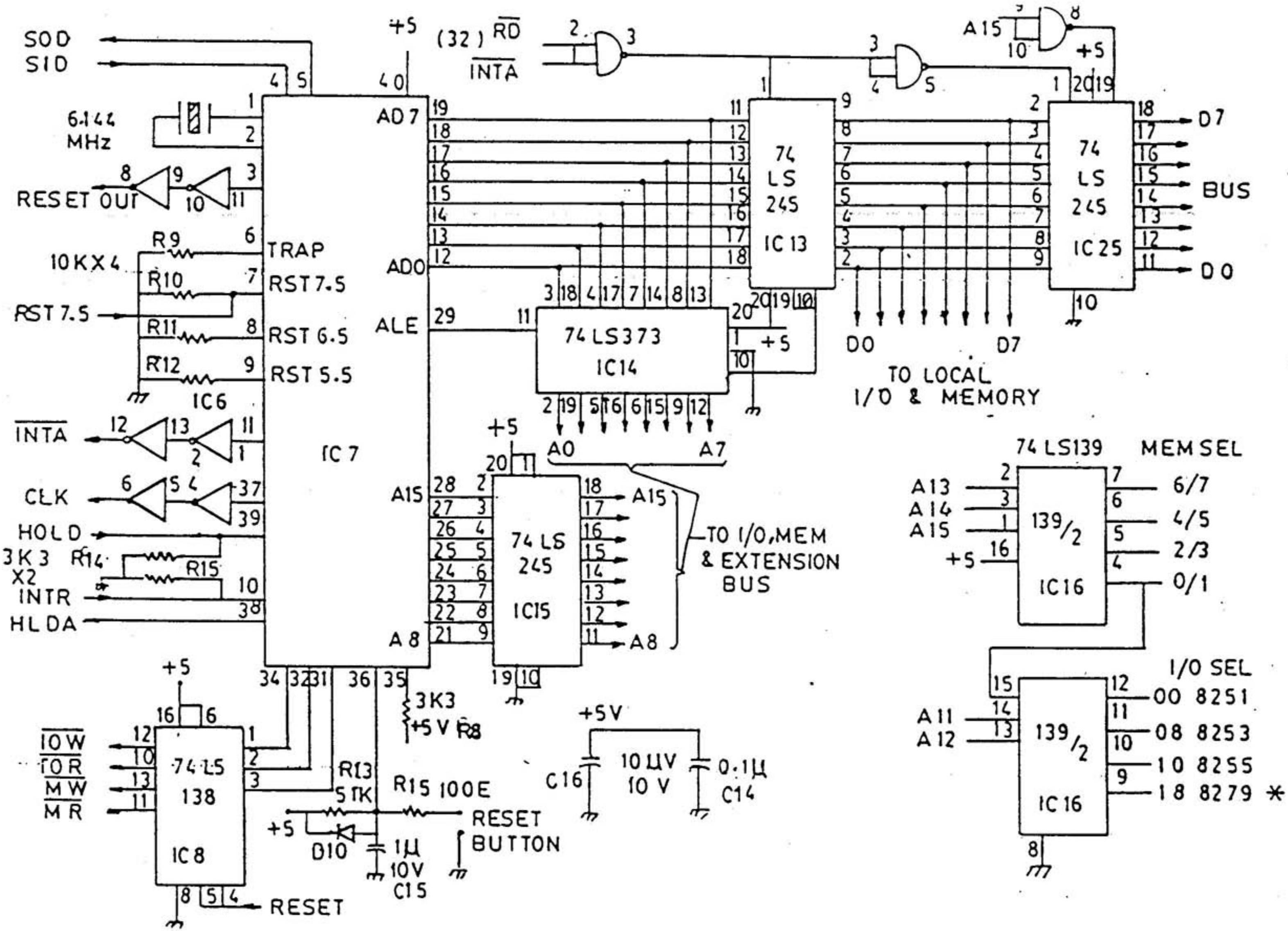


FIG.-3 MICROPROCESSOR, BUFFERS, MEMORY & I/O DECODERS

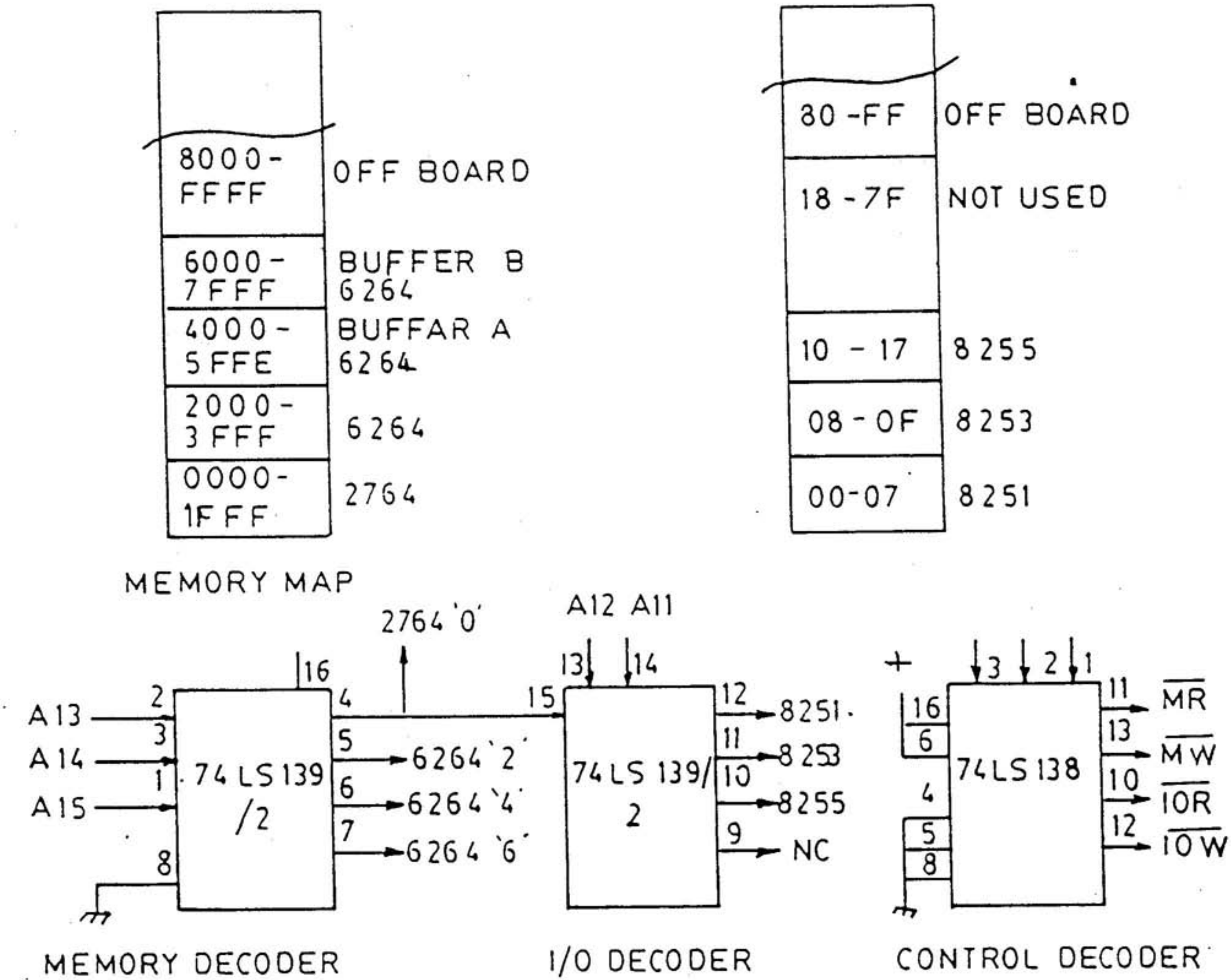


FIG.-4 MEMORY & I/O MAP, DECODER LOGIC

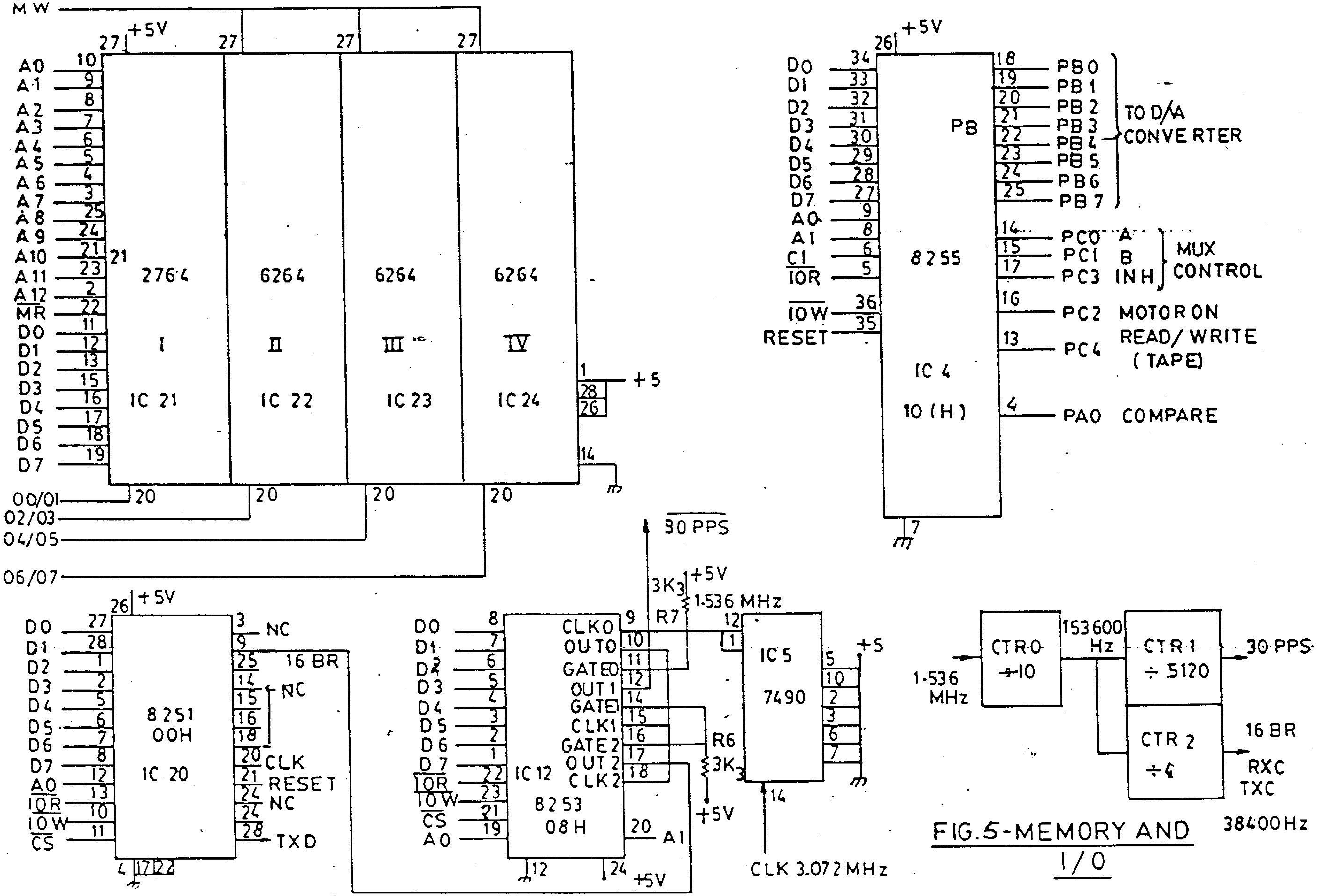


FIG.5-MEMORY AND I/O

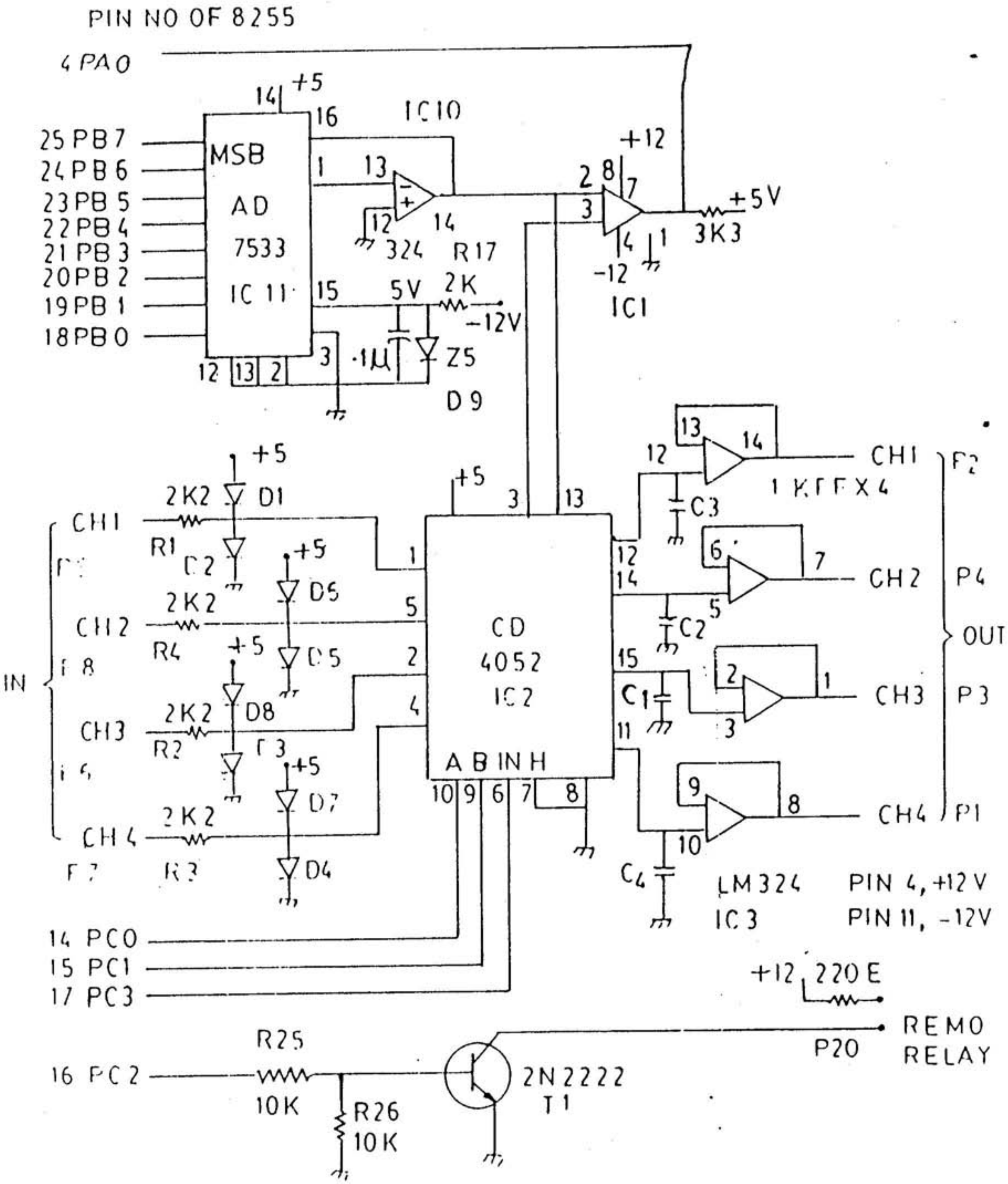


FIG. 6 -D/A & A/D CONVERTER, MULTIPLEXER

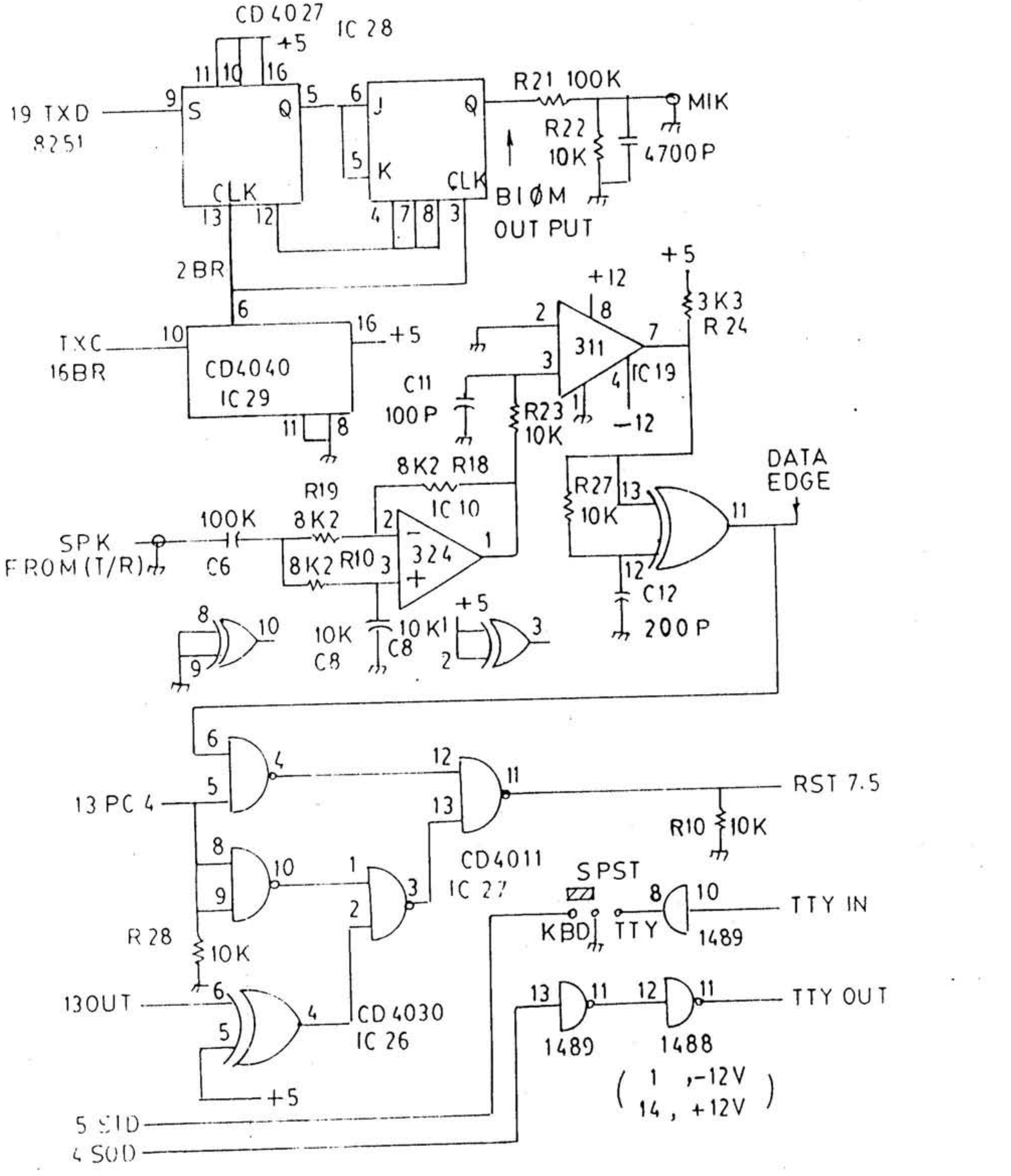


FIG. 7 CASSETTE TAPE READ WRITE & RS 232 C INTERFACE

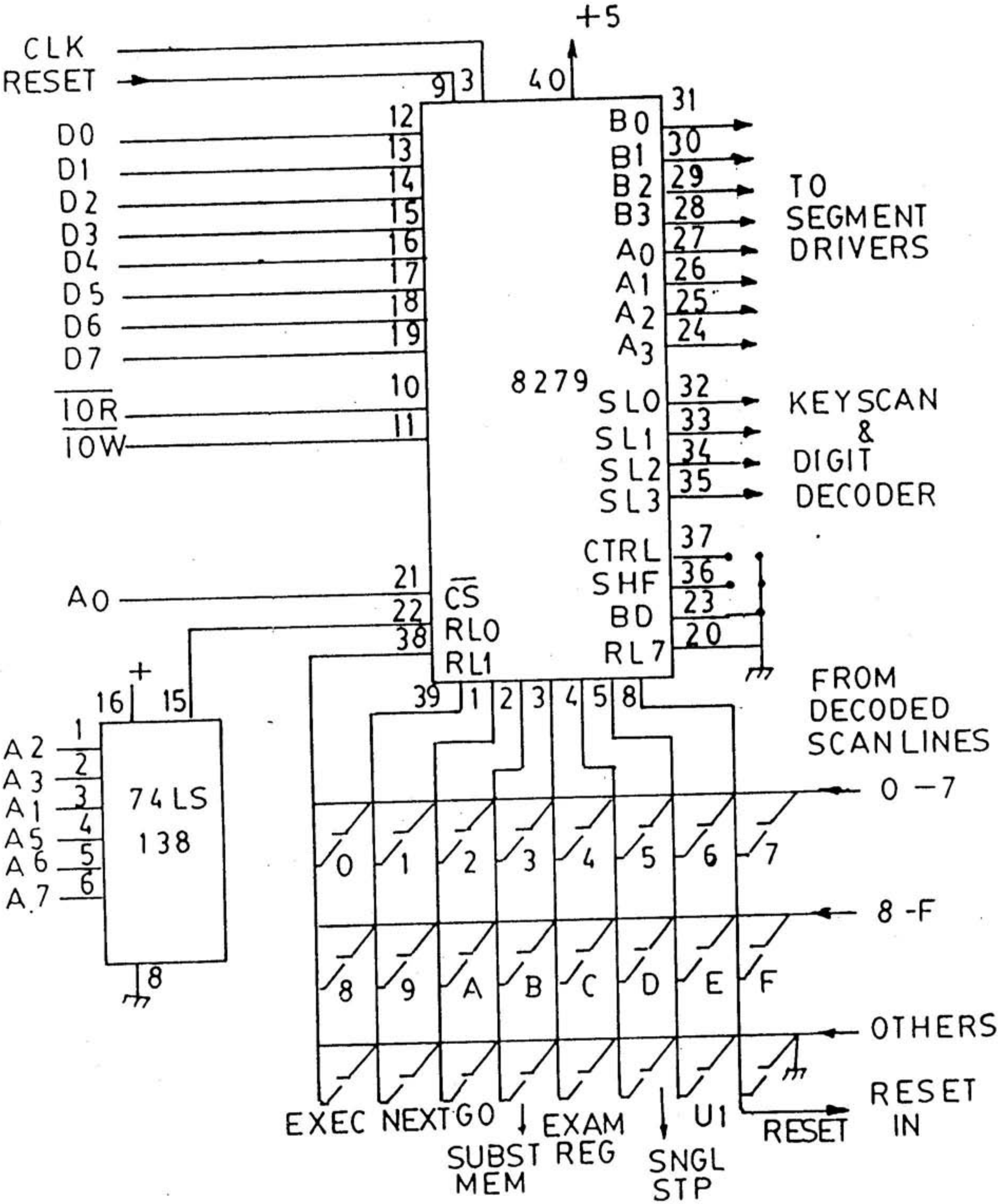


FIG. 8 - KEY SCAN CKT & KEY BOARD

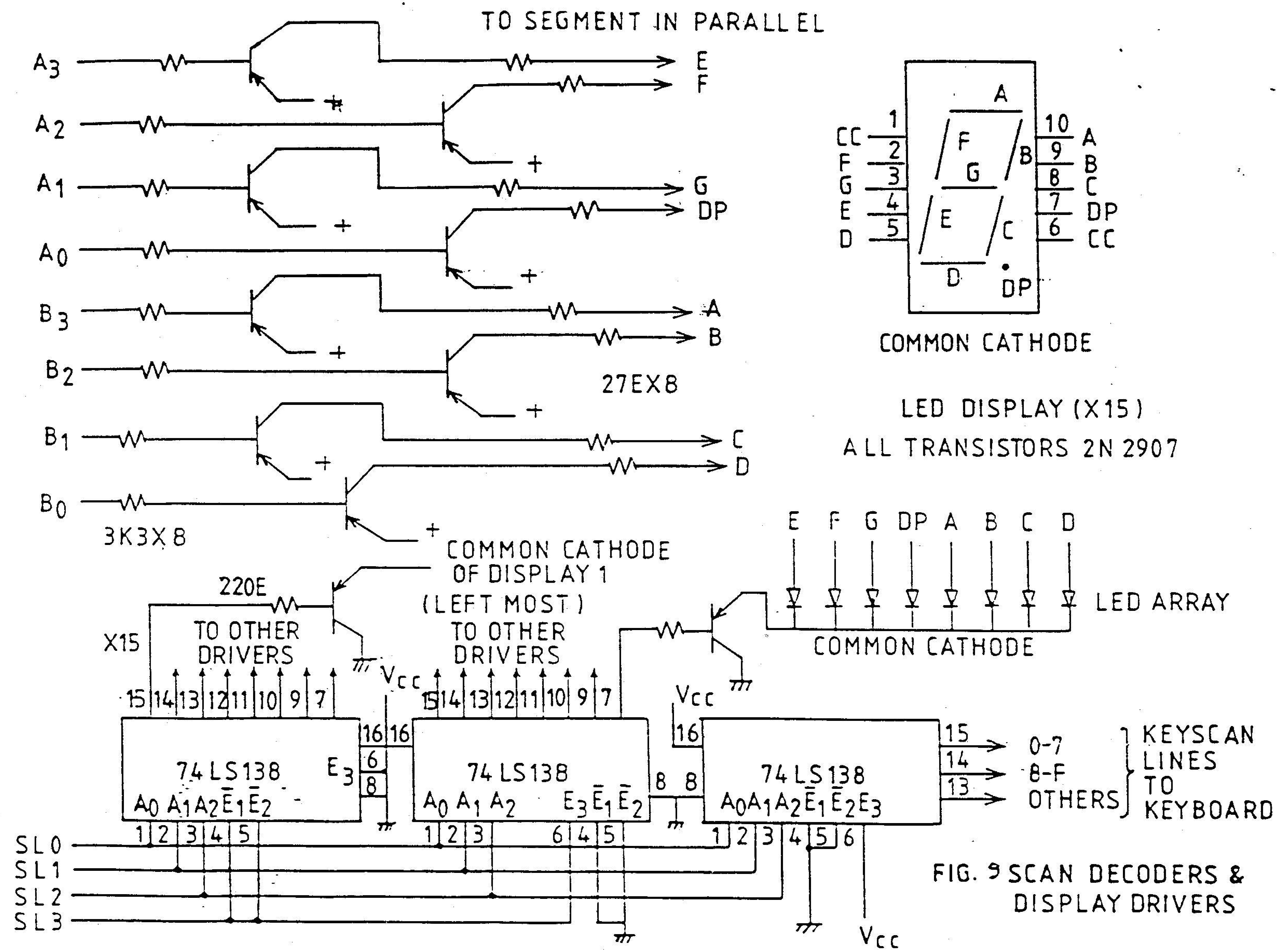
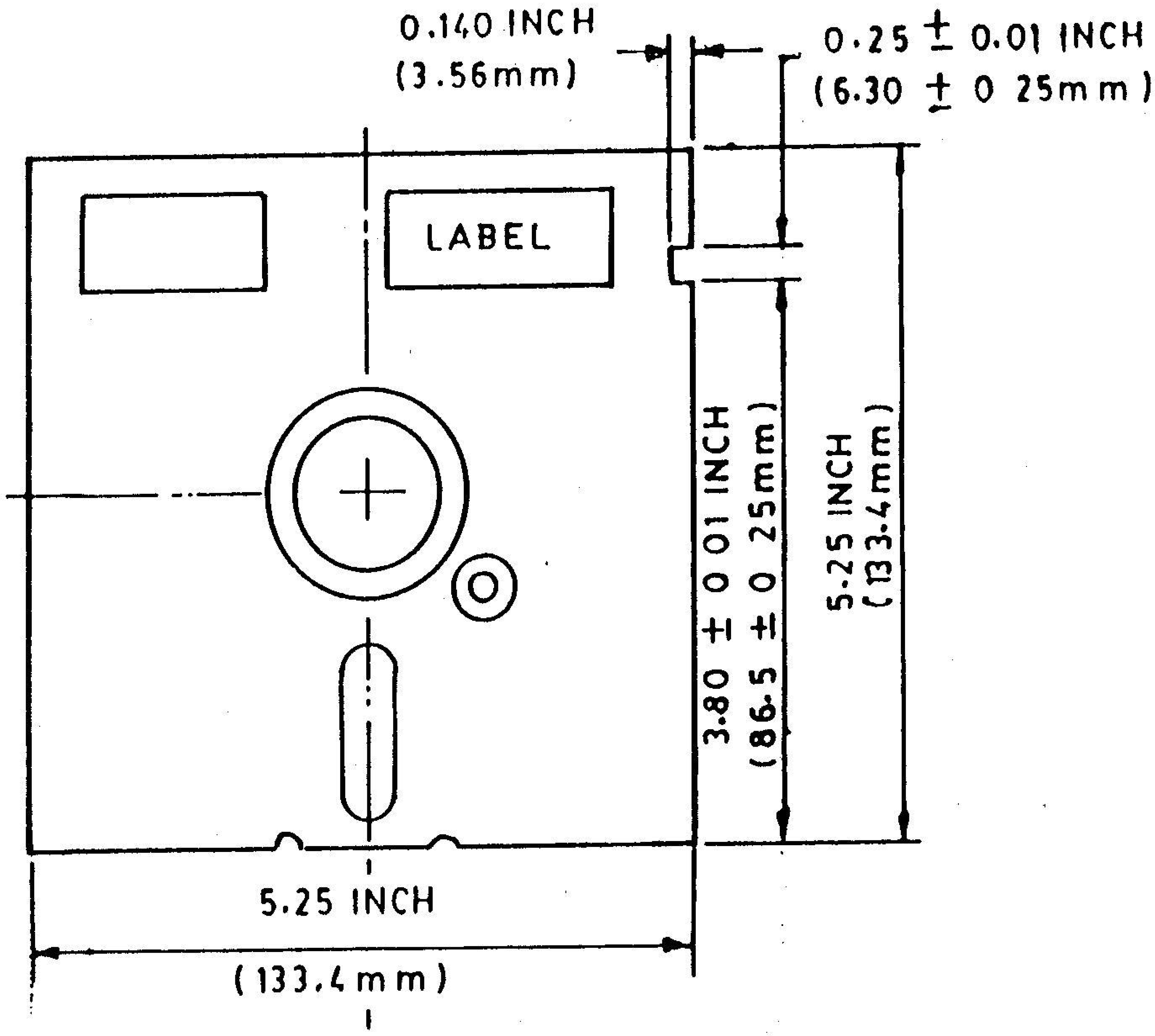


FIG. 9 SCAN DECODERS & DISPLAY DRIVERS



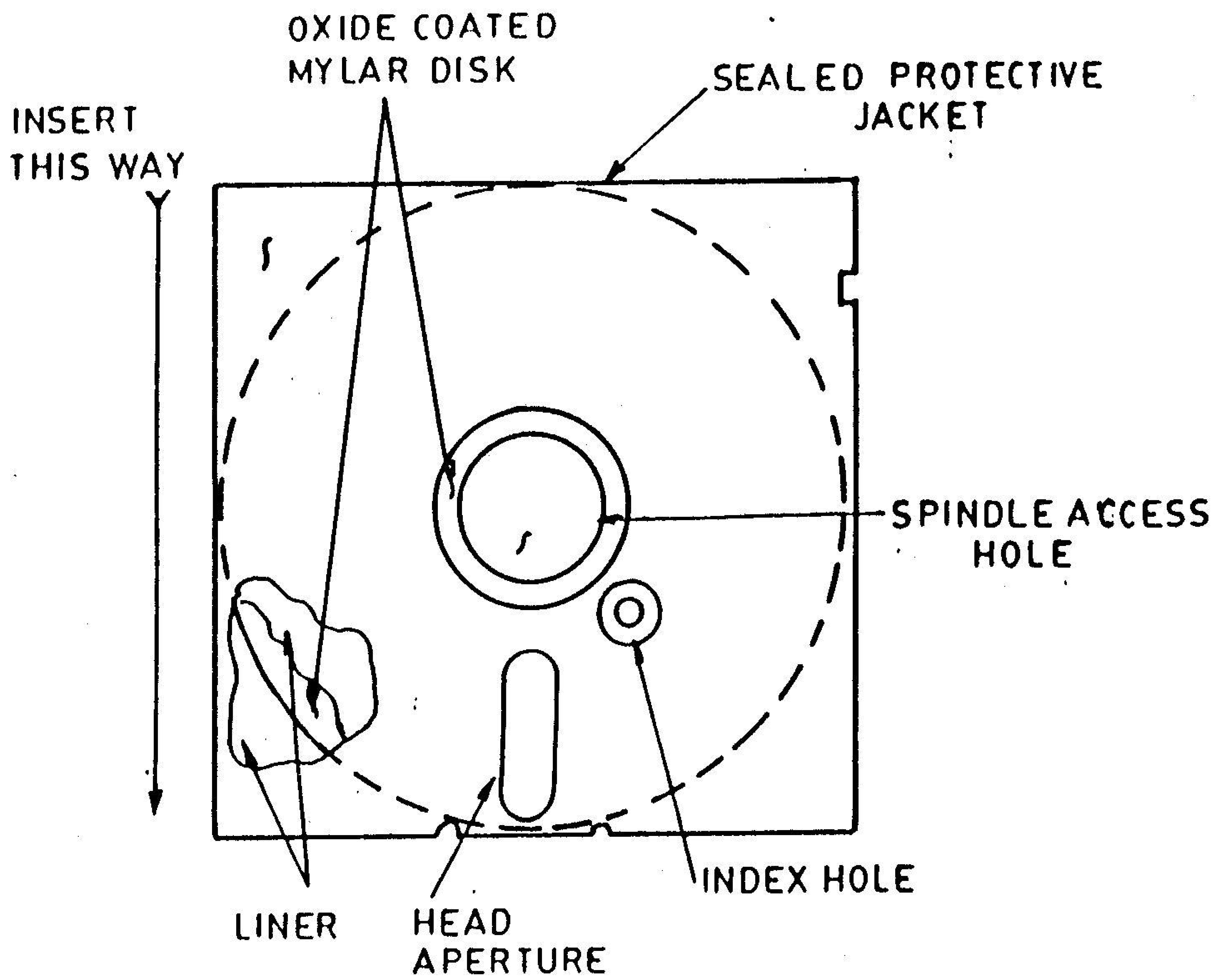


FIG.10-FLOPPY DISK

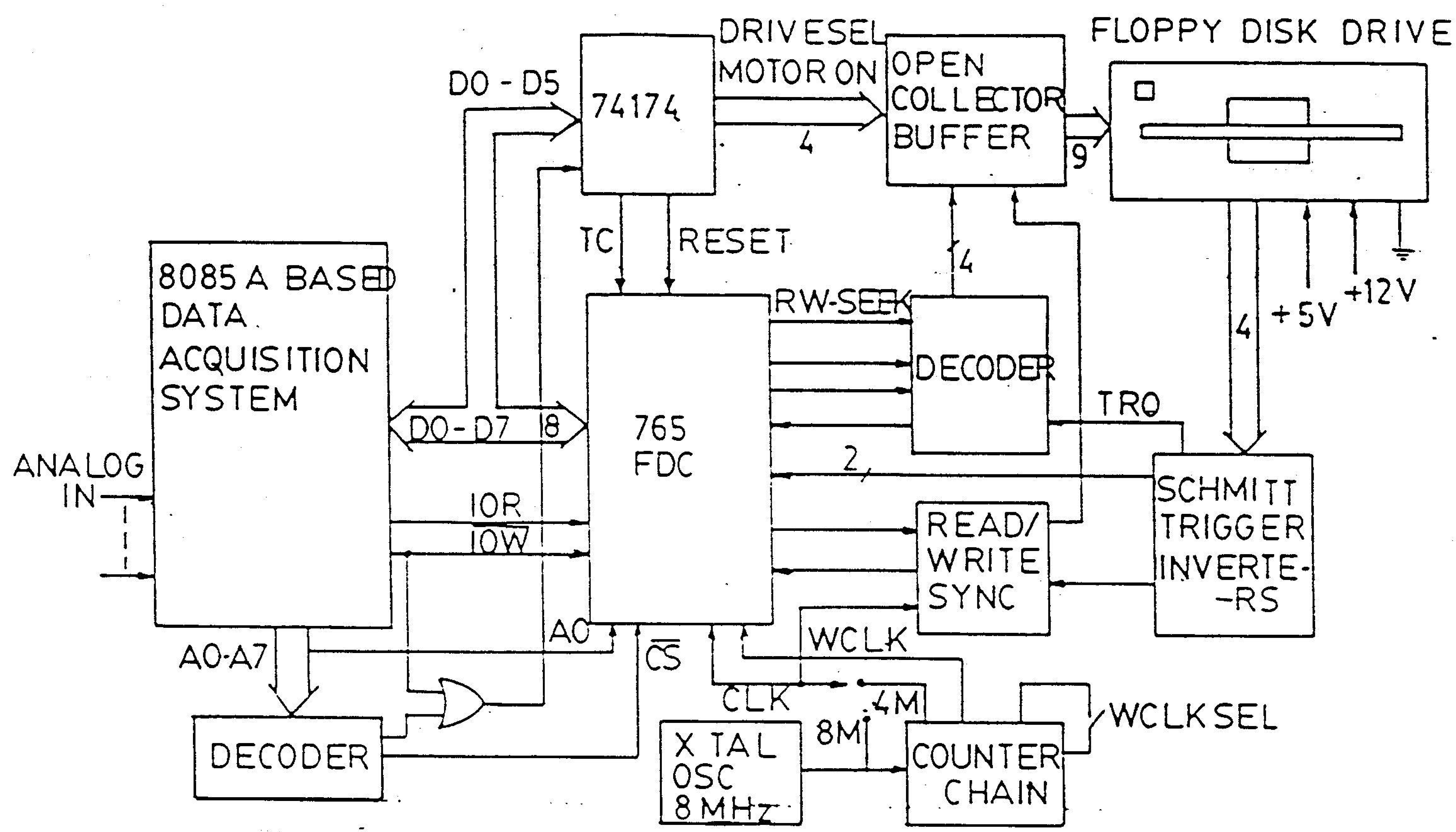


FIG.11 - BLOCK DIAGRAM OF DATA STORAGE SYSTEM

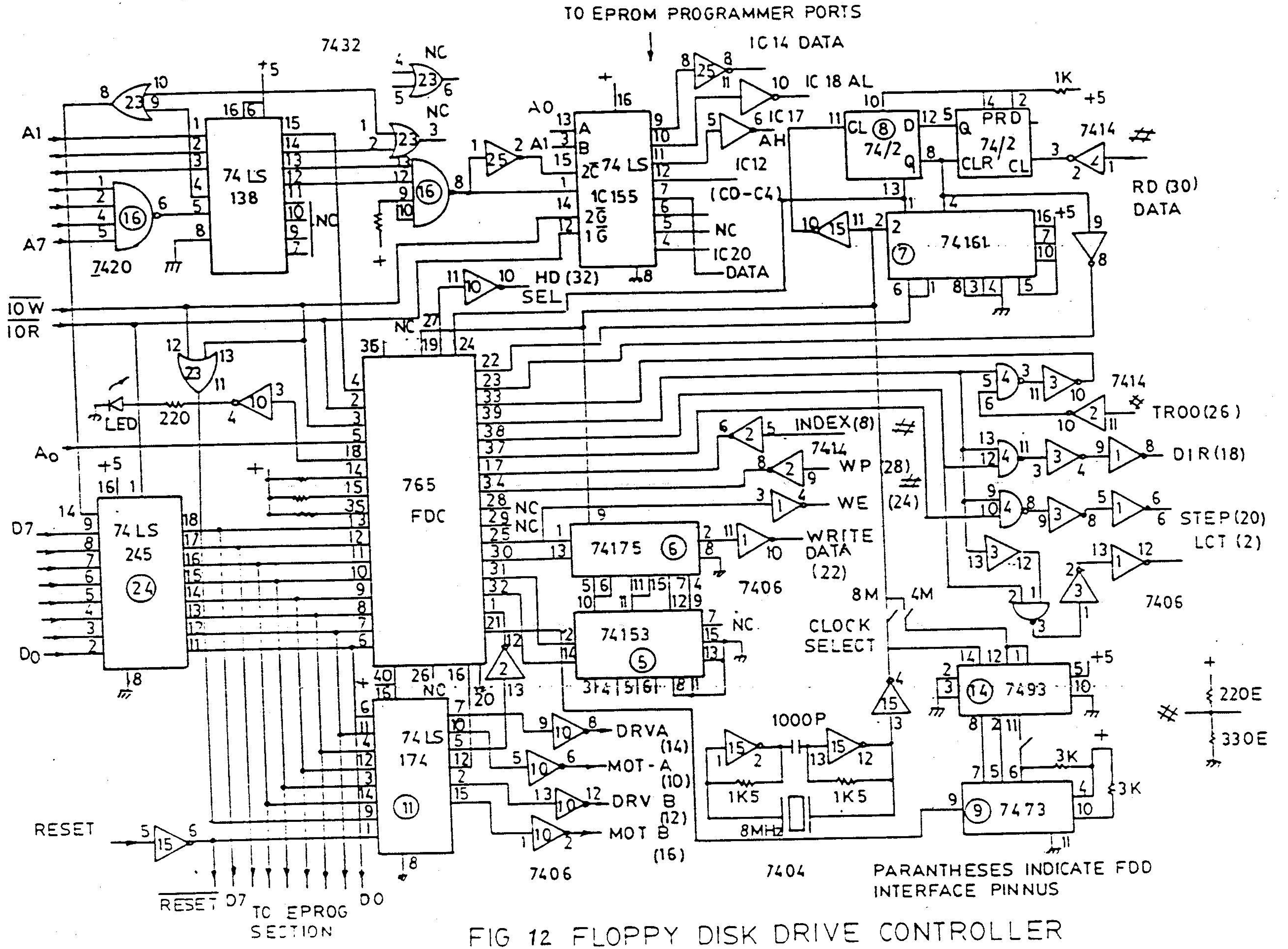


FIG 12 FLOPPY DISK DRIVE CONTROLLER

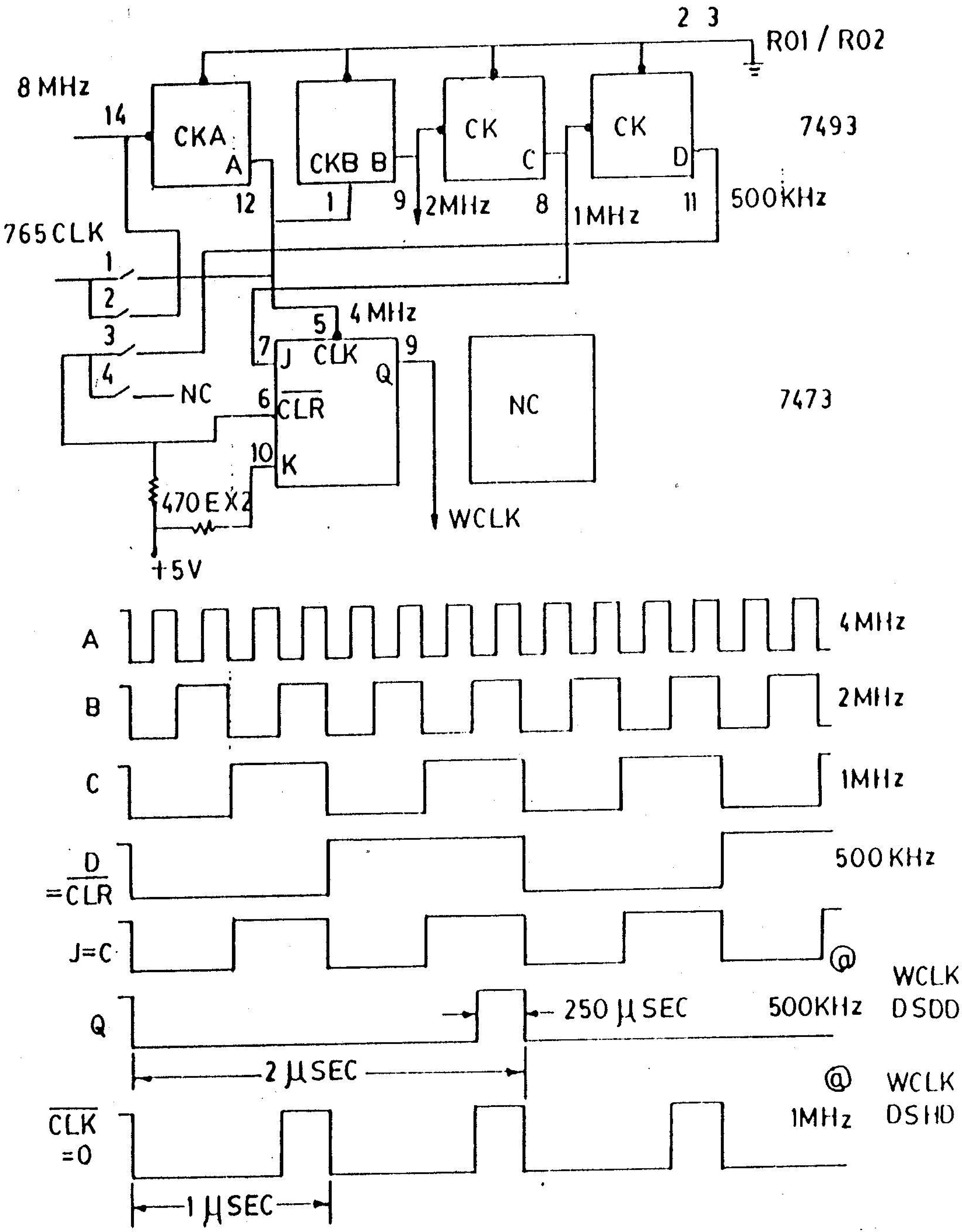
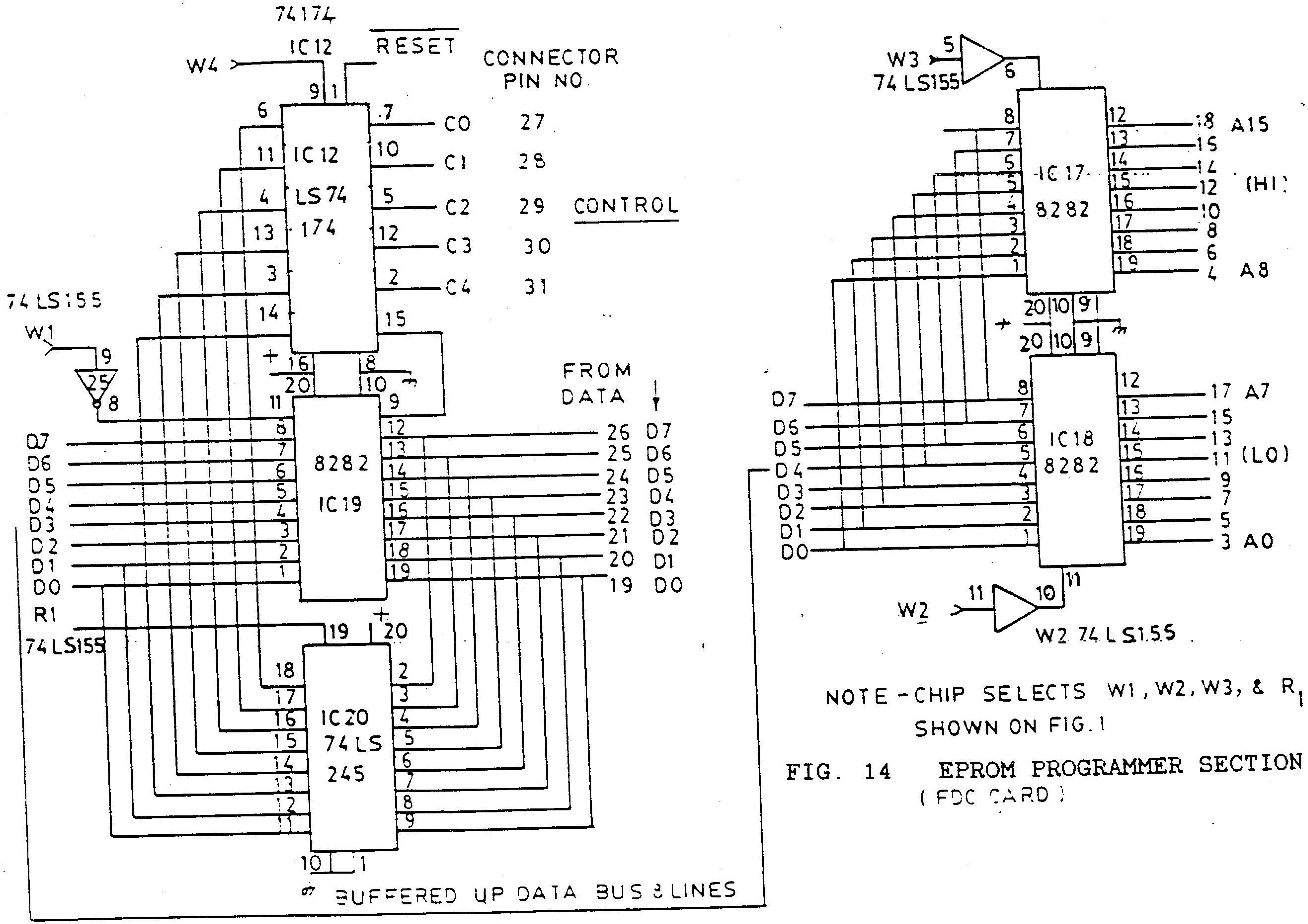


FIG. 13 - CLOCK SIGNALS FOR FDC AND DATA SYNCHRONIZATION



NOTE - CHIP SELECTS W1, W2, W3, & R_1 SHOWN ON FIG. 1

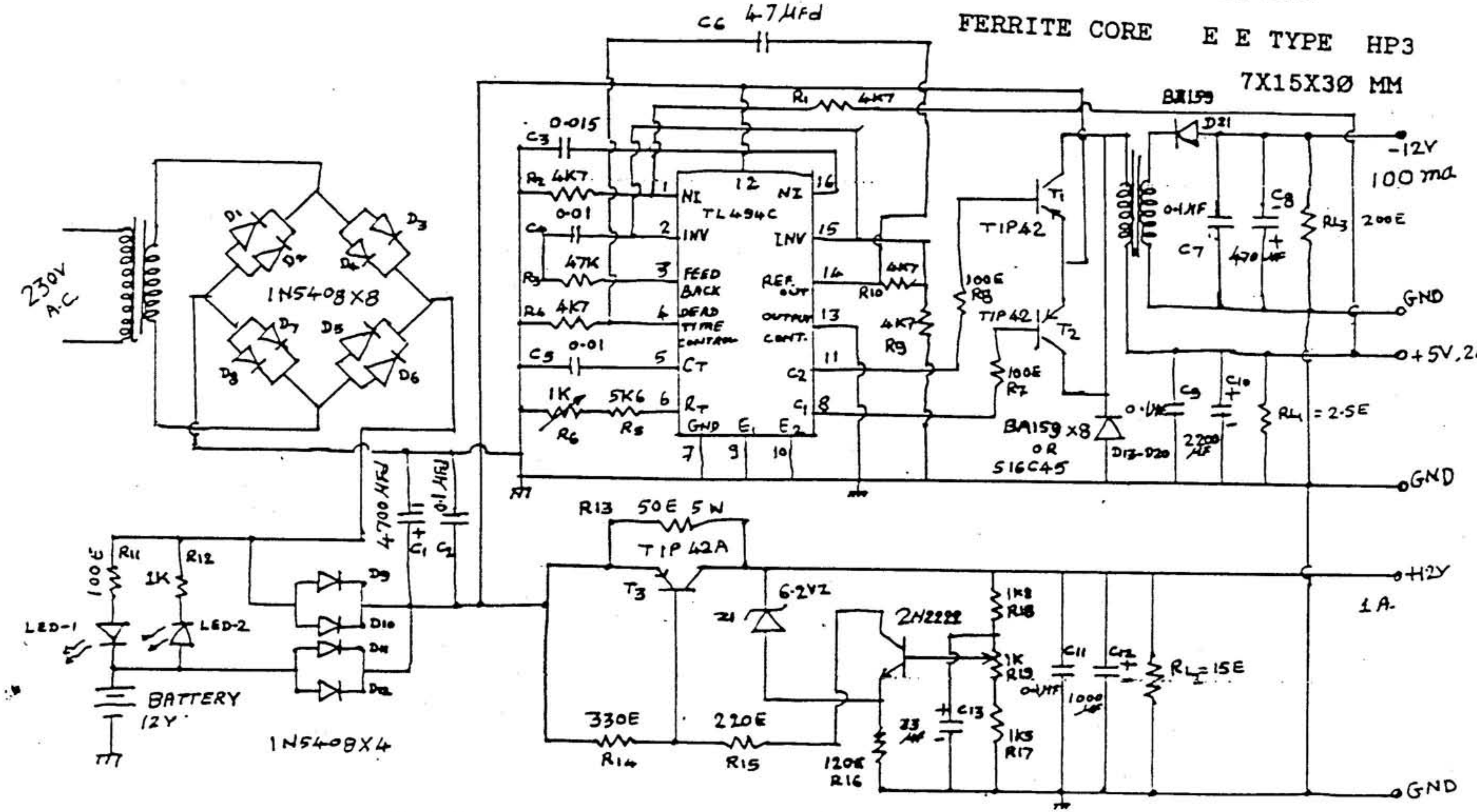
FIG. 14 EPROM PROGRAMMER SECTION (FDC CARD)

PRIMARY : 85.5 UH BIFILAR 2Ø TURNS 18 SWG

SECONDARY : 5Ø1 UH SINGLE WOUND 42 TURNS 24 SWG

FERRITE CORE E E TYPE HP3

7X15X3Ø MM



(+5 V AND -12 V SMPS , +12 V LINEAR REGULATED SUPPLY)

Fig. 15 CIRCUIT DIAGRAM OF POWER SUPPLY

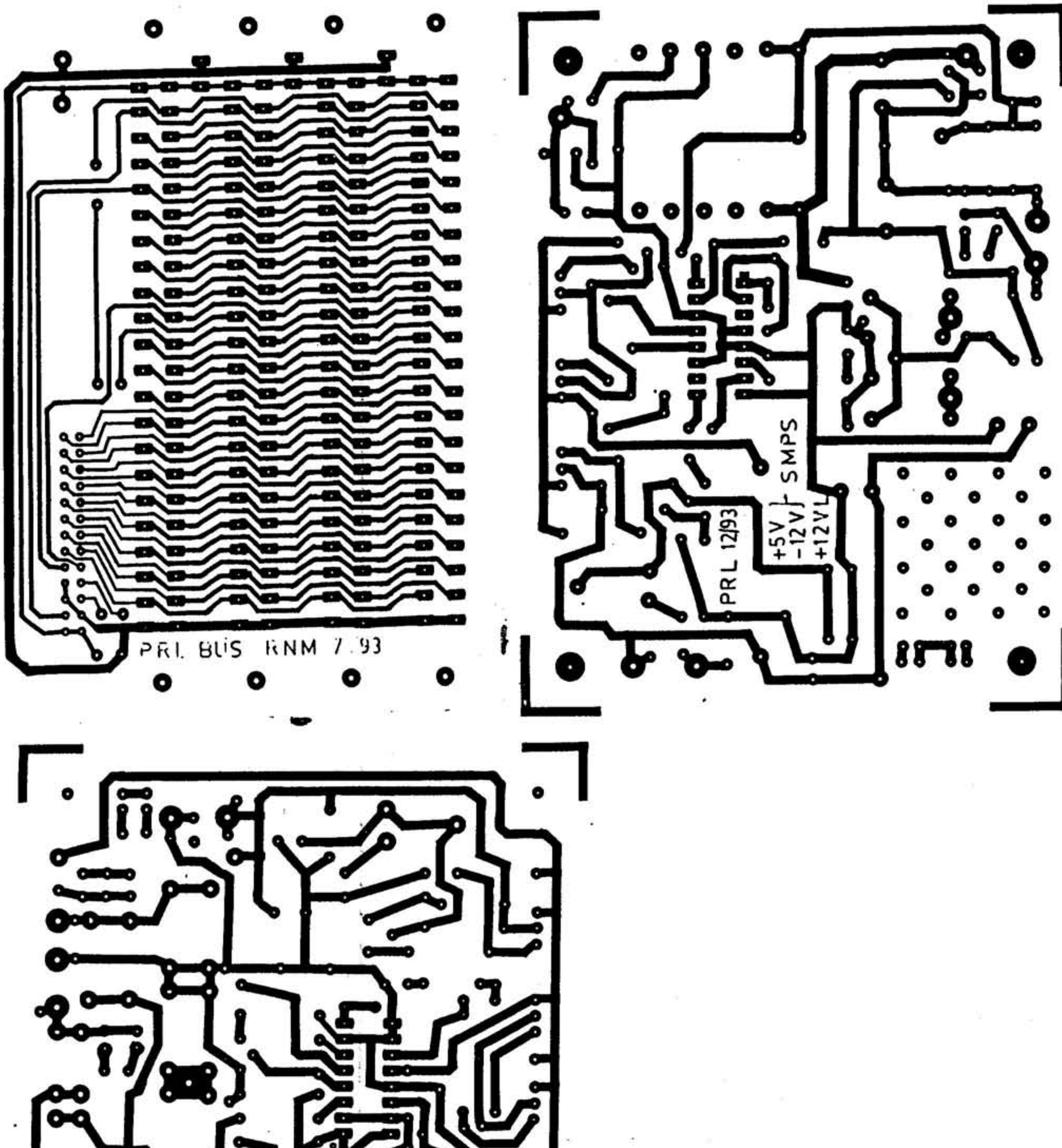


FIG. 16 PCB LAYOUT OF SMPS AND BACKPLANE BUS

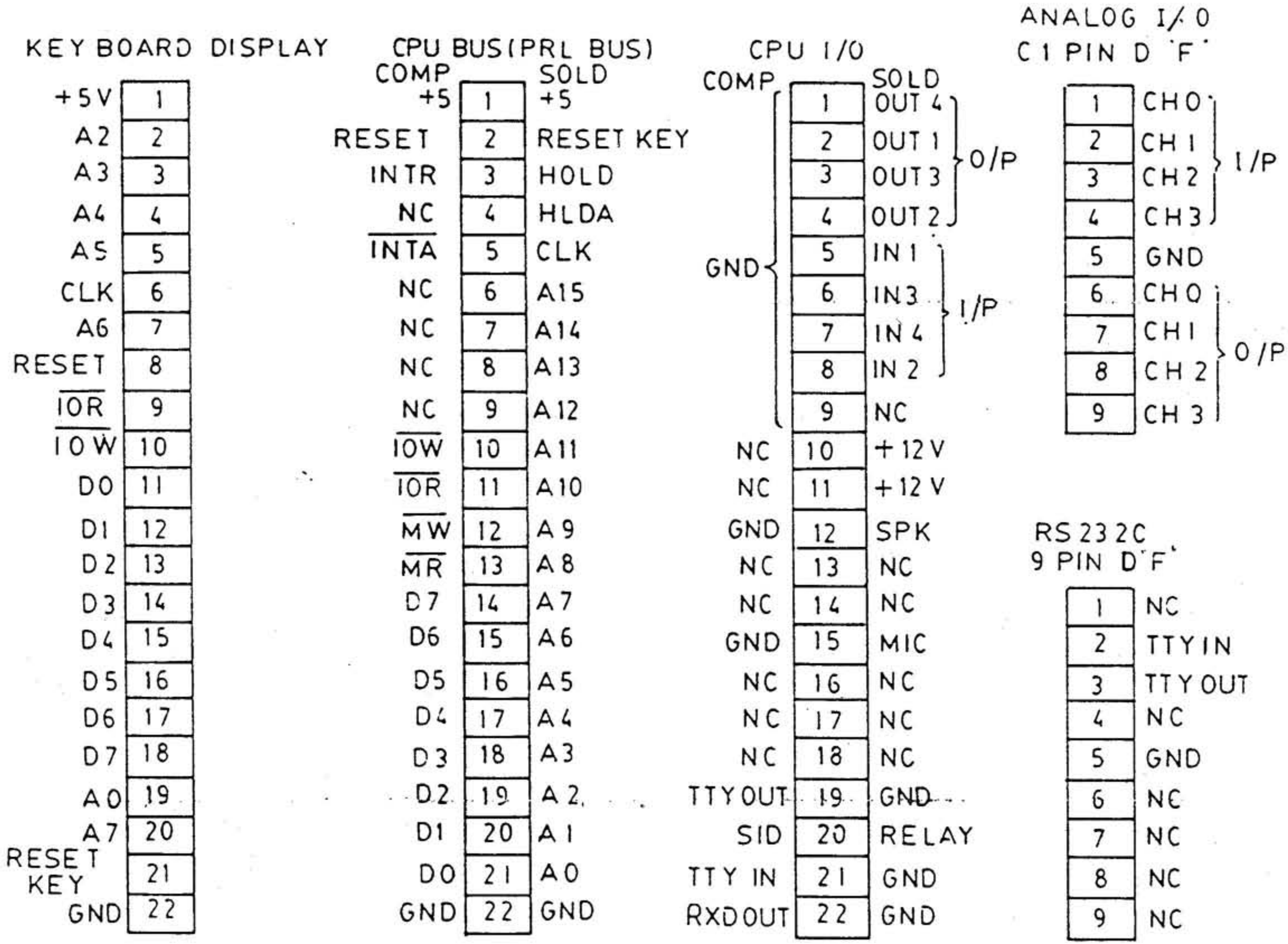


FIG. 17 INTER CONNECTION DETAILS

DRN: Panchal DATE - 19 .2/9

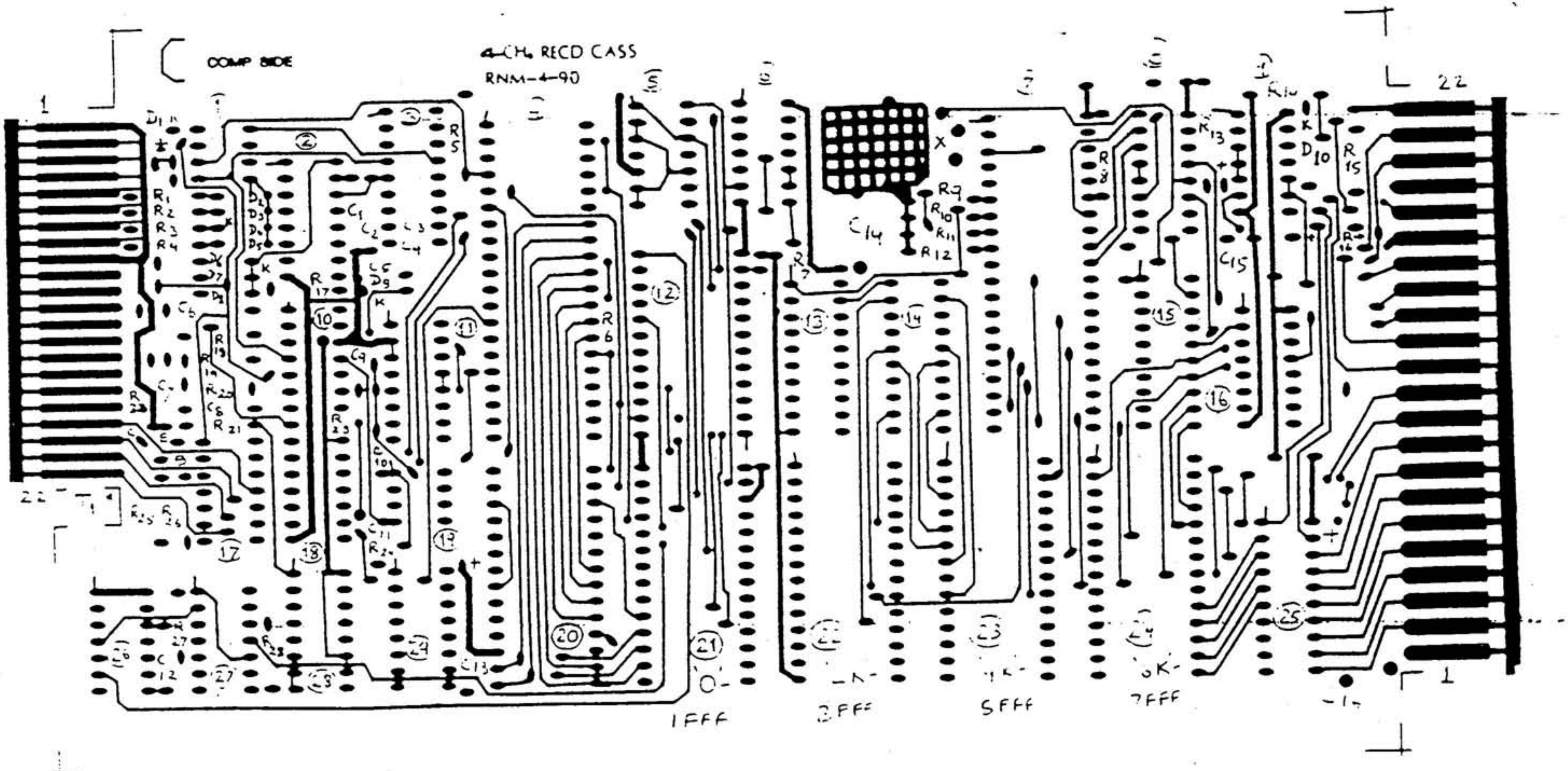


FIG 18 PCB LAY OUT COMP SIDE

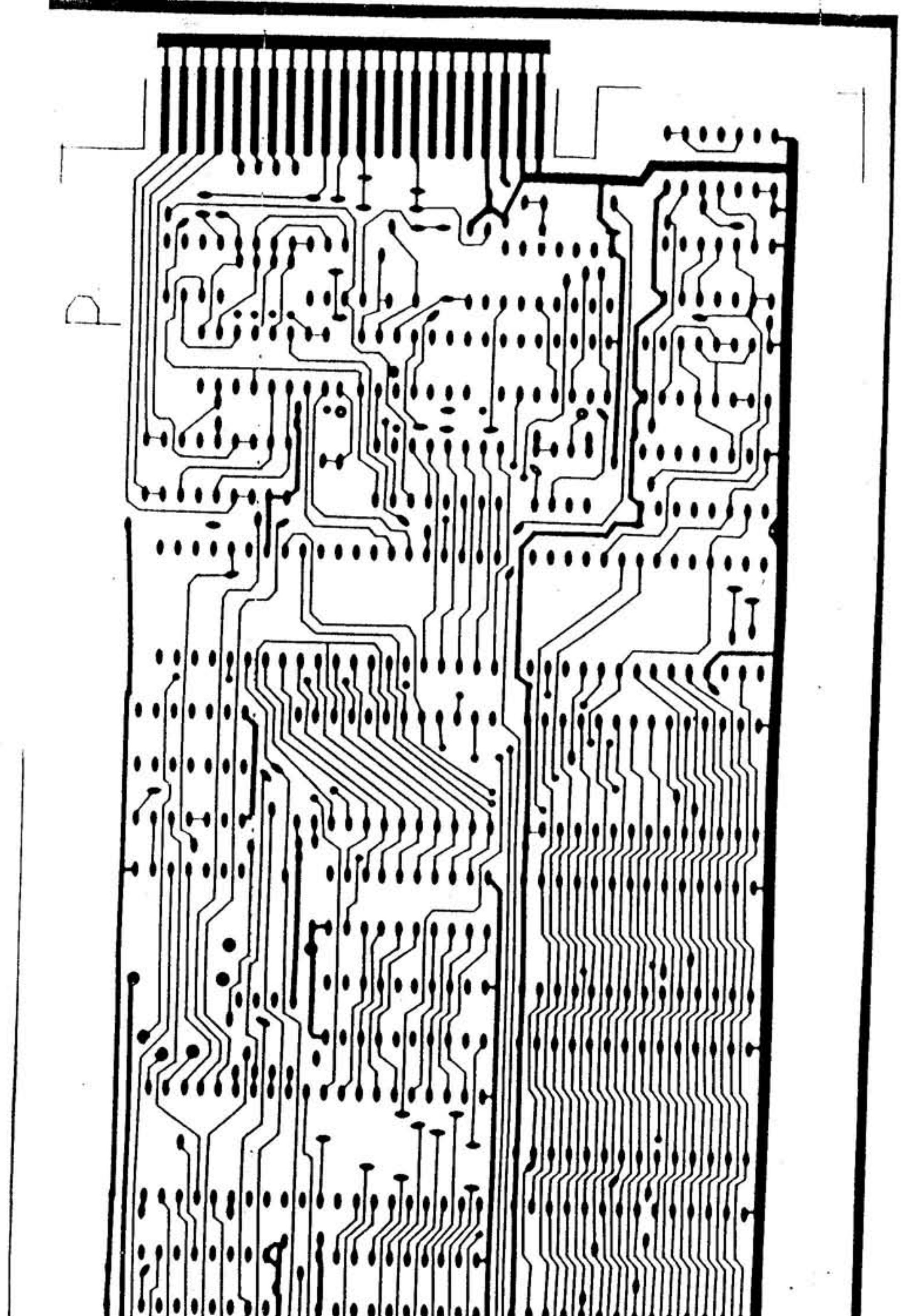


FIG.-9 PCB LAYOUT SOLDER SIDE

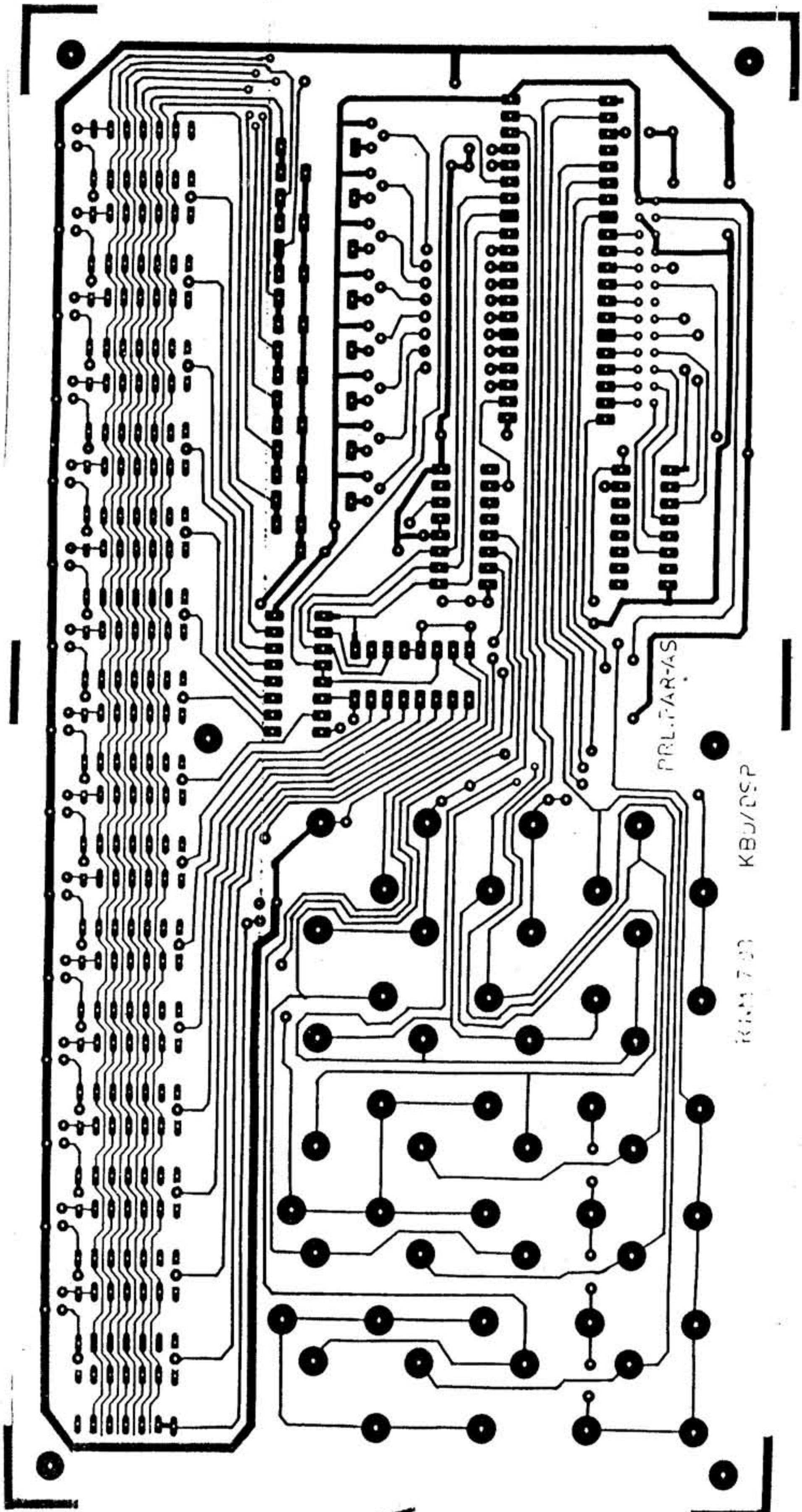
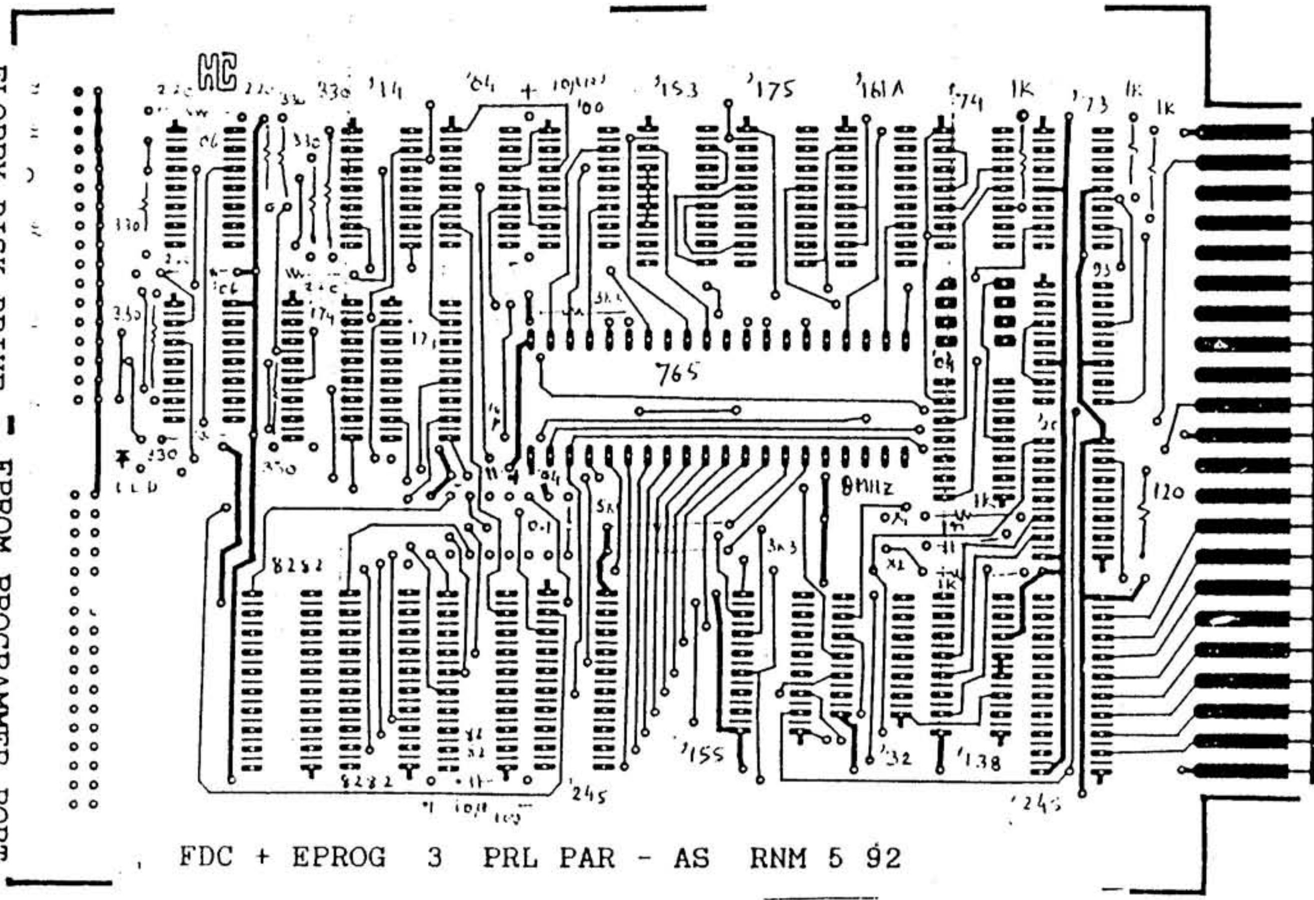
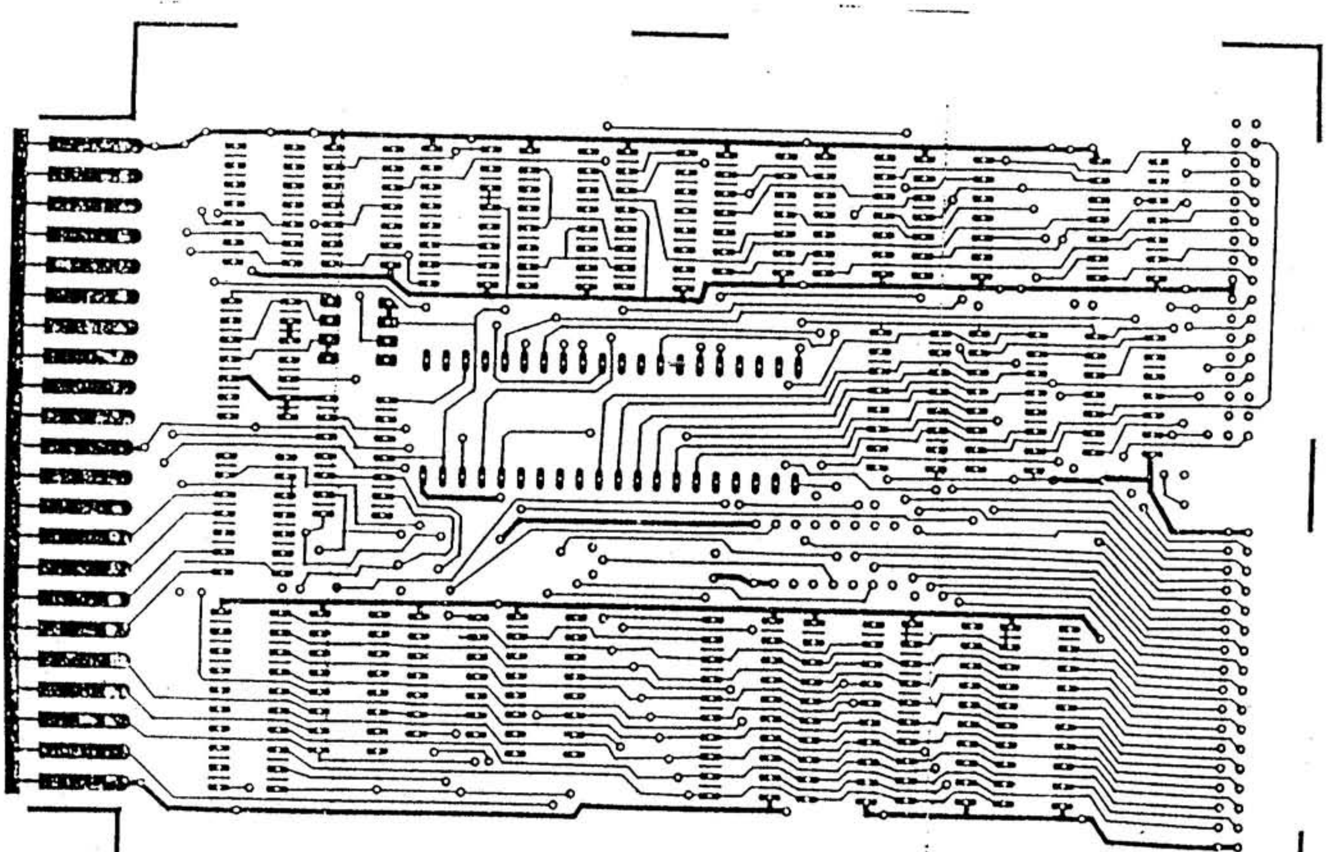


FIG. 20 PCB LAYOUT OF KEYBOARD DISPLAY CARD



COMPONENT SIDE

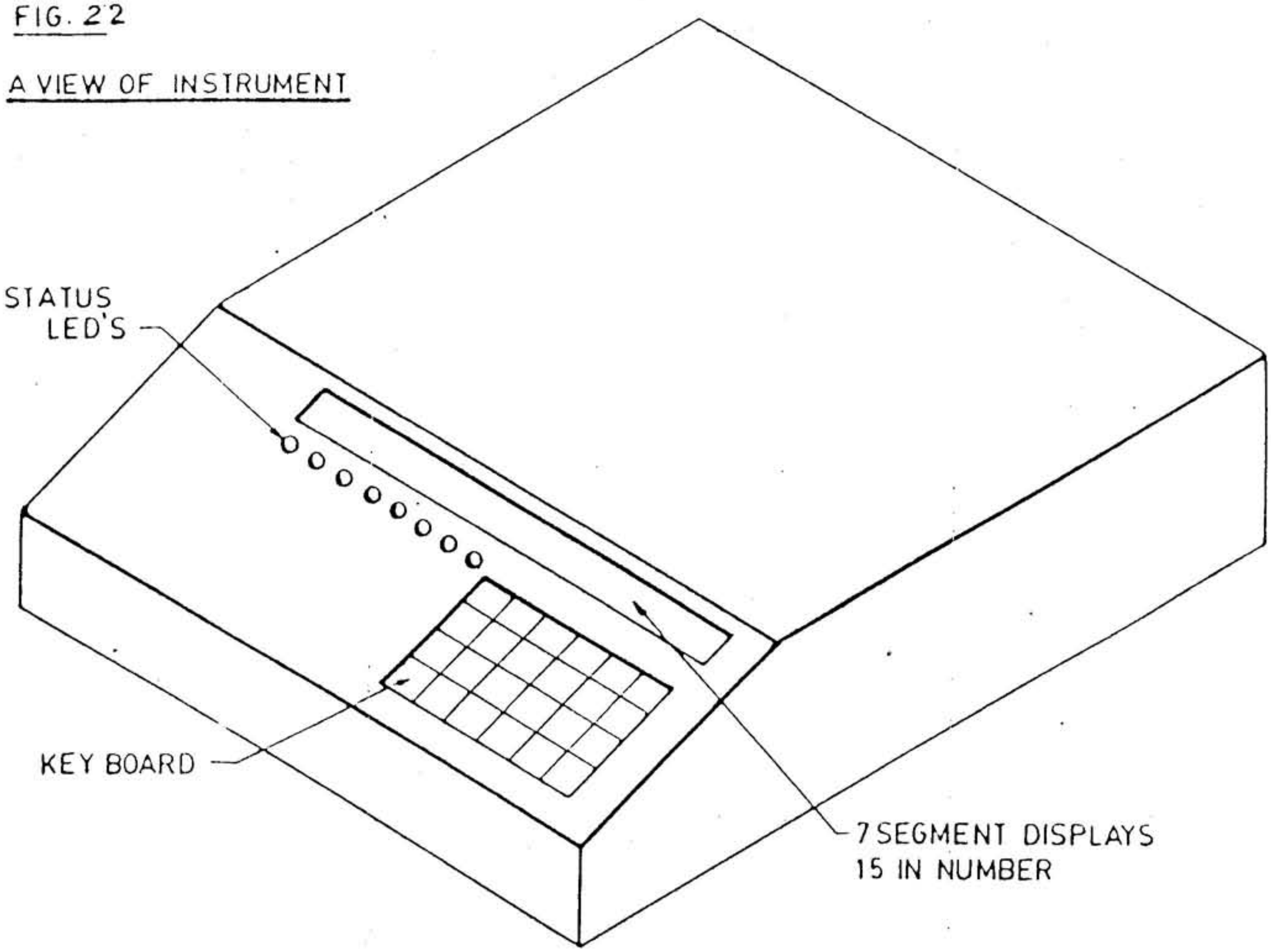


SOLDER SIDE

FIG. 21 PCB LAYOUT OF FLOPPY DISK CONTROLLER CARD

FIG. 22

A VIEW OF INSTRUMENT



REAR VIEW OF INSTRUMENT